

## مقدمه

گسترش دانش بشري ، توليد "اطلاعات" در حوزه هاي مختلف و نياز به استفاده ي كارآمد از آن ، محافل علمي ، دانشگاهي و فني را برآن داشته است كه با دقت و تامل بيشتر و با روشهاي سريع و مناسب تر از اطلاعات استفاده نمايند. امروزه نرم افزارهاي مختلف با تواناييهاي متفاوت و به وسيله ي شرکتهاي معتبر براي مديريت داده ها ايجاد شده اند و استفاده مي شوند.

در اين كتاب مفاهيم عام سيستم هاي ذخيره و بازيابي ، فايلينگ و بانك اطلاعاتي در فصلهاي 1 و 7 بيان شده است و در فصلهاي 2 تا 6 اين مفاهيم با استفاده از نرم افزار مديريت بانك اطلاعاتي Access 2003 به كار گرفته شده است. پيشنهاد ميشود قبل از انجام كار عملي ، مفهوم يا مفهوم هاي مورد نياز به دقت بيان گردد تا هنجارها بدانند "چرا" و براي رسيدن به چه "هدف" مي بايد از بانك اطلاعاتي استفاده كند.

مناسب است با ياد آوري مطالب مربوط به فايلينگ در كتاب برنامه سازي 3 ، تفاوت فايلينگ با بانك اطلاعاتي به درستي شناسايي گردد. در فصل 8 ، مراحل ايجاد يك بانك اطلاعاتي راهنمايي و توضيح داده شده است.

بر خود لازم مي دانم از هدايت هاي موثر اعضاي محترم كميسيون تخصصي كامپيوتر تشكر نمايم.

در پايان ، از معلماني محترم ، دانش آموزان گرامي و تاممي خوانندگان ارجمند كه مولف را از راهنمايي هاي خويش بهره مند مي سازند ، سپاسگزار مي كنم .

مولف

## 1- فصل اول: مفاهیم بانک اطلاعاتی 1

در این فصل به معرفی تعاریف و مفاهیم اولیه بانک‌های اطلاعاتی خواهیم پرداخت. اگرچه نرم‌افزارهای کاربردی به سهولت قابل استفاده هستند، اما بدون دانستن دقیق مفاهیم نمی‌توان از آنها به‌طور مناسب استفاده کرد و هیچ‌گاه نمی‌توان بانک اطلاعاتی کارآمدی را ایجاد و پیاده‌سازی کرد.

- در پایان این فصل انتظار می‌رود که هنرجو بتواند:
- انواع سیستم‌های ذخیره و بازیابی را بشناسد.
  - با مفهوم فیلد، رکورد، فایل آشنا شود.
  - بانک اطلاعاتی را تعریف کند.
  - سیستم مدیریت بانک اطلاعاتی و معماری آن را بشناسد.
  - با مفاهیم مدل رابطه‌ای آشنا شود.
  - سیستم بانک اطلاعاتی را طراحی کند.

### 1-1- کلیات

مدیریت بانک‌های اطلاعاتی یکی از انواع سیستم‌های "ذخیره و بازیابی اطلاعات"<sup>2</sup> است. سیستم ذخیره و بازیابی سیستمی است که به کاربر امکان می‌دهد تا داده‌ها و اطلاعات خود را ذخیره، بازیابی و پردازش نماید.

- نظر به تعریف عام بالا، موارد زیر قابل ذکر است:
1. کاربر می‌تواند دارای قابلیت "برنامه‌سازی" یا فاقد این مهارت باشد.
  2. داده می‌تواند متن، تصویر، صوت و نظایر آن باشد.

سیستم ذخیره و بازیابی اطلاعات با گذشت زمان، از مفهوم ساده‌ی فایلینگ<sup>3</sup> به صورت زیر توسعه پیدا کرده است:

Filing System (FS)

سیستم فایلینگ

Data Management System (DMS)

سیستم مدیریت داده‌ها

---

<sup>1</sup> - در این کتاب پایگاه داده‌ها، بانک اطلاعاتی و بانک داده به یک معنی استفاده شده است.

<sup>2</sup> - Information Storage and Retrieval

<sup>3</sup> - Filing

پیش از تعریف سیستم مدیریت پایگاه داده بهتر است ، برخی از مفاهیم ، تعاریف مقدماتی و مفهوم سیستم فایلینگ بررسی شود تا با دقت بیشتری بتوانیم مفهوم بانک اطلاعاتی را بشناسیم .

## 2-1-1- مفاهیم مقدماتی در ذخیره و بازیابی

• **فیلد<sup>1</sup> : يك قطعه داده است.<sup>2</sup>**

هر فیلد دارای دو جزء می‌باشد :

1. نام
2. مقدار

مقدار حکم داده<sup>3</sup> را دارد و نام فیلد به همراه مقدار در حکم اطلاع<sup>4</sup> خواهد بود. به عنوان مثال:

نام خانوادگی = علوي

\_\_\_\_\_

نام فیلد                      مقدار فیلد

در واقع هر فیلد یک ویژگی یا صفت<sup>5</sup> به حساب می‌آید. توجه کنید که از نام فیلد و مقدار فیلد به همراه هم اطلاع پیدا می‌کنیم که نام خانوادگی فرد مورد نظر علوي است. مثال جالبتر این‌که : مقدار عددی "88965423" هیچ معنی خاصی ندارد. حال به این عبارت توجه کنید: "تلفن=88965423" ، با درج نام فیلد در کنار مقدار عددی ، شما اطلاع پیدا می‌کنید که آن عدد شماره تلفن است. اگر به جای تلفن ، شماره شناسنامه قرار گیرد اطلاع کسب شده متفاوت خواهد بود.

نکته :

یکی از خصوصیات مهم فیلد ، نوع داده‌ای است که در آن قرار می‌گیرد. مثلاً Integer یا Real یا این‌که Text باشد و ... .

• **رکورد<sup>6</sup> : به مجموعه‌ای فیلدهای مرتبط ، يك رکورد می‌گوییم .**

به عنوان مثال ، قالب رکورد مربوط به يك دانش‌آموز می‌تواند شامل نام فیلدهای زیر باشد :

نام ، نام خانوادگی ، سال تولد ، مقطع تحصیلی ، کلاس و ...

<sup>1</sup> - Field

<sup>2</sup> - فیلد واحد معنایی داده است. (Semantic Data Unit)

<sup>3</sup> - Data

<sup>4</sup> - Information

<sup>5</sup> - Attribute

<sup>6</sup> - Record

محتوای رکورد دانش‌آموزان می‌تواند برای دو دانش‌آموز همکلاسی به صورت زیر باشد :

1. مجید ، شجاعی ، 1370 ، اول دبیرستان ، کلاس 1 الف و ...
2. حسن ، کمالیان ، 1371 ، اول دبیرستان ، کلاس 1 الف و ...

محتوای رکورد می‌تواند دارای طول ثابت یا متغیر باشد. در حالت رکورد با طول ثابت ، تعداد ، ترتیب و اندازه‌ی فیلدهای متناظر در تمام رکوردها یکسان در نظر گرفته می‌شود و در حالت رکورد با طول متغیر ، چنین الزامی وجود ندارد.

- **فایل<sup>1</sup> :** مجموعه‌ای از رکوردها ، تشکیل فایل را می‌دهند. به عنوان مثال : فایل تحصیلی دانش‌آموزان ، محتوای فیلدهای رکوردهای مربوط به دانش‌آموزان را دارا می‌باشد.
- **کلید<sup>2</sup> :** فیلد یا زیرمجموعه‌ای از فیلدها ، که باعث یکتایی هر یک از رکوردها شود ، کلید گویند. به وسیله‌ی کلید می‌توان به یک رکورد مشخص دسترسی داشت. به عنوان مثال ، به رکوردهای زیر توجه کنید :

ردیف	شماره دانش‌آموزی	نام	نام خانوادگی
1	100	علی	حسینی
2	110	حسن	احمدی
3	108	علی	علوی
4	94	رضا	علوی

فیلد نام نمی‌تواند کلید باشد زیرا دو مقدار مشابه "علی" در رکوردهای ردیف 1 و 3 وجود دارد . همچنین فیلد نام خانوادگی نمی‌تواند کلید باشد زیرا مقادیر "علوی" در رکوردهای ردیف 3 و 4 وجود دارد. اما شماره دانش‌آموزی می‌تواند کلید باشد زیرا هیچ دو شماره دانش‌آموزی برابر وجود ندارد.

با توجه به رکوردهای این جدول ، فیلدهای نام و نام خانوادگی نیز با هم می‌توانند کلید باشند . اگرچه در یک محیط عملیاتی مدرسه‌ی واقعی ، ممکن است دو دانش‌آموز نام و نام خانوادگی یکسان داشته باشند.

### در انتخاب کلید موارد زیر قابل توجه است:

1. کلید می‌تواند ساده یا مرکب باشد.
2. ممکن است کلیدهای مختلفی اعم از ساده یا مرکب بتوان تشخیص

### کنجاوی :

آیا ممکن است در یک سیستم ذخیره و بازیابی کلید وجود نداشته باشد؟

### نکته :

شماره دانش‌آموزی یک صفت مجازی است که به دانش‌آموزان به صورت منحصر به فرد اختصاص می‌یابد و نقش کلید را دارد.

داد ولي يکي از آن‌ها را با توجه به ملاحظات ي به عنوان کلید اصلي<sup>1</sup> در نظر مي‌گيريم .

3. در برخي موارد به جاي استفاده از فيلدها و صفات موردنياز شناسايي شده ، به صورت مجازي فيلدي با مقدار يکتا ، به عنوان کلید اضافه مي‌شود. مانند شماره دانش‌آموزي. مجازي بودن يعني يك فرد زماني که در يك مدرسه ثبت‌نام مي‌کند و دانش‌آموز آن مدرسه مي‌شود ، يك شماره ي دانش‌آموزي مي‌گيرد و زماني که فارغ التحصيل مي‌شود ديگر اين شماره را از دست مي‌دهد.

4. مقدار فيلدي که کلید در نظر گرفته مي‌شود ، تهی(خالي) نيست و طبق تعريف کلید ، مقدار تکراري ندارد.

### 3-1- عمليات روي رکوردها

اين عمليات عبارت است از :

- |    |                       |             |
|----|-----------------------|-------------|
| 1. | بازيابي رکوردها       | Retrieve    |
| 2. | درج رکوردها           | Insert      |
| 3. | حذف رکوردها           | Delete      |
| 4. | به هنگام سازي رکوردها | Update      |
| 5. | تغيير ساختار          | Restructure |

### 3-1-1- بازيابي رکوردها

منظور از بازيابي به دست آوردن رکورد يا رکوردهايي است که داراي شرايط خاصي هستند. به عنوان مثال مي‌خواهيم اطلاعات دانش‌آموزاني که معدل آن‌ها بيش از 16 است را به دست آوريم. به اين کار بازيابي گفته مي‌شود. ممکن است نتيجه ي بازيابي يك يا چند رکورد باشد.

به طور كلي براي دستيابي و بازيابي رکوردها دو روش وجود دارد :

1. ترتيبي
2. مستقيم

در روش ترتيبي ، رکوردها از ابتدا يکي پس از ديگري بررسي مي‌شوند تا "محتوای" موردنظر بازيافت شود. مثلا فرض کنيد مي‌خواهيم منزل شخصي به نام علي علوي را در يك ساختمان چند طبقه پيدا کنيم. براي يافتن آپارتمان موردنظر بايد مثلا از طبقه ي اول ، زنگ واحدها را پشت سرهم بزنيم تا به منزل اين شخص برسيم. اين روش کند و زمان بر است.

در روش مستقيم با توجه به محتوا ، "آدرس" رکورد به دست مي‌آيد و رکورد موردنظر به طور مستقيم با آن آدرس بازيابي مي‌شود. روش مستقيم به وسيله ي ساختارهاي مختلف پياده سازي مي‌شود مثل شاخص<sup>2</sup>. در مثال بالا اگر به دفتر سرايدار آپارتمان مراجعه کنيم. وي مي‌گويد منزل اين شخص مثلا طبقه ي چهارم ، واحد 2 است. ما مي‌توانيم از آسانسور استفاده کنيم و سريع به منزل اين شخص برسيم. حتي اگر آسانسور دردسترس نباشد از راه پله‌ها خود را به طبقه چهارم مي‌رسانيم بدون آن‌که در طبقه‌ها وقت صرف جستجو کنيم<sup>3</sup>.

<sup>1</sup> - Primary Key

<sup>2</sup> - Index

<sup>3</sup> - اگرچه از نظر زماني نسبت به روش ترتيبي سريع تر است اما مصرف حافظه در آن بيشتر خواهد بود

### 2-3-1- درج رکوردها

منظور ، افزودن رکوردي است که تاکنون در سیستم وجود نداشته است. درج می‌تواند در انتهاي فایل یا محل منطقي آن رکورد صورت گیرد. محل منطقي ، محلي است که نظم فایل را حفظ می‌کند.

به عنوان مثال فرض کنید ، فایلي به صورت زیر وجود دارد. اگر بخواهیم دانش‌آموزي با معدل 15 را اضافه کنیم و نظم فایل که به صورت صعودي مقادیر معدل مرتب است حفظ شود ، باید این رکورد بین رکورد اول و دوم قرار گیرد. یعنی رکوردي بین اول و دوم اضافه کنیم و اطلاعات دانش‌آموز جدید را در آن ثبت کنیم.

درج  
رکوردي  
با



	فیلد معدل	سایر فیلدها .....
رکوردي اول	144	
رکوردي دوم	157	
رکوردي سوم	178	
رکوردي چهارم	180	
رکوردي پنجم	20	

**شکل 2-1: درج در محل منطقي (نظم صعودي فیلد معدل حفظ شده است)**

### 3-3-1- حذف رکوردها :

در صورتی که به رکورد یا رکوردهایی نیاز نداشته باشیم ، آن را از فایل مربوطه حذف می‌کنیم.

### 4-3-1- به‌هنگام‌سازي رکوردها

اصلاح مقادیر فیلد یا فیلدهایی از رکورد ، در اصطلاح به‌هنگام‌سازي نام دارد. به عنوان مثال تغییر نمره ي يك درس دانش‌آموز به‌هنگام‌سازي محسوب می‌شود.

### 5-3-1- تغییر ساختار

اگر به هر دلیل در ساختار رکوردها تغییر ایجاد شود مثلاً اندازه ي فیلد تغییر کند یا فیلد جدیدی اضافه شود ، تغییر ساختار صورت می‌گیرد.

### 1-1- سیستم فایلینگ

همان‌طور که گفتیم سیستم فایلینگ ، اولین نوع سیستم ذخیره و بازیابی اطلاعات است. در این سیستم نرم افزارهای کاربردی ، مدیریت ذخیره و بازیابی داده‌ها را نیز برعهده داشتند. در این سیستم برای کاربردهای مختلف ، فایل‌های جداگانه‌ای طراحی و پیاده‌سازی می‌شوند. در سیستم فایلینگ انجام عملیات به وسیله‌ی برنامه‌سازی و پیاده‌سازی الگوریتم‌ها انجام می‌شود. زمانی که سیستم بزرگ و پیچیده می‌شود ، هر تغییر جزئی در خواسته‌ها ، منجر به کدنویسی می‌گردید. به علت وجود فایل‌های مختلف ، کنترل رعایت استاندارد در زمان پیاده‌سازی سخت است. به عنوان مثال فیلد نام‌خانوادگی در دو فایل ممکن است اندازه‌های مختلف داشته باشد. در سیستم فایلینگ، ممکن است "ناسازگاری داده‌ها" به وجود می‌آید. در نظر بگیرید شخصی در زیرسیستم اول خود را "علی‌علوی‌تهرانی" و در زیرسیستم دوم "علی‌علوی" معرفی کند. بدین ترتیب مثلاً در مقایسه‌ی دو زیرسیستم ، این دو مقدار یکسان نیستند و این به معنی ناسازگاری داده است. همچنین احتمال بروز افزونگی<sup>1</sup> نیز وجود دارد. یعنی محتویات بعضی از فیلدها به صورت متعدد و تکراری ذخیره شوند. به‌طور مثال ، مشخصات سکونت یک فرد هم در سیستم فایلینگ اول و هم در سیستم دوم ثبت گردد. نتیجه‌ی افزونگی ، اتلاف حافظه و مشکلات در انجام عملیات مبنایی است.

#### **نکته :**

افزونگی ، ناشی از طراحی نامناسب است که در سایر سیستم‌های ذخیره و بازیابی نظیر بانک‌اطلاعاتی نیز وجود دارد و منحصر به سیستم فایلینگ نیست

برنامه‌سازی در این روش معمولاً با استفاده از یک زبان سطح بالا صورت می‌گیرد و معمولاً فایل‌های سیستم قابل استفاده به وسیله‌ی سایر سیستم‌های فایلینگ نیستند.

به‌طور کلی معایب روش فایلینگ را می‌توان به صورت زیر خلاصه کرد:

1. احتمال بروز افزونگی (تکرار در ذخیره‌سازی)
2. حجم زیاد کدنویسی
3. احتمال بالای ناسازگاری داده‌ها
4. سخت بودن اعمال استاندارد در سیستم‌ها

## **1-2- پایگاه داده‌ها**

سیستم مدیریت پایگاه داده ، یکی از سیستم‌های توسعه یافته‌ی ذخیره و بازیابی اطلاعات است. اگرچه این مفهوم

<sup>1</sup> - Redundancy

بسیار شناخته شده و رایج است ، اما واقعیت آن است که تعریف کاملاً واحدی درمیتون معتبر برای آن ذکر نشده است در نگرش بانك اطلاعاتي ، طراحی به صورت یکپارچه انجام میگیرد. از آنجایی که یک محیط عملیاتی می تواند دارای زیرمحیطهای مختلفی باشد ، نگرش هنجار و باعث می شود برخی معایب روش فایلینگ نظیر ناسازگاری داده ها و اعمال نشدن استانداردها رفع گردد ، مثلاً برای فیلد نام در تمام محیط، مقدار یکسانی از حافظه اختصاص می یابد . حال آنکه در روش فایلینگ به علت مجزا بودن فایلها به سختی و با صرف هزینهی بالا می توان نگرش هنجار داشت.

یکپارچه سازی و مدیریت متمرکز باعث جلوگیری از تکرار در ذخیره سازی (افزونگی) خواهد شد و در نتیجه از حافظه ، استفاده بهتری صورت میگیرد. در مجموع می توان پایگاه داده را به صورت زیر تعریف کرد :

**"مجموعه ای از داده ها به صورت یکپارچه با حداقل افزونگی تحت یک سیستم متمرکز و در چارچوب یک مدل داده ای"**

با ترکیب دستاوردهای شبکه ای ، پایگاه داده می تواند به وسیلهی چندکاربر و همزمان استفاده شود.

#### **کنجکاو ی :**

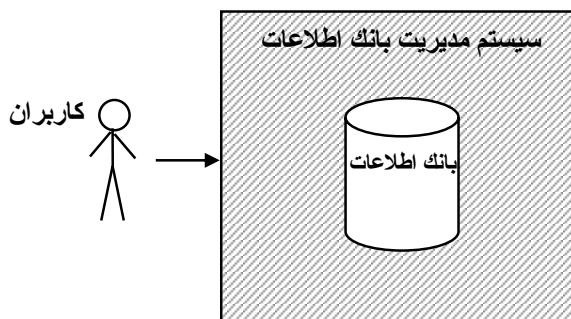
تعاریف مختلف پایگاه داده را بررسی کنید.



### 1-3- سیستم مدیریت پایگاه داده<sup>1</sup>

اصلی‌ترین تفاوت روش پایگاه داده با روش‌های قبلی مثل سیستم فایلینگ، وجود حصاری نفوذناپذیر به نام "سیستم مدیریت بانک اطلاعاتی" است. هرگونه دستیابی به داده‌ها باید از طریق آن انجام شود.

در DBMS، بانک اطلاعاتی و تمامی فایل‌های آن فقط و فقط در اختیار این نرم‌افزار قدرتمند هستند. کاربران درخواست خود را به این نرم‌افزار ارسال می‌کنند و در صورت تایید، کار خواسته شده انجام می‌گیرد.



### 1-6-1- معماری پایگاه داده

سوال این است که چگونه می‌توان در یک سیستم بزرگ و پیچیده بانک اطلاعات، بین کاربران و داده‌های بسیار زیاد ذخیره شده روی رسانه ارتباط برقرار کرد و چگونه سیستم مدیریت پایگاه داده درخواست‌های کاربران برای عملیات پایگاه داده‌ای (مانند: بازیافت، درج، به‌هنگام‌سازی، حذف و غیره) را اداره و کنترل می‌کند.

پاسخ این سوال در معماری سه لایه‌ای است که به وسیله کمیته ANSI/SPARC<sup>2</sup> عرضه شده است. در ایجاد این معماری سطوح به گونه‌ای از هم مجزا شده‌اند که برای رسیدن به هدف با هم به خوبی مراوده و تعامل کنند و هم به اندازه‌ی کافی مجزا و مستقل باشند که تغییر در یک سطح به سطوح دیگر تسری پیدا نکند. این سه لایه عبارت است از:

1. سطح خارجی<sup>3</sup>

2. سطح مفهومی<sup>4</sup>

3. سطح داخلی<sup>5</sup>

❖ سطح خارجی: همان دید کاربری است. بالاترین سطح انتزاع است. به این ترتیب که فقط بخشی از پایگاه داده که به مسایل کاربر و یا برنامه‌ی کاربردی مربوط است را شامل می‌شود.

<sup>1</sup> - Database Management System (DBMS)

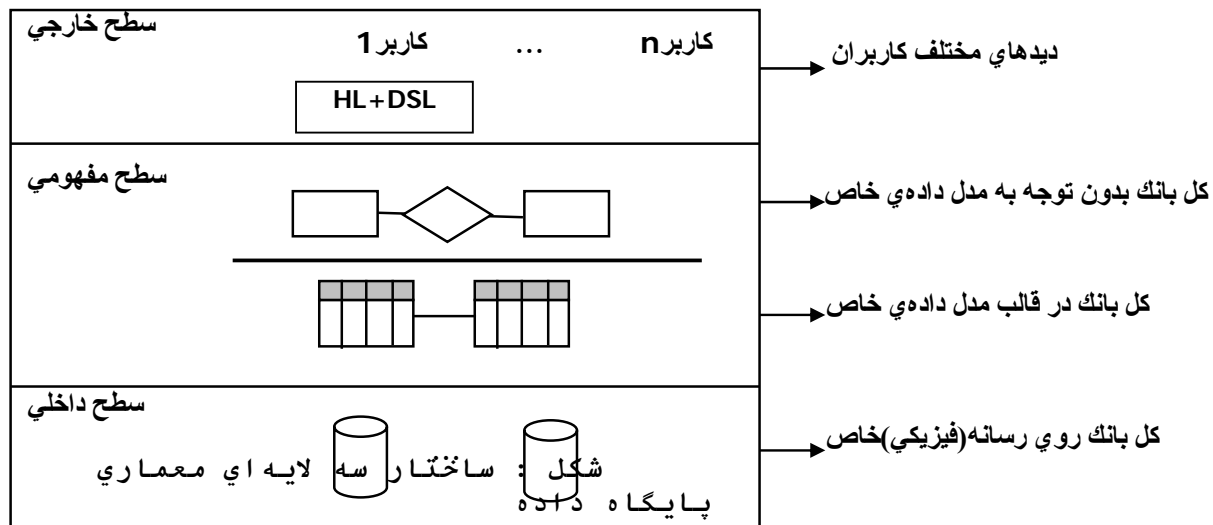
<sup>2</sup> - ANSI Standard Planning and Requirements Committee

<sup>3</sup> - External Level

<sup>4</sup> - Conceptual Level

<sup>5</sup> - Internal Level

- ❖ سطح مفهومي : ساختار پایگاه داده است. تمام موجودیتها و ارتباطات بین آنها را شامل می‌شود.
- ❖ سطح داخلي : جزییات ذخیره‌سازی است. پایین‌ترین سطح انتزاع است و به روش‌های فیزیکی ذخیره و بازیابی نزدیک است. چگونگی ذخیره‌سازی را نشان می‌دهد. ساختارهای داده را توضیح می‌دهد و همچنین روش‌های دسترسی را نشان می‌دهد. سطح داخلي ، سطح فایلینگ است.



**نکته :**  
 HL : زبان برنامه نویسی غیربانکی است. مثل Visual Basic و ...  
 DSL : زبان برنامه نویسی بانکی یا به عبارتی زبان فرعی داده‌ای است. مثل SQL و ...

## 2-6-1- مدل داده‌ای

در تعریف پایگاه داده از "مدل داده‌ای" نام برده‌ایم. سیستم مدیریت بانک اطلاعاتی پنهان از دید کاربر، هرگونه دستیابی به داده‌ها را انجام می‌دهد. به این منظور در معماری پایگاه، مدل داده‌ای استفاده شده است. مدل‌های داده‌ای مختلفی در بانک‌های اطلاعاتی استفاده می‌شوند که سه نوع متداول آن عبارت است از:

1. رابطه‌ای Relational Data Structure
2. سلسله‌مراتبی Hierarchical Data Structure
3. شبکه‌ای Network Data Structure

### کنجکاو ی :

در مورد مدل‌های داده‌ای دیگر تحقیق کنید.

همان‌طور که دیدیم، سیستم مدیریت پایگاه داده‌ها، نرم‌افزاری است پیچیده که واسط بین کاربران و محیط فیزیکی ذخیره‌ی داده‌ها است. از جمله سیستم‌های مدیریت پایگاه اطلاعات می‌توان از : Oracle ، My Sql ، SQL Server ، Access و ... نام برد.

برخی از این سیستم‌های نرم‌افزاری از دیگری پیچیده‌تر و بنابراین گران‌تر هستند و برای ذخیره‌سازی داده‌های بسیار زیاد استفاده می‌شوند. به برخی از آن‌ها نیز غیر از مدیریت داده‌ها، بخش‌های دیگری اضافه شده است که واسط کاربری را ارابه می‌کنند. مثلاً در Oracle ، بخش Development 2000 جدای از بخش مدیریت بانک عرضه می‌شود که طراح به کمک آن می‌تواند فرم‌ها و گزارش‌ها را ایجاد کند و آسان‌تر با محیط بانک ارتباط برقرار کند. در نرم‌افزار Access هم بخش‌هایی چون فرم‌ها، گزارش‌ها، ماژول‌ها و ... برای این منظور ایجاد شده است. که در بخش‌های بعدی بیشتر با آن آشنا می‌شویم.

## 1-4- مفاهيم پايه‌ي پايگاه داده‌ي رابطه‌اي

### 1-7-1- محيط عملياتي<sup>1</sup> و تعريف موجوديت :

محيط عملياتي ، محيطي است كه مي‌خواهيم يك سيستم ذخيره و بازيابي براي آن ايجاد كنيم. مانند : مدرسه يا دانشگاه ، بانك ، تعميرگاه ، فروشگاه و .... در هر محيط عملياتي تعدادي موجوديت<sup>2</sup> وجود دارد. موجوديت ، هر مفهوم و يا شيء در محيط عملياتي است . مثلا دانش‌آموز ، معلم ، كارمند مدرسه ، كلاس و درس موجوديت‌هاي محيط عملياتي مدرسه هستند. تمام موجوديت‌هاي يك محيط عملياتي مهم نيستند بلكه موجوديت‌هايي براي ما مهم هستند كه مي‌خواهيم در مورد آنها اطلاع داشته باشيم. به عنوان يك مثال ساده در محيط عملياتي مدرسه موجوديت‌هاي دانش‌آموز، معلم و درس براي سيستم فرضي ثبت‌نام مهم و قابل توجه هستند. در هر محيط عملياتي مجموعه‌اي از موجوديت‌هاي مناسب و مرتبط بر اساس مرز و محدوده‌ي سيستم مورد طراحي ، انتخاب مي‌شوند.

#### تمرين 1-1 :

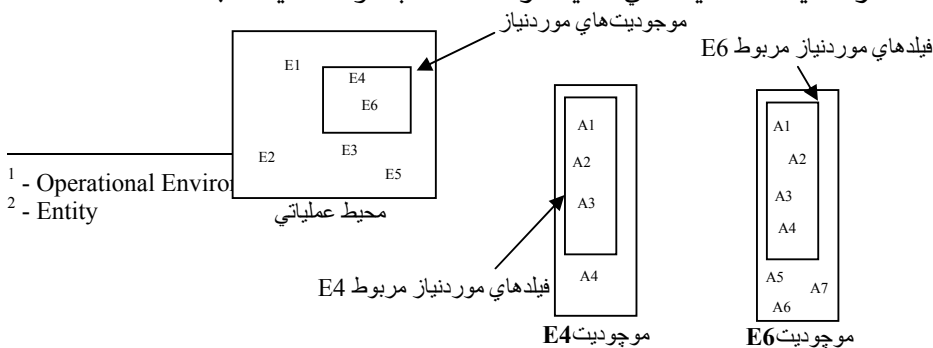
در محيط‌هاي عملياتي بانك ، خانواده و باشگاه ورزشي چه موجوديت‌هايي قابل تشخيص است؟

#### پرسش :

در چه مواقعي از فيلدهاي خود موجوديت براي تعريف كليد استفاده نمي‌شود؟

### 1-7-2- ويژگي‌هاي هر موجوديت :

صفات و ويژگي‌هاي هر موجوديت در واقع همان فيلدها هستند. در مورد انتخاب فيلدهاي يك موجوديت نيز مانند انتخاب موجوديت‌هاي محيط عملياتي فقط صفات موردنياز و مرتبط به عنوان فيلد در نظر گرفته مي‌شوند و تمام صفات يك موجوديت در نظر گرفته نمي‌شوند. به عنوان مثال ، در محيط عملياتي مدرسه در مورد اطلاعات مربوط به دانش‌آموز ويژگي‌هاي نظير قد ، وزن ، رنگ چشم در نظر گرفته نمي‌شود چراكه در محيط عملياتي موردنظر به آنها نياز نيست ، ولي ممكن است در محيط عملياتي ديگر مناسب و مفيد باشند.



شکل 1-1: زیرمجموعه موجودیتها و فیله‌های انتخاب‌شده برحسب نیاز

**کنجاوی :**

در نظر گرفتن محدوده‌ی محیط عملیاتی قبل از تشخیص موجودیت و ویژگی‌ها چه فایده‌هایی دارد؟

**تمرین 1-2 :**

در سه محیط عملیاتی تمرین 1-1 موجودیت‌های مرتبط به‌همراه فیله‌های موردنیاز را شناسایی کنید.

**مثال 1-1 :**

محیط عملیاتی مدرسه را در نظر بگیرید. موجودیت‌ها و ویژگی‌های هر موجودیت را تعریف کنید.

**موجودیت‌ها :** دانش‌آموز ، معلم و درس است.

**ویژگی‌های موجودیت دانش‌آموز :**

شماره دانش‌آموزی ، نام ، نام‌خانوادگی ، تلفن ، آدرس ، رشته تحصیلی.

**ویژگی‌های موجودیت معلم :**

نام ، نام‌خانوادگی ، تلفن ، آدرس.

**ویژگی‌های موجودیت درس :**

شماره درس ، نام درس ، تعداد واحد ، ساعت کلاس و نام کلاس.

\* دقت کنید نام معلم ویژگی درس نیست زیرا تداخل اطلاعاتی رخ می‌دهد ، بعدها خواهیم دید چرا مشخصه معلم به درس اضافه می‌شود.

**1-7-3- ارتباط<sup>1</sup>**

پیش از این عنوان شد که در محیط عملیاتی ، موجودیت‌هایی وجود دارند. معمولا بین موجودیت‌ها ارتباط یا ارتباط‌هایی وجود دارد که قابل شناسایی است. در

<sup>1</sup> - Relation

- محیط عملیاتی مدرسه ، بین موجودیتهای درس ، دانش‌آموز و معلم ارتباطی زیر قابل تعریف است :
1. معلم درس ارایه می‌کند.
  2. دانش‌آموز هر ترم چندین درس ثبت‌نام می‌کند.
  3. دانش‌آموز برای هر یک از درس‌های ثبت‌نام شده نمره می‌گیرد .

#### همان‌طور که ملاحظه می‌شود :

- در هر ارتباط موجودیتهایی شرکت دارند .
- هر ارتباط دارای یک مفهوم یا عملکرد است. مثلا در ارتباط اول "ارایه‌کردن" عملکرد می‌باشد.
- هر ارتباط دارای "ماهیتی" خاص است. مثلا در ارتباط دوم ماهیت ارتباط می‌تواند یک‌به‌چند باشد. یعنی یک دانش‌آموز چند درس اخذ می‌کند.

#### کنجاوی :

آیا بین سه موجودیت نیز می‌توان ارتباط برقرار کرد ؟ مثال بزنید ؟

- برای ایجاد ارتباط ، از کلیدخارجی استفاده می‌شود. کلید اصلی جدول اول را که به جدول دیگر اضافه می‌شود کلیدخارجی گویند .
- ارتباط مانند موجودیت می‌تواند ویژگی‌هایی داشته باشد .

#### 1-7-4- انواع ماهیت در ارتباط<sup>1</sup>

تناظر بین موجودیت‌ها ، ماهیت ارتباط است و به یکی از سه نوع زیر وجود دارد :

1. یک‌به‌یک 1:1
2. یک‌به‌چند 1:N
3. چندبه‌چند M:N

در ارتباط یک‌به‌یک ، یک نمونه از موجودیت اول فقط با یک نمونه از موجودیت دیگر ارتباط دارد. مثلا ، هر مدرسه یک مدیر دارد. در ارتباط یک‌به‌چند ، یک نمونه موجودیت با چندین نمونه از موجودیت دیگر مرتبط است. مثلا یک معلم چند درس ارایه می‌کند. در حالت سوم ، یک نمونه از موجودیت اول با چندین نمونه از موجودیت دوم مرتبط است و برعکس. مثلا یک دانش‌آموز چند درس اخذ می‌کند و یک درس به وسیله‌ی چند دانش‌آموز اخذ می‌شود.

<sup>1</sup> - Cardinality

### تمرین 1-3 :

مثالی از حالت چندبه‌چند ارائه کنید.

### کنجکاو :

حالتی را مثال بزنید که موجودیت با خودش مرتبط باشد.

### مثال 1-2 :

در محیط عملیاتی مدرسه ، ماهیت ارتباطها به صورت زیر است :

1. یک معلم چند درس ارائه می‌کند و هر درس به‌وسیله‌ی یک معلم ارائه می‌شود. (1:n)
2. دانش‌آموز (در هر ترم) چندین درس ثبت‌نام می‌کند و یک درس به‌وسیله‌ی چندین دانش‌آموز انتخاب می‌شود. (m : n)
3. یک دانش‌آموز برای هر درس ثبت‌نام شده ، یک نمره می‌گیرد. یک نمره به یک درس داده می‌شود. (1: n)

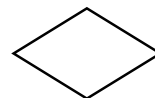
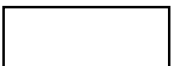
### 1-7-5- نمودار ارتباط- موجودیت / ER<sup>1</sup>

برای نشان دادن ارتباط بین موجودیتها و بیان عملکرد ارتباط ، از نمودار ER استفاده می‌شود. به عبارت دیگر ، یک مدل‌سازی معنایی از داده‌هاست.

اجزای نمودار ER عبارت است از :

1. موجودیتها
2. عملکرد ارتباط
3. ماهیت ارتباط
4. ویژگی‌های موجودیت

برای ترسیم نمودار ، از شکل‌هایی به صورت زیر استفاده می‌شود<sup>2</sup> :



<sup>1</sup> - Entity Relationship Diagram

<sup>2</sup> - شکل‌ها به‌صورت قراردادی است و در منابع مختلف متفاوت است.

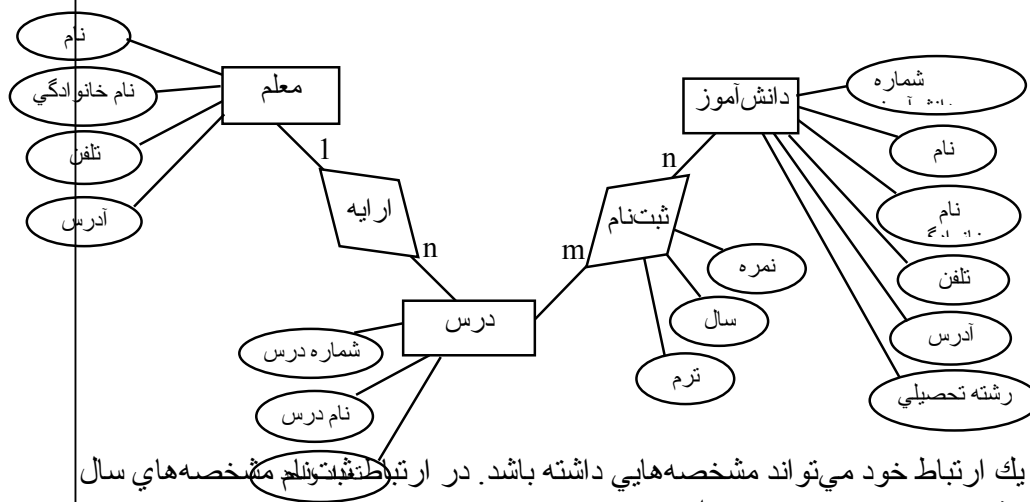
موجودیت‌ها از جنس "اسم" هستند مانند دانش‌آموز و درس. عملکرد ارتباط از جنس "فعل" یا "مصدر" است مانند اخذ کردن. ماهیت ارتباط تناظر بین موجودیت‌ها را نشان می‌دهد، ویژگی‌های موجودیت همان صفات قابل شناسایی هر موجودیت است.

**نکته:**

ماهیت ارتباط روی خطوط متصل بین موجودیت‌ها در دو طرف شکل لوزی ذکر می‌گردد.

**مثال 1-3:**

نمودار ER محیط عملیاتی مدرسه به صورت زیر است:



دقت: یک ارتباط خود می‌تواند مشخصه‌هایی داشته باشد. در ارتباط ثبت‌نام مشخصه‌های سال، ترم و نمره‌ی درس وجود دارد.

**1-7-6- رابطه / جدول:**

مدل رابطه‌ای بر مبنای یک مفهوم ریاضی به نام "رابطه"<sup>1</sup> تعریف شده است. استحکام نظریه مجموعه‌ها و سهولت پیاده‌سازی آن باعث رواج این مدل داده‌ای شده

<sup>1</sup>- رابطه، زیرمجموعه‌ای از حاصل‌ضرب دکارتی مجموعه‌ها است. مفاهیم مرتبط در این نظریه به ترتیب عضو، مجموعه حاصل‌ضرب دکارتی، رابطه و تابع است.



است. بر این اساس هر بانک اطلاعاتی از تعدادی رابطه یا همان جدول تشکیل می‌شود. هر جدول می‌تواند نشان‌دهنده‌ی یک موجودیت و یا ارتباط بین موجودیت‌ها باشد. ER مثال 1-1 را دوباره ملاحظه نمایید. طبق مدل داده‌ای رابطه‌ای جدول‌های زیر را می‌توان در نظر گرفت:

- جدول دانش‌آموز (Student)
- جدول درس (Course)
- جدول معلم (Teacher)

Student:

نشانی	تلفن	نام خانوادگی	نام	شماره دانش‌آموزی

Teacher:

آدرس	تلفن	نام خانوادگی	نام

هر جدول با نام خود و مجموعه‌ی نام ستون‌هایش مشخص می‌شود. دقت کنید بهتر است در نرم افزار Access نام جدول و نام ستون‌ها به صورت لاتین باشد ولی در این‌جا فارسی آورده شده است.

**نکته :**

**محاسن مدل داده‌ای رابطه‌ای:**

1. از نظر نمایش، ساده و قابل فهم است و تنها از یک مفهوم اصلی به نام رابطه یا جدول استفاده می‌کند.
2. انواع ماهیت‌های ارتباط را پشتیبانی می‌کند.
3. مبنای ریاضی دارد.

هنگامی که طراح، موجودیت‌های موردنیاز و ویژگی‌های آن‌ها را به همراه ارتباط بین موجودیت‌ها شناسایی کرد، کافی است نام و نوع آن‌ها را در حالت جدولی به پایگاه داده معرفی نماید.

### 1-7-7- برای تبدیل ER به جدول به صورت زیر عمل می‌کنیم :

- هر موجودیت مستقل به یک جدول تبدیل می‌شود.
- صفات موجودیت‌ها به فیلدهای جدول مربوطه تبدیل می‌شوند.
- موجودیت‌هایی که ارتباط 1:1 دارند به یک جدول تبدیل می‌شوند. [اگرچه ممکن است تداخل اطلاعاتی رخ دهد]. در مواردی نیز مانند 1:n یکی از جدول‌ها به دیگری منتقل می‌شود.

- در ارتباط  $n : 1$  ، کلید اصلی جدول طرف 1 به جدول طرف  $n$  اضافه می‌شود.

- ارتباط  $m : n$  ، تبدیل به یک جدول می‌شود که کلید اصلی دو موجودیت را شامل می‌شود.

### 1-5- قدم‌های طراحی بانک اطلاعاتی :

در طراحی هر سیستم اطلاعاتی ، صرف‌نظر از روش و متدولوژی طراحی باید ابتدا محیط سیستم شناخته و مطالعه شود ، سپس نیازسنجی صورت می‌گیرد یعنی تعیین نیازهای اطلاعاتی-پردازشی ، تشخیص محدودیت‌ها و قواعد حاکم بر محیط.

روش‌های مختلفی برای طراحی وجود دارد. با بیان ساده می‌توان گفت که در طراحی بانک داده ابتدا مدلسازی معنایی داده صورت می‌گیرد یعنی ابتدا طراحی مفهومی انجام می‌شود و نمودار ER رسم می‌گردد و در طراحی سیستم اطلاعاتی ، معمولا ابتدا تحلیل فرایندی صورت می‌گیرد و پردازش‌ها و گردش اطلاعات شناسایی می‌شوند . با مشخص شدن پردازش‌ها ، داده‌هایی که از طریق فرم‌ها ردوبدل می‌شوند و تغییراتی که روی آن‌ها انجام می‌شود ، مشخص می‌گردند و به این ترتیب ذخایر داده‌ای شکل می‌گیرند و برنامه‌ی کاربردی طراحی می‌شود یعنی واسط‌های کاربری مثل فرم‌ها و گزارش‌ها و کنترل‌های موردنیاز تعیین می‌شود. در این روش‌ها ، پس از انجام طراحی تفصیلی ، جدول‌ها ، فیلدها و کلید اصلی آن‌ها از نمودار ER و یا ذخایر داده‌ای تعیین می‌شوند. روش تبدیل موجودیت‌ها و رابطه‌ها (نمودار ER) را به جدول‌ها ، فیلدها و کلید قبلا توضیح داده ایم.

بحث شناخت سیستم و طراحی فرم‌ها و گزارش‌ها به مبحث تجزیه و تحلیل سیستم‌های اطلاعاتی مرتبط است که خود محدوده‌ی وسیعی دارد و از بحث ما خارج است.

با این سرآغاز طراحی یک بانک داده را به قدم‌های زیر ساده می‌کنیم :

#### کنجکاوئ :

یک روش و متدولوژی طراحی نام برده و خصوصیات آن را به طور خلاصه بیان کنید.

### برای مطالعه‌ی بیشتر :

در هر روش و متدولوژی طراحی ، مراحل انجام طراحی ، اصول و مفروضات مورد استفاده ، ابزار و نکات کاربردی و امحصولات و نتایج حاصل در هر مرحله و همچنین ارتباط بین مراحل ، چگونگی عبور از هر مرحله ، روش کنترل مستندات حاصل در هر مرحله ، و در برخی متدولوژی‌ها ممیزی ، کنترل و حسابرسی کیفیت مراحل و محصولات هر مرحله مشخص است. به‌طور کلی سه روش وجود دارد : فرایندگرا (Process Oriented) ، داده‌گرا (Data Oriented) و شی‌گرا (Object Orienter). روش فرایندگرا مانند SSADM ، Jackson ، Urdon و غیره. داده‌گرا مانند روش (IE) Engineering Information و روش شی‌گرا مانند RUP(UML).

### 1. ابتدا هدف از طراحی پایگاه داده را مشخص کنید.

مشخص کنید چه نوع اطلاعاتی باید از بانک دریافت شود. با افرادی که از بانک استفاده خواهند کرد صحبت کنید و در مورد فرم‌ها و گزارش‌های موردنیاز ، اطلاعات جمع‌آوری کنید.

### 2. نمودار ER را طراحی کنید.

موجودیت‌ها ، ویژگی‌ها و ارتباط بین آن‌ها را شناسایی کنید و نمودار ارتباط-موجودیت‌ها (ER) را ترسیم کنید. برای هر موجودیت کلید اصلی را تعیین یا ایجاد کنید.

#### نکته :

به‌خاطر بسپارید که يك موجودیت نباید اطلاعات افزونه داشته باشد یعنی هر موجودیت باید اطلاعات يك موضوع را نگهداری کند و از اختلاط اطلاعاتی پرهیز شود.

### 3. جدول‌ها و فیلدهای آن‌ها را مشخص کنید.

نمودار ER ترسیم شده را به جدول‌ها ، فیلدهای مربوطه تبدیل کنید. بعدها خواهیم دید که جدول ایجادشده از نظر سطوح نرمال بررسی شده و ممکن است خود به جدول‌های جدید تجزیه شوند.

### 4. فرم‌های موردنیاز را تعریف کنید.

فرم‌های ورود ، ویرایش و حذف داده را رسم کنید و از وجود فیلدهای فرم در جدول‌های بانک مطمئن شوید.

### 5. گزارش‌های موردنیاز را تعریف کنید.

گزارش‌های موردنیاز کاربران بانک را معین کرده ، قالب نمایش را تعیین کنید. وجود داده‌های لازم و موردنیاز نمایش گزارش در جدول‌های بانک را بررسی نمایید.

#### **6. طراحی خود را اصلاح و بازنگری کنید.**

یافته‌های خود را با افرادی که از بانک استفاده می‌کنند مطرح کنید و بار دیگر نیازها را بررسی و طراحی خود را اصلاح و بازنگری نمایید.

#### **کنجکاو :**

تفاوت‌های طرح خود را با طراحی مثال بعد بررسی و تحلیل کنید.

#### **نکته :**

دقت کنید که ممکن است طراحی شما با جدول‌های زیر متفاوت باشد. این مساله نشان‌دهنده‌ی غلط بودن طراحی نیست. بلکه برحسب مفروضات ، دقت کار و محدوده‌ی سیستم کاملاً طبیعی است.

### مثال 1-4 :

محیط عملیاتی مدرسه را طراحی کنید.  
نمودار ER رسم شده است.

1. نمودار ER به جدول های زیر تبدیل می شود :

جدول معلم	کد معلم	نام	نام خانوادگی	تلفن	آدرس
<b>Tbl_Teacher</b>	Teacher Code	Teacher FName	Teacher LName	Teacher Tel	Teacher Address

\* در جدول معلم ، فیلد کدمعلم را با عنوان کلید اصلی اضافه می کنیم.

جدول دانش آموز	شماره دانش آموز	نام	نام خانوادگی	تلفن	آدرس	سال
<b>Tbl_Student</b>	Student Code	Student FName	Student LName	Student Tel	Student Address	Year

جدول درس	شماره درس	نام درس	تعداد واحد	کد معلم
<b>Tbl_Course</b>	Course Code	Course Name	Course Unit	Teacher Code

\* ارتباط 1:n بین درس و معلم ، کلید جدول معلم (کدمعلم) به جدول درس اضافه می شود. دقت کنید که کدمعلم در جدول درس بخشی از کلید اصلی نیست.

جدول ثبت نام	سال	ترم	شماره درس	شماره دانش آموز	نمره درس
<b>Tbl_Register</b>	Year	Term	Course Code	Student Code	Course Grade

\* ارتباط n:m بین دانش آموز و درس به جدول ثبت نام تبدیل می شود و کلید اصلی جدول درس (شماره درس) و کلید اصلی جدول دانش آموز (شماره دانش آموز) به آن اضافه می شود.

چون ارتباط n:m مفروض خود دارای ویژگی هایی است پس این دو کلید به تنهایی کلید اصلی آن جدول نیستند و ترکیب آن دو با فیلد سال و ترم جدول کلید اصلی شده است.  
با توجه به اصول و مفروضات طراحی که در بخش نرمال سازی خواهیم دید ، جدول ثبت نام به دو جدول زیر تبدیل می شود :

جدول ثبت نام	سال	ترم	شماره دانش آموز	سریال ثبت نام
<b>Tbl_Register</b>	RegisterYear	Term	Student Code	Register Code

\* ما شماره سریال ثبت نام را به عنوان کلید اصلی اضافه می کنیم. می توانستیم سال و ترم را هم کلید بگیریم ولی بهتر است از کلید ترکیبی به علت پیچیدگی هایی که ایجاد می کنند استفاده نکنیم.

جدول دروس ثبت نام شده	سریال ثبت نام	شماره درس	نمره درس
<b>Tbl_Register</b>	Register Code	Course Code	Grade

\* دقت کنید که در جدول دروس ثبت نام شده شماره درس بخشی از کلید اصلی است.

شکل [ ] نشان دهنده کلید اصلی و شکل [ ] نشان دهنده کلید خارجی اضافه شده از جدول دارای ارتباط است.

**مثال 1-4 (ادامه) :**  
 2. فرم های مورد نیاز :

فرم تعریف درس

شماره درس :  
 نام درس :  
 تعداد واحد :  
 نام معلم :

فرم تعریف معلم

کد معلم :  
 نام :  
 نام خانوادگی :  
 تلفن :  
 آدرس :  
 توضیحات :

فرم ثبت نام

سال :  
 ترم :  
 نام دانش آموز :

شماره درس	نام درس	تعداد واحد

فرم تعریف دانش آموزان

شماره دانش آموز :  
 نام :  
 نام خانوادگی :  
 تلفن :  
 آدرس :  
 رشته تحصیلی :

فرم ورود نمره

سال :  
 ترم :  
 نام دانش آموز :

شماره درس	نام درس	تعداد واحد	نمره نهایی

**مثال 1-4 (ادامه) :**  
**3. گزارش‌های مورد نیاز :**

"گزارش ثبت نام"

سال :  
 ترم :  
 نام دانش‌آموز :  
 شماره دانش‌آموز :

دروس ثبت نام شده

شماره درس	نام درس	تعداد واحد	نام معلم

"گزارش لیست معلمان مدرسه"

کد معلم	نام معلم	تلفن	آدرس

"گزارش لیست حضور و غیاب"

سال :  
 ترم :  
 نام معلم :  
 درس :  
 تعداد واحد :  
 ساعت تشکیل کلاس :

شماره دانش‌آمو	نام دانش‌آمو																				

"گزارش کارنامه دانش‌آموزان"

نام و نام‌خانوادگی :  
 تاریخ گزارش :  
 رشته :  
 سال :  
 ترم :

شماره درس	نام درس	تعداد واحد	نمره

جمع واحد :  
 معدل :

## خلاصه ي فصل

سیستم ذخیره و بازیابی از مفهوم ساده ي فایلینگ توسعه پیدا کرده است. فیلد کوچکترین واحد ذخیره ي داده است. رکورد مجموعه ي فیلدهای مرتبط است و مجموعه ي رکوردها فایل را تشکیل می‌دهند. این مفاهیم پایه‌ای در هر سیستم اطلاعاتی دیگر نیز مطرح‌اند. فیلد یا زیرمجموعه‌ای از فیلدها که باعث یکتایی رکورد شوند را کلید گویند. کلیدها شرایط خاصی دارند که در فصل بحث شده است.

در سیستم‌های ذخیره و بازیابی، عملیات مشخصی روی رکوردها انجام می‌گیرد که عبارت است از بازیابی، درج، حذف، به‌هنگام‌سازی و تغییر ساختار. این عملیات را در محیط Access بیشتر بررسی می‌کنیم.

برای جلوگیری از درگیر شدن کاربر با مسایل برنامه‌نویسی و انجام سریع‌تر و دقیق‌تر تغییرات و در مجموع برای برقرارکردن ارتباط بین حجم بالای داده‌های ذخیره شده ي روی رسانه و کاربران معماری سه لایه‌ای مطرح شده است: سطح خارجی، سطح مفهومی، سطح داخلی.

نرم افزاری که این ارتباط را برقرار، مدیریت و اداره می‌کند، سیستم مدیریت بانک اطلاعاتی نام دارد که برای مدیریت دستیابی به داده‌ها و پنهان کردن آن از دید کاربران از یک مدل مفهومی به نام مدل داده‌ای استفاده می‌کند. مدل رابطه‌ای از جمله مدل‌های داده‌ای مهم است که بر پایه مفهوم ریاضی مجموعه‌ها بنا شده است و بیشتر سیستم‌های مدیریت بانک اطلاعاتی رایج از آن پشتیبانی می‌کنند.

در این فصل مفاهیم پایه مدل داده‌ی رابطه‌ای مانند: رابطه/جدول، ارتباط، ماهیت ارتباط، نمودار ارتباط-وجودیت (ER) و روش تبدیل این نمودار به جدول‌های پایگاه داده مطرح شده است.

## خودآزمایی

1. دلایل متغیر شدن طول رکورد چیست؟
2. معایب روش فایلینگ چیست؟
3. کدامیک از اجزای محیط ذخیره و بازیابی بیشتر اهمیت دارد؟ چرا؟
4. هدف از ایجاد سیستم ذخیره و بازیابی چیست؟



5. کلید را تعریف کنید و انواع آن را با ذکر مثال توضیح دهید.
6. معماری پایگاه داده را توضیح دهید.
7. مدل داده‌ای را تعریف کنید و توضیح دهید.
8. ماهیت ارتباط را تعریف کنید و در یک محیط عملیاتی مثال بزنید.
9. مراحل طراحی پایگاه داده را توضیح دهید.
10. لزوم طراحی قبل از پیاده‌سازی و شروع کار با نرم افزار Access را بیان کنید و مزایا و معایب احتمالی آن را بررسی کنید.

## 2- فصل دوم: مقدمات بانک اطلاعاتي Access 2003

در فصل اول ، با مفاهيم پايه‌اي بانک هاي اطلاعاتي و تعاريف آن آشنا شدیم. در اين فصل مي‌خواهيم با محيط بانک اطلاعاتي Access ، ابزارها و اجزاي آن آشنا شويم.

در پايان اين فصل انتظار مي‌رود که هنرجو بتواند :

- يك بانک نمونه را طراحي نمايد.
- نرم افزار Access را اجرا کند.
- از ابزارها ، پنجره‌ها و منوهاي محيط کار استفاده کند.
- فايل پايگاه داده را ايجاد کند.
- اجزاي بانک را تعريف و هدف کاربردي هريك را توضيح دهد.

### 2-1- راه اندازي نرم افزار Access

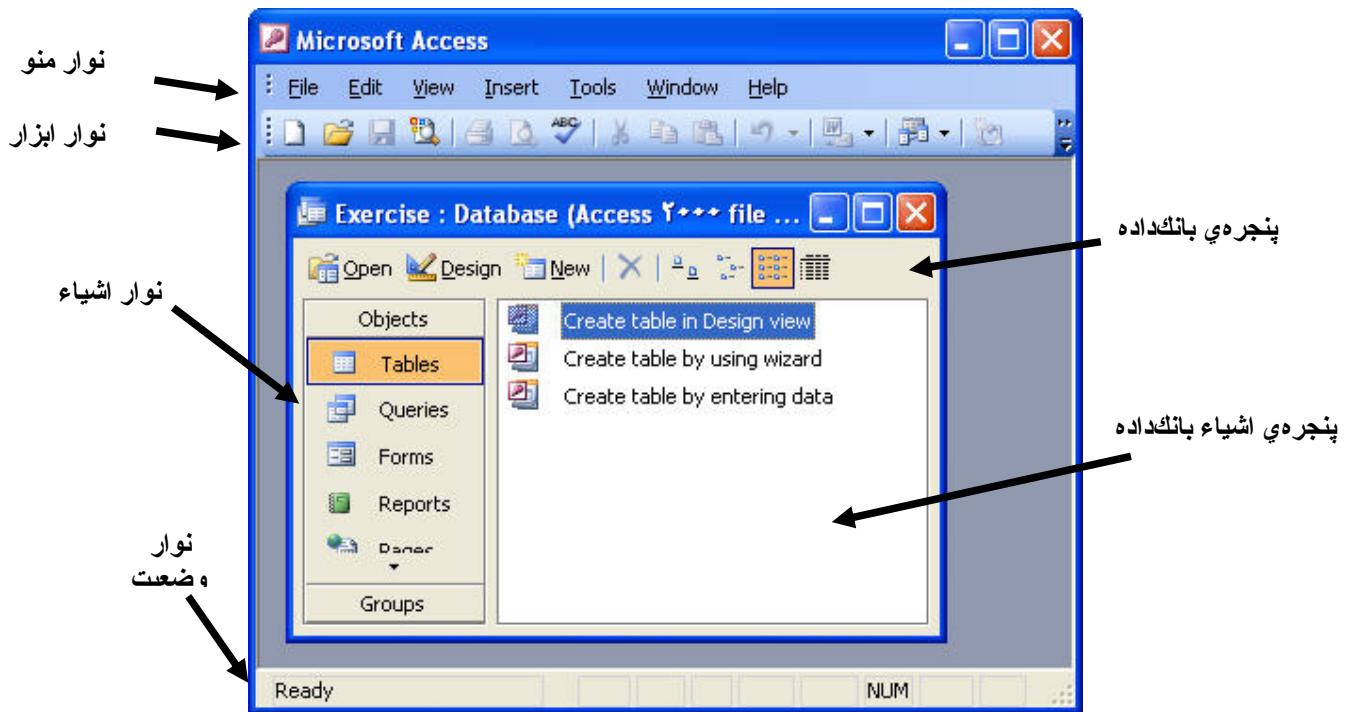
براي استفاده از اين نرم افزار مانند ساير برنامه‌ها ، بايد آن را در سيستم کامپيوتر موردنظر نصب کرده باشيم . نرم افزار Access به همراه بسته‌ي نرم افزار Office 2003 نصب مي‌شود که در بسته‌هاي نرم افزاري 1 و 2 با نصب آن آشنا شده ايد.

### 2-2- محيط کار ، منوها و ابزارهاي محيط

اين نرم افزار نيز مانند ساير برنامه‌هاي تحت ويندوز شامل منوها و انواع ابزارهاي کار است که در يك محيط کار قرار گرفته اند . تمامي قابليت‌هاي اين نرم افزار به وسيله‌ي اين محيط کار قابل اجرا و دسترسي است (شکل 1-2).

#### کنجاوي :

نوار منوي محيط کار را بررسي کنيد. بسياري از گزینه‌هاي آن مانند ساير نرم‌افزارهاي تحت ويندوز است. هريك را که مي‌شناسيد مختصر توضيح دهيد.



شکل 1-2 : محیط کار سیستم

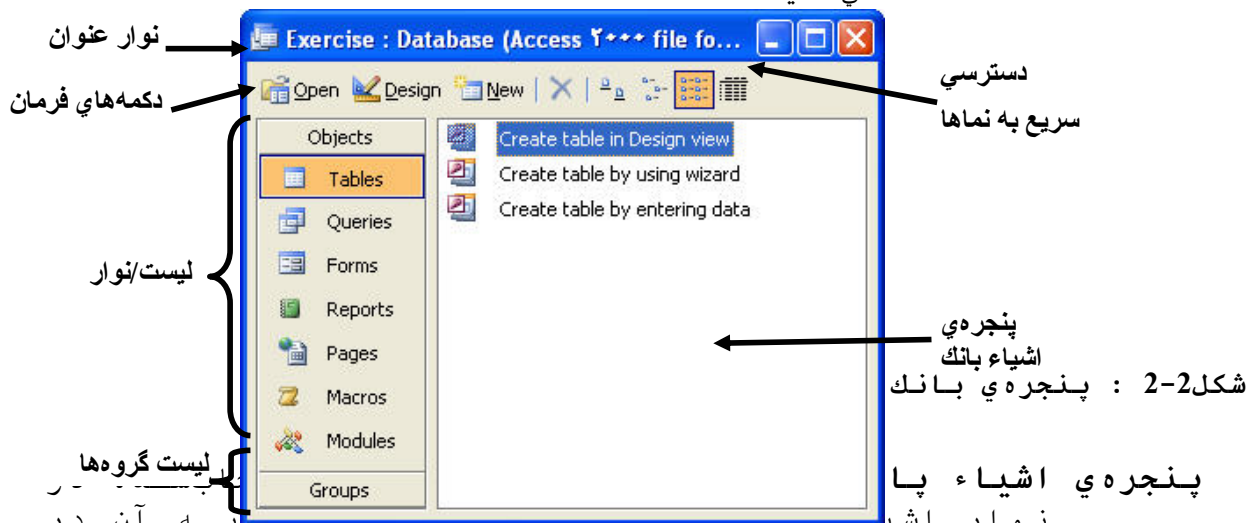
عملکرد نوار منو ، نوار ابزار و نوار وضعیت مشابه آن‌ها در سایر نرم افزارهای Office 2003 است و از توضیح مجدد آن‌ها خودداری می‌کنیم<sup>1</sup>.

**نوار اشیاء :** شامل بخش های مختلف پایگاه داده است . جدول‌ها ، فرم ها ، پرسوجوها و ...

**پنجره ی پایگاه داده :** این پنجره امکانات خاصی را فراتر

از محیط کار در اختیار قرار می‌دهد که در شکل 2-2

مشاهده می‌کنید :



نوار اشیاء پنجره ی اشیاء بانک داده می‌شود . این قسمت نمایش داده می‌شود .

پنجره ی اشیاء پایگاه داده شامل امکانات زیر است :

<sup>1</sup> - برای یادآوری به کتاب‌های بسته‌های نرم افزاری 1 و 2 مراجعه شود.

- **دکمه های فرمان :** با دکمه‌ی **Open** می‌توان با شیء انتخاب‌شده در لیست اشیاء ، کار کرد. با دکمه **Design** شیء موجود قابل‌اصلاح است و با دکمه‌ی **New** می‌توان شیء جدیدی را ایجاد کرد.

- **دسترسی سریع به نماها :** با استفاده از نماها ، چگونگی نمایش اشیاء در پنجره‌ی اشیاء پایگاه داده (پانل راست لیست‌اشیاء) تعیین می‌شود.

- **نوار عنوان :** نام و قالب فایل پایگاه داده را نمایش می‌دهد.

- **لیست اشیاء :** انواع اشیاء قابل تعریف در پایگاه داده هستند . با انتخاب هر یک ، اشیاء ایجادشده از آن نوع در پنجره‌ی اشیاء پایگاه داده (سمت راست پانل) نشان داده می‌شوند و می‌توان از آن نوع شیء را ایجاد ، ویرایش و یا حذف نمود.

- **لیست گروه‌ها :** می‌توان انواع مختلف اشیاء پایگاه داده را در این قسمت دسته‌بندی کرد. به این ترتیب جدول‌ها ، فرم‌ها و پرس‌وجوها و ... را می‌توان با هم در یک گروه مشاهده کرد. این گروه مانند یک دسترسی سریع به آن اشیاء عمل می‌کند.

### 3-2- انواع اشیاء و اجزای پایگاه داده

پایگاه داده **Access** از اشیاء مختلفی تشکیل شده است ، که عبارتند از جدول‌ها ، پرس‌وجوها ، فرم‌ها ، گزارش‌ها ، ماکروها ، ماجول‌ها و صفحات دسترسی به داده. کارهای متفاوتی را با هر یک از این انواع انجام می‌دهیم ، مانند : ذخیره‌سازی ، نمایش و چاپ داده‌ها ، نگهداری برنامه و ...

در ابتدا ممکن است از تعداد کمی از آن‌ها استفاده کنید. مثلاً از جدول<sup>1</sup> شروع می‌کنید که برای ذخیره داده هاست ، از فرم‌ها<sup>2</sup> برای ویرایش داده‌های روی صفحه و از گزارش‌ها<sup>3</sup> برای چاپ داده‌ها و همچنین از پرس‌وجوها<sup>4</sup> برای انتخاب و ترکیب داده‌ها استفاده می‌کنید. بعد از آن ، ممکن است از ماکروها<sup>5</sup> و ماجول‌ها<sup>6</sup> برای نگهداری برنامه‌ها استفاده کنید و یا در پروژه‌هایی که برای ویرایش و دسترسی داده‌ها از مرورگر وب<sup>7</sup> استفاده می‌شود ، از صفحات دسترسی به داده<sup>8</sup> استفاده کنید.

در این بخش این انواع اشیاء را با کمک بانک اطلاعاتی نمونه با نام **NorthWind.mdb** که در **Access** وجود دارد ، بررسی

---

<sup>1</sup> - Table

<sup>2</sup> - Forms

<sup>3</sup> - Reports

<sup>4</sup> - Query


<sup>5</sup> - Macro

<sup>6</sup> - Module

<sup>7</sup> - Web Browser

<sup>8</sup> - Data Access Page

می‌کنیم. برای آن که بتوانیم از این بانک استفاده کنیم باید فایل مربوطه را در محیط Access باز کنیم. یکی از این روشها عبارت است از :

برای استفاده از مثال موجود در نرم افزار ، از منوی Help-> Sample Databases... گزینه‌ی Northwind Sample Database را انتخاب کنید. در صورت پدیدار شدن پنجره‌ی خوشامدگویی آن را ببندید. می‌توانید با انتخاب گزینه‌ی Don't show this screen again در صفحه‌ی خوشامدگویی آن را برای همیشه غیرفعال کنید. پنجره‌ی شکل 2-3 را نیز با دکمه‌ی  ببندید تا در محیط کار Access قرار گیرید.

**نکته :**  
برای استفاده از مثال موجود در نرم افزار باید آن را نصب کرده باشید. با بازکردن برای اولین بار ، در صورتی که مثال نصب نشده باشد، پیغامی مبنی بر نصب ظاهر می‌شود و می‌توانید با آن مثال را فعال کنید.



شکل 2-3 : پنجره‌ی شروع کار مثال Northwind.mdb

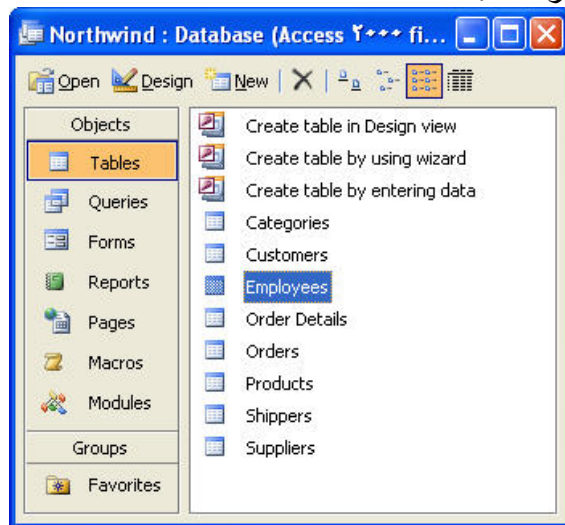
### 1-3-2- جدول‌ها برای ذخیره داده‌ها :

جدول ، جایی است که می‌توان داده‌ها را ذخیره کرد. یک جدول از رکوردهایی با قالب مشابه تشکیل شده است. یک پایگاه داده می‌تواند جدول‌های متعددی داشته باشد. مثلا پایگاه داده یک فروشگاه کتاب می‌تواند شامل جدول کتاب : (عنوان ، انتشارات ، نویسنده ، قیمت وسایر اطلاعات یک کتاب است) ، جدول فروشندگان : (نام شرکت ، نشانی ، تخفیف‌ها و سایر اطلاعات هر فروشنده) و جدول مشتریان فروشگاه : (نام ، آدرس و دیگر اطلاعات یک فرد) تشکیل شده باشد. وقتی جدول‌ها ایجاد شد و داده‌ها در آن ثبت شدند ، می‌توانید داده‌ها را مرتب کنید ، شرط‌های خاص بر آن اعمال

و رکوردهای دارای آن شروط را مشاهده یا چاپ کنید ، داده/داده‌های با شرط خاص را جستجو کرده، آن‌ها را پیدا کنید.

### تمرین 2-1 :

در بانک Northwind.mdb از نوار اشیاء ، گزینه‌ی Tables را انتخاب کنید. جدول‌های ایجادشده مطابق شکل زیر است.



### کنجکاو ی :

روی هر جدول دوبار کلیک کنید. به نظر شما هر یک چه داده‌هایی را نگه می‌دارد؟

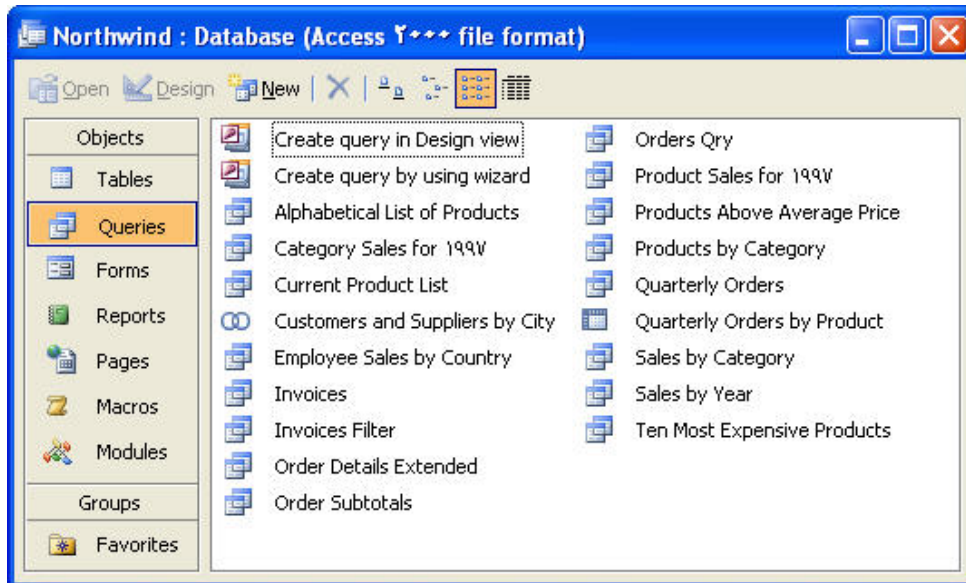
در فصل اول در خصوص یک طراحی خوب صحبت کردیم و این‌که چگونه از موجودیت‌ها و ارتباطات آن‌ها به تعریف جدول‌ها می‌رسیم. در فصل‌های بعد با توجه به محیط عملیاتی مدرسه‌ای که طراحی کرده‌ایم ، کار با نرم افزار را دنبال می‌کنیم.

### 2-3-2- پرسوجوها برای انتخاب داده‌ها :

یک پرسوجو برای جستجو ، مشاهده و ویرایش (به‌هنگام‌سازی) داده‌های موجود در جدول به‌کار می‌رود. انواع مختلفی از پرسوجو وجود دارد. ساده‌ترین و معمول‌ترین نوع پرسوجو انتخاب داده‌ها (Select) از جدول است ، مثلاً رکوردهایی که می‌خواهید در یک گزارش نمایش داده شوند. با توجه به مثال فروشگاه کتاب ، می‌توان پرسوجویی ایجاد کرد که مشخصات تمام مشتریانی که در شهر مفروض X زندگی می‌کنند یا شماره تلفنی از آن‌ها ندارید را نشان دهد.

## تمرین 2-2 :

در بانک اطلاعاتی NorthWind ، تعدادی از پرس‌وجوهای ایجادشده را باز کنید و آن‌ها را مشاهده کنید.



می‌توان پرس‌وجوهای ایجاد کرد که از ترکیب اطلاعات چند جدول استفاده کند. مثلاً در مثال فروشگاه کتاب که اطلاعات مشتریان در جدول مشتریان و اطلاعات تامین‌کنندگان در جدول فروشندگان است، می‌توان پرس‌وجویی ایجاد کرد که اطلاعات افراد بانک را نمایش دهد، این پرس‌وجو، اطلاعات هر دو جدول را هم‌زمان نشان می‌دهد. علاوه بر آن می‌توان پرس‌وجویی ایجاد کرد که فیلدهای محاسباتی<sup>1</sup> مانند تعداد کل، جمع کل، متوسط مقادیر و... را نمایش دهد. مثلاً تعداد کل مشتریانی که به فروشگاه مراجعه کرده‌اند.

از انواع دیگر پرس‌وجوها، پرس‌وجوی عملیاتی<sup>2</sup> است یعنی ایجاد رکوردهای جدید یا ویرایش و حذف رکوردها را انجام دهد. مثلاً پرس‌وجویی که رکوردهای یک جدول را انتخاب کند و در جدول دیگری کپی نماید، یا تغییری را بر تمام رکوردهای انتخاب‌شده اعمال کند، یا رکوردهایی را حذف نماید. مشتریانی که طی دو سال خریدی نداشته‌اند را حذف نماید. پرس‌وجوها ابزاری مناسب برای خارج کردن اطلاعات مفید از جدول‌ها هستند و ممکن است در یک پایگاه داده، تعداد زیادی پرس‌وجو ایجاد کنید.

### کنجکاو :

در بانک Northwind :

<sup>1</sup> - Calculated Field

<sup>2</sup> - Action Query

(1) هر یک از این پرس‌وجوها چه کاری انجام می‌دهند؟ چه نوعی هستند (انتخاب داده یا عملیاتی)؟

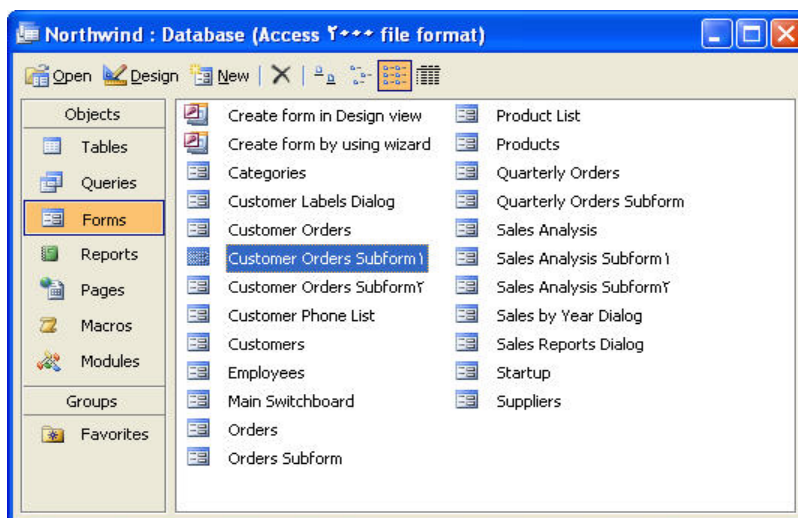
(2) محیط عملیاتی مدرسه را در نظر بگیرید. به نظر شما چه پرس‌وجوهایی لازم است؟

### 2-3-3- فرم ها براي ويرايش و نمايش داده ها :

ديديم با پرسوجوي عملياتي مي توانيم اطلاعاتي را در جدول ها وارد كنيم. در محيط Access پس از باز كردن جدول مي توانيم داده ها را به آن وارد كنيم. راه ديگر ورود اطلاعات به جدول ها ، استفاده از فرم است. با كمك فرم مي توان ورود اطلاعات را آسان تر كرد. داده هاي يك يا چند جدول در صفحه ي فرم نمايش داده مي شود و مي توان با امكانات فرم ، اطلاعات نمايش يافته را ويرايش يا حذف كرد و يا ركورد جديدي را ايجاد نمود. طرح هاي مختلفي براي نمايش اطلاعات در فرم وجود دارد كه در زمان ايجاد يك فرم خواهيم ديد. مثلا قالب جدولي<sup>1</sup> نمايش اطلاعات در فرم ، شبيه نماي جدولي است. مي توانيم با اشكال گرافيكي مثل مستطيل و خطوط ، اطلاعات را روي فرم گروه بندي كنيم . همچنين مي توانيم از ليست هاي بازشونده<sup>2</sup> ، دكمه هاي انتخاب<sup>3</sup> و ديگر انواع كنترل هاي روي صفحه براي ورود و ويرايش آسان تر داده ها استفاده كرد.

#### تمرين 2-3 :

در بانك Northwind.mdb فرم هاي ايجاد شده را مشاهده كنيد.



#### كنجاوي :

تحقيق كنيد هر يك از فرم هاي بالا به چه منظوري ايجاد شده اند؟ از چه كنترل هايي در فرم ها براي ورود و ويرايش آسان تر داده ها استفاده شده است؟ راهنمايي : با انواع كنترل هاي برنامه نويسي در درس برنامه سازي آشنا شده ايد مثل انواع List ها ، انواع Button ها ، انواع OptionBox ها ، CheckBox ها و ....

<sup>1</sup> - Table Layout

<sup>2</sup> - Pull-Down Lists

<sup>3</sup> - Radio/Option Buttons



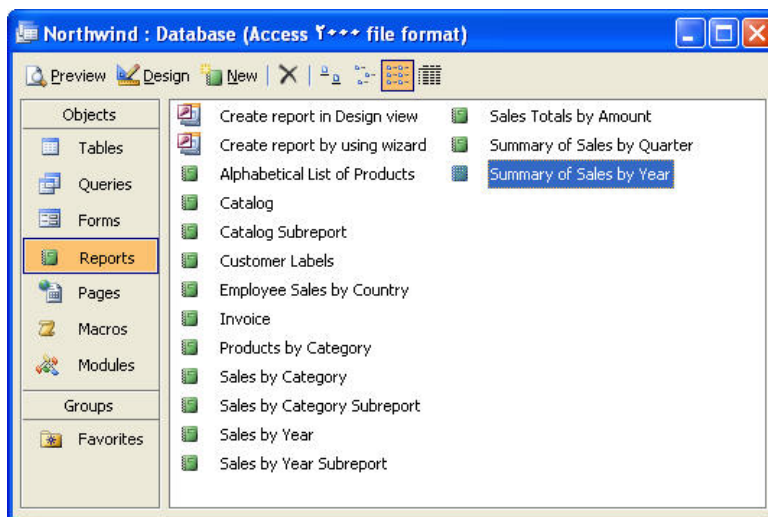
### 2-3-4- گزارشها برای چاپ داده ها :

فرمها عموماً برای ورود اطلاعات و گزارشها برای چاپ اطلاعات استفاده میشوند. گزارشها مانند فرمها اطلاعات را نشان میدهند و میتوان قالب مشخصی را برای نمایش اطلاعات انتخاب کرد. معمولاً گزارشها بر اساس پرسوجوها ایجاد میشوند و پرسوجو اطلاعات موردنظر برای نمایش گزارش را انتخاب میکند. در طراحی گزارش (Report Design) میتوان ترتیب نمایش رکوردها، محلی که فیلدها نمایش داده میشوند، نوع قلم نمایش و اندازهی آن، خطوط، فضای خالی و غیره را مشخص کرد. علاوه بر آن میتوان تعیین کرد رکوردها بر روی کاغذ چاپ شوند، یا روی پاکت نامه، یا به صورت برجسته و یا انواع دیگر چاپی.

Access ابزار جادویی (Wizard) دارد که میتوان گزارشهای ساده را به آسانی و با سرعت ایجاد کرد و یا نمودارها (Chart) را بر اساس دادههای موردنظر به سادگی ایجاد و چاپ کرد.

### تمرین 2-4 :

در بانک Northwind.mdb گزارشهای ایجاد شده را مشاهده کنید.



### کنجکاو ی :

- 1) تحقیق کنید چه شباهتهایی در این گزارشات مشاهده میشود؟
- 2) امکانات در دسترس در گزارشات در زمان اجرا و نمایش چیست؟ راهنمایی: امکاناتی مثل چاپ، جستجو، حرکت بین صفحات و ....

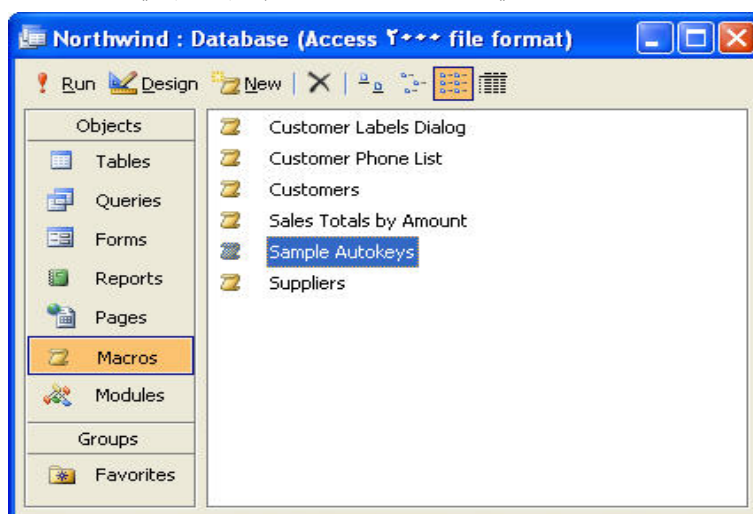
### 2-3-5- ماکروها برای کنترل کلیدها :

Access شامل دو امکان برنامه نویسی مجزا است : ماکروها (macro) و محیط شبیه VisualBasic (VBA). ماکروها، کد

برنامه‌ی فرمان‌های مورد استفاده در محیط Access هستند. می‌توان با ماکرو به‌جای انجام کارها به‌صورت دستی و با کمک کلیدها، عملیات را خودکار کرد. به‌طور مثال، می‌توانید ماکرویی ایجاد کنید زمانی‌که جدولی باز شود نشانگر در آخرین رکورد قرار گیرد بدون آن که پس از باز شدن جدول، از کلیدها یا دکمه‌هایی استفاده کنید. زمانی‌که یک ماکرو ایجاد شد، برای اجرا در زمان مناسب می‌توانید دکمه‌ای بر روی فرم قرار دهید تا با کلیک روی آن‌ها، آن ماکرو اجرا شود. همچنین می‌توانید تعیین کنید زمانی‌که نشانگر یا ماوس روی یک فیلد قرار گرفت، به‌طور اتوماتیک ماکرویی اجرا شود. نیازی نیست برنامه‌نویس باشید تا ماکرویی را ایجاد کنید چراکه محیط کار کمک می‌کند تا به سادگی ماکرو ایجاد شود.

### تمرین 2-5 :

در بانک Northwind.mdb ماکروهای ایجاد شده را مشاهده کنید.  
\* دقت کنید که این ماکروها پیچیده به نظر می‌رسند و البته کار بسیاری هم انجام می‌دهند.



### 2-3-6- ماجول‌ها برای نوشتن برنامه‌های مورد نظر:

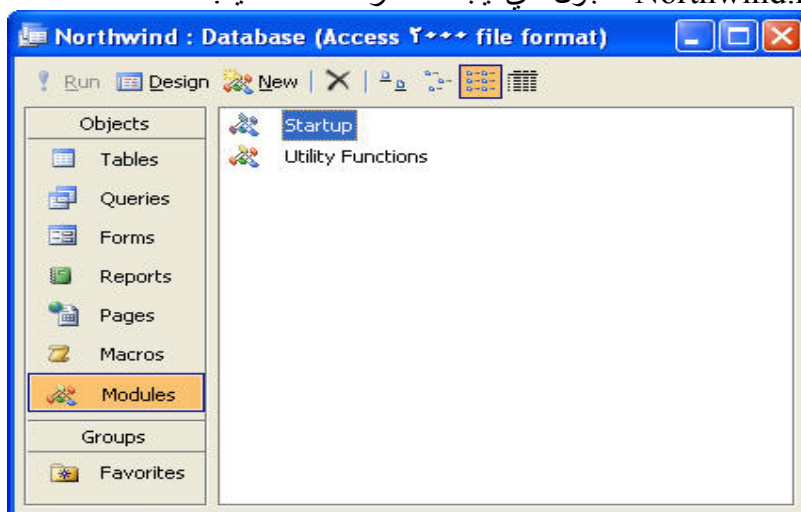
ماجول، بخش اصلی برنامه‌نویسی است. ماجول اصطلاح دیگری برای برنامه‌های VBA<sup>1</sup> است. VBA زبان برنامه‌نویسی است که بر اساس زبان Basic ایجاد شده است. ماکروها برای ذخیره‌ی کلیدهای فرمان و یا پاک کردن داده‌های وارد شده در یک فیلد مناسب هستند ولی زمانی‌که عملیات و دستورات پیچیده‌تری نیاز باشد، باید برنامه‌نویسی کرد. مثلاً فرض کنید سفارشات کتاب را به‌وسیله‌ی پست الکترونیک دریافت کرده‌اید، آنگاه می‌توانید برنامه‌ای بنویسید که پیام‌ها را از صندوق پست خوانده، رکوردهایی را به جدول خاصی در بانک وارد نماید. با این کار دیگر نیازی به کپی اطلاعات به‌وسیله‌ی یک اپراتور نیست. همچنین می‌توان برنامه‌ای نوشت که با کلیک

<sup>1</sup> - Visual Basic for Application، در کتاب برنامه‌سازی 3 با VBA آشنا می‌شوید.

روي دکمه‌ي ارسال ، لیست اقلام رسالي و برچسب نامه را چاپ کرده ، موجودي انبار را نیز کسر نماید و يك پیغام نیز به وسیله‌ي پست الکترونیک به مشتري ارسال کند.

### تمرین 2-6 :

در بانک Northwind.mdb ماجول‌هاي ایجاد شده را مشاهده کنید.



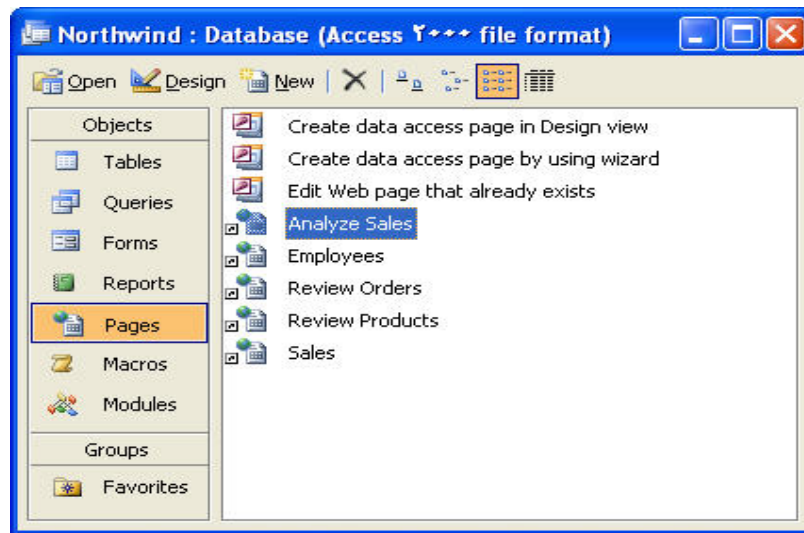
### 7-3-2- صفحات براي ويرایش داده‌ها در مرورگر وب :

فرم‌ها براي ورود و ويرایش داده به کار مي‌روند ولي براي نمايش آن‌ها بايد محيط Access اجرا شود. بدین معنی که باید Access روی کامپیوتر نصب شده باشد. اگر نخواهید از محیط Access استفاده کنید باید از صفحه‌ها (Page) استفاده کنید که فرم‌ها را در بر می‌گیرند. هر فردي مي‌تواند با مرورگر وب مثل IE (Internet Explorer) صفحه‌ي شامل فرم را باز کند و مانند فرم ، رکوردهاي جدول را ويرایش ، اضافه یا حذف نماید. اگر در اینترنت براي جستجو یا عضویت ، اطلاعاتي را وارد کرده باشید ، صفحه‌ي شامل فرم را دیده‌اید<sup>1</sup>. این‌گونه صفحه‌ها عموماً به بانک اطلاعاتي متصل شده‌اند و Access مي‌تواند يکي از آن‌ها باشد.

<sup>1</sup> - در کتاب بسته‌هاي نرم‌افزاري 3 ، جلد دوم ، با چگونگي ایجاد فرم و کار کردن با آن در صفحات وب آشنا مي‌شوید.

## تمرین 2-7 :

در بانک Northwind.mdb صفحات وب ایجاد شده را مشاهده کنید.



## 2-4- ایجاد يك پایگاه داده در نرم افزار Access

برای آنکه بتوانیم اشیاء بانک اطلاعاتی را ایجاد کنیم و در واقع سیستم بانک اطلاعاتی طراحی شده خود را پیاده‌سازی نماییم باید یک بانک پایگاه داده با پسوند .mdb داشته باشیم. این فایل واقعیت فیزیکی جدولها، فرمها، پرسوجوها، گزارشها و سایر اشیاء بانک را در بر می‌گیرد. برای ایجاد فایل بانک می‌توانیم از ویزارد استفاده کنیم. با ویزارد می‌توانیم براساس یکی از بانک‌های اطلاعاتی از پیش تعریف شده، بانک اطلاعاتی دخواه خود را همراه با جدولها، گزارشها و فرم‌های موردنیاز بسازیم. اما از آنجاییکه ممکن است بانک اطلاعاتی ما با هیچیک از بانک‌های اطلاعاتی از پیش ساخته شده موجود در Access سازگار نباشد، این روش مناسب نیست. بنابراین ابتدا یک بانک اطلاعاتی خالی ایجاد می‌کنیم و سپس جدولها، فرمها، گزارشها، پرسوجوها را به آن اضافه می‌کنیم.

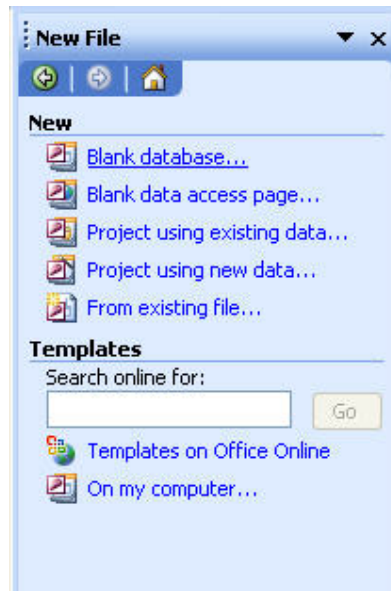
## تمرین 2-8 :

بانک اطلاعاتی DBProj.mdb را بدون کمک ویزارد ایجاد کنید.

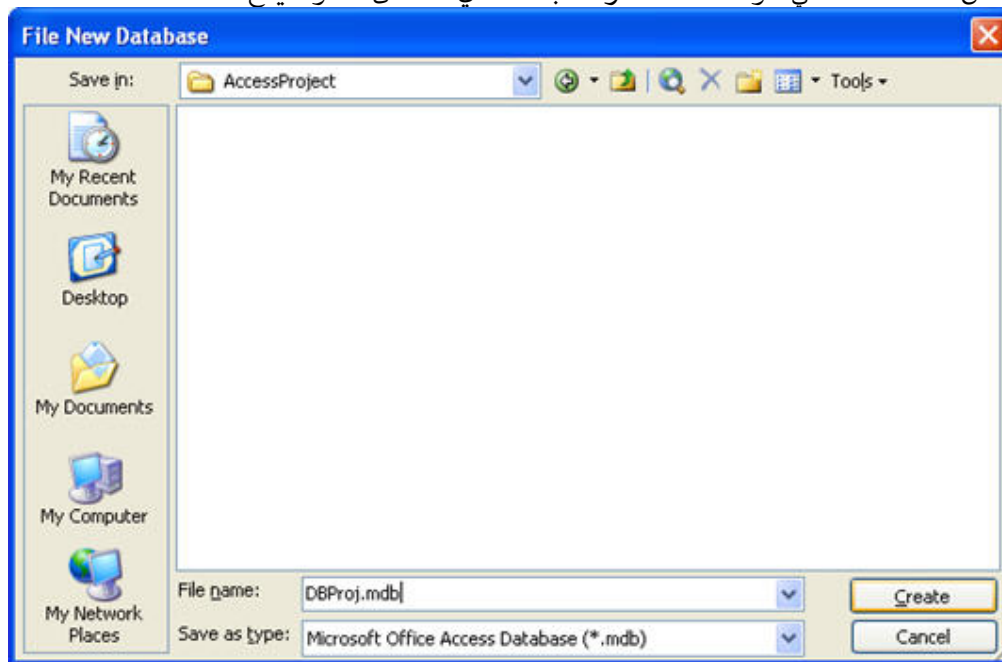
مراحل ایجاد به صورت زیر است :

- پس از ورود به Access از منوی File گزینه New را انتخاب کنید.
- روی گزینه Blank Access database از پانل ایجاد مطابق شکل زیر کلیک کنید.

ادامه ی تمرین 2-8 :



- Access نام و محل ذخیره سازی فایل را می پرسد. در قسمت File Name ، نام بانک اطلاعاتی را DBProj تایپ کرده ، سپس دکمه ی Create را کلیک نمایید. همیشه محل ذخیره ی مناسبی را برای فایل در نظر بگیرید. پس از ایجاد بانک ، پنجره ی بانک نشان داده می شود که در ابتدای فصل توضیح داده شده است.



## 2-5- بستن پایگاه داده

پایگاه داده‌ی ایجاد شده از طریق منوی File ، گزینه‌ی Close بسته می‌شود ولی توجه کنید که هنوز محیط کار Access فعال است ، اگرچه یک سری از قابلیت‌های محیط کار Access در دسترس هستند.

### کنجکاو ی :

چه قابلیت‌هایی در محیط کار Access در صورت فعال نبودن یک بانک داده در دسترس است؟

## 2-6- باز کردن پایگاه داده‌ی موجود

مانند بازکردن فایل در سایر نرم افزارهای تحت ویندوز است. مجدداً یادآوری می‌کنیم که از منوی File ، گزینه‌ی Open را انتخاب کنید. از پنجره‌ی باز شده ، نام و محل فایل ذخیره شده را انتخاب نمایید و دکمه‌ی Open را کلیک کنید. Access پایگاه داده‌ی موردنظر را باز می‌کند و پنجره‌ی بانک نشان داده می‌شود.

## خلاصه ي فصل

پايگاہ داده Access يکي از ابزارهاي مديريت داده است. با افزايش حجم داده‌ها بايد بتوانيم آن‌ها را تجزيه و تحليل کنيم و اطلاعات موردنياز را بازيافت نمايم. Access ، مانند ساير نرم افزارهاي تحت ويندوز داراي يك محيط کار است که ابزارها ، منوها و امکانات متعددي را در بر دارد. پنجره ي اصلي آن علاوه بر امکاناتي که از آن‌ها نام برده‌ايم ، شامل اشياء پايگاہ داده است. مهمترين اشياء بانک عبارتند از : جدول‌ها ، پرسوجوها ، گزارش‌ها و فرم‌ها.

جدول مهم ترين جزء بانک است که داده‌ها را نگه مي‌دارد. با پرسوجو مي‌توان داده‌هاي با شرط خاص را بازيافت کرد. از فرم‌ها براي نمايش ، ورود و ويرايش ساده‌تر داده‌ها استفاده مي‌شود ؛ از گزارش‌ها براي نمايش داده‌ها و چاپ آن‌ها در طرح مناسب استفاده مي‌شود.

قبل از ايجاد بانک و اشياء آن بايد ، بانک اطلاعاتي طراحي گردد. هدف از طراحي ، شناخت محيط عملياتي ، نياز کاربران و در نظر گرفتن تمام امکانات موردنياز پايگاہ داده قبل از پياده‌سازي است. بايد جدول‌ها ، فرم‌ها ، گزارش‌ها و غيره طراحي شوند با انجام طراحي ، در زمان پياده‌سازي در محيط Access ، با اشکالات اساسي مواجه نمي‌شويم که در غير اين‌صورت کار اصلاح اشکالات ايجاد شده به مراتب بسيار دشوارتر و وقت‌گيرتر خواهد بود.

## خود آزمايي

1. محيط کار Access شامل چه امکاناتي است ؟
2. امکانات پنجره ي اشياء بانک را نام برده و مختصر توضيح دهيد.
3. انواع اصلي اشياء پايگاہ داده را نام ببريد و توضيح دهيد.

### 3- فصل سوم : ایجاد و کار با جدول ها

در فصل دوم با مقدمات بانک اطلاعاتی و از آن جمله جدول آشنا شدید. در این فصل ایجاد جدول و عملیات روی آن را بررسی می‌کنیم .

در پایان این فصل انتظار می‌رود که هنرجو بتواند :

- به وسیله‌ی روش‌های گوناگون جدول‌ها را ایجاد کند.
- فیلدهای جدول را متناسب با طراحی انتخاب کند و خصوصیات فیلدها را تغییر دهد.
- به کمک Import Table و Link Table جدول را از بانک‌های دیگر موجود به بانک خود منتقل کند.
- ثبت ، درج ، ویرایش و حذف داده‌ها در جدول را انجام دهد.
- طراحی جدول را تغییر دهد.
- داده‌ها را جستجو کند ، با اعمال شرایطی داده‌ها را فیلتر نماید و یا مرتب کند.
- ارتباط بین جدول‌ها را در بانک تعریف کند.

در نرم افزار Access ، سه روش برای ایجاد جدول وجود دارد که عبارتند از:

- ایجاد جدول در نمای Design
- ایجاد جدول به وسیله‌ی Wizard
- ایجاد جدول با ورود اطلاعات ( در نمای Datasheet)

علاوه بر این سه روش می‌توان جدول را از جدول‌های موجود در پایگاه داده‌ی دیگر نیز ایجاد کرد که خود به دوسورت انجام می‌گیرد : Import Table و Link Table.



در فصل‌های قبل سیستم ثبت‌نام محیط‌عملیاتی مدرسه را طراحی کردیم و جدول‌ها ، فرم‌ها و گزارش‌های موردنیاز را به‌دست آوردیم. می‌خواهیم در این فصل جدول‌هایی با مشخصات زیر را در بانک اطلاعاتی ایجادشده‌ی DBProj.mdb ، وارد نماییم.

( محیط عملیاتی = مدرسه و سیستم = ثبت‌نام )

#### 1. Tbl\_Course دروس

Name	Type	Size
CourseCode	Text	5
CourseName	Text	20
CourseUnit	Byte	-
TeacherCode	Integer	-
ProgramHour	Text	50
ProgramClass	Text	50

#### 3. Tbl\_Major رشته تحصیلی

Name	Type	Size
MajorCode	Integer	-
MajorName	Text	20

#### 4. Tbl\_Register ثبت نام

Name	Type	Size
RegisterCode	Integer	-
StudentCode	Text	5
RegisterDate	Date/Time	-

#### 5. Tbl\_RegisterItem دروس ثبتنام شده

Name	Type	Size
RegisterCode	Integer	-
CourseCode	Text	5
Grade	Decimal	(Precision=5, Scale=2, DecimalPlace=2)

#### 6. Tbl\_Student دانش آموز

Name	Type	Size
StudentCode	Integer	-
StudentFName	Text	20
StudentLName	Text	30
StudentTel	Text	20
StudentAddress	Memo	-
Year	Integer	-
MajorCode	Integer	-

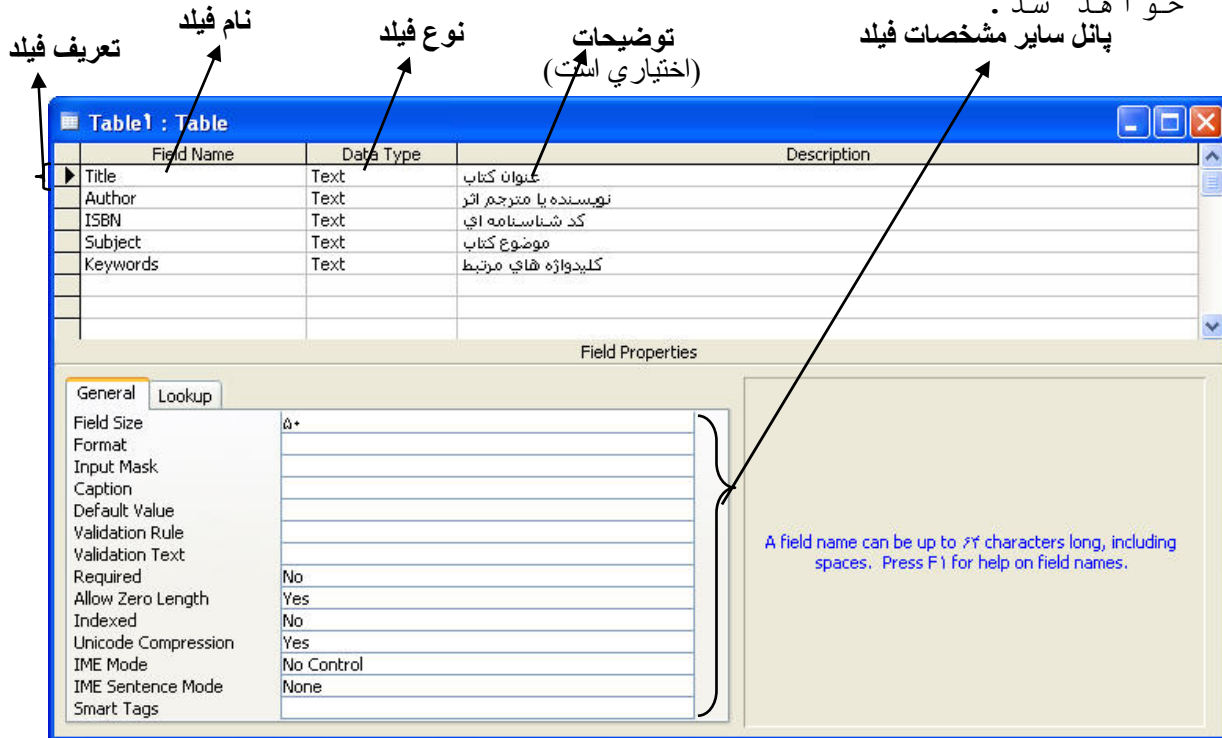
#### 7. Tbl\_Teacher معلم

Name	Type	Size
TeacherCode	Integer	2
TeacherFName	Text	20
TeacherLName	Text	30
TeacherTel	Text	20
TeacherAddress	Memo	-

### 3-1- ایجاد جدول در نمای Design

- در لیست اشیاء روی گزینه ی Tables کلیک کنید.
- دکمه ی New را انتخاب نمایید.
- از پنجره ی ایجاد جدول ، گزینه ی Design View را انتخاب کنید.

در پنجره‌ی طراحی بازشده‌ی جدول (شکل 1-3) ، اطلاعات لازم را وارد کنید که در ادامه به تفصیل توضیح داده خواهد شد.



شکل 1-3 : پنجره‌ی طراحی جدول برای جدول Books

### 1-1-3 اجزای پنجره‌ی طراحی جدول :

با توجه به پنجره‌ی طراحی جدول شکل 1-3 ، دو ستون نام فیلد و نوع فیلد برای تعریف فیلد ضروری هستند. هر فیلد در یک ردیف تعریف می‌شود. هر فیلد غیر از نام و نوع ، مشخصات دیگری نیز دارد که در پنجره‌ی پانل سایر مشخصات فیلد تنظیم می‌شوند.

❖ **نام فیلد :** نام فیلد بهتر است به گونه‌ای انتخاب شود که نشان‌دهنده‌ی محتوای آن باشد. مثلاً عنوان کتاب ، Title نامیده شود. بهتر است از حروف لاتین استفاده کنید . هرچند مجاز هستید که از کاراکتر Space در نام استفاده کنید ولی بهتر است از آن استفاده نکنید. اندازه نام نباید بیش از 64 کاراکتر باشد.

❖ **نوع فیلد :** نوع داده‌ای است که در فیلد وارد می‌شود. عبارت است از موارد زیر :

نوع فیلد	توضیحات
<b>Text</b>	نوع پیشفرض است. این نوع هر ترکیبی از حروف و ارقام است. حداکثر تا 255 کاراکتر در این نوع فیلد امکان پذیر است.
<b>Memo</b>	از نوع متن است که تا 64.000 کاراکتر را ذخیره می‌کند.
<b>Number</b>	فقط اعداد را ذخیره می‌کند. نوع صحیح در اندازه‌ی 16 بیتی و یا 32 بیتی، نوع ممیز اعشار یا سایر انواع عددی دیگر که در قسمت <b>General</b> در پانل پایین پنجره‌ی طراحی جدول (نمای <b>Design</b> ) تعیین می‌شوند. پانل پایین را در ادامه توضیح می‌دهیم.
<b>Date/Time</b>	تاریخ، زمان و یا ترکیبی از آن‌ها را نگه می‌دارد. به‌خاطر داشته باشید که این نوع به تاریخ لاتین حساس است و در صورتی که برای نگهداری تاریخ شمسی استفاده شود در برخی تاریخ‌ها مثلاً "1385/02/31" پیغام عدم اعتبار خواهد داد. بنابراین برای نگهداری تاریخ شمسی بهتر است از این نوع استفاده نشود بلکه از <b>Text</b> استفاده کنید.
<b>Currency</b>	مقادیر پولی را نگه می‌دارد و می‌تواند به‌طور خودکار علامت \$ داشته و محل‌های علامت جداساز هزارتایی "کاما" را نگه دارد.
<b>Auto Number</b>	در هنگام ایجاد رکورد جدید، <b>Access</b> به‌طور خودکار عدد صحیح منحصربه‌فرد را برای آن جدول، تولید و به این فیلد منسوب می‌کند. در قسمت <b>General</b> در پایین پنجره‌ی طراحی جدول در ردیف <b>New Values</b> لیست بازشونده‌ای شامل دو مقدار وجود دارد: <b>Increment</b> که عدد منحصربه‌فرد به‌طور سریال تولید می‌شود و <b>Random</b> که هر عددی به‌طور تصادفی و منحصربه‌فرد ایجاد می‌شود. از آن جایی‌که هر جدول باید یک فیلد کلید اصلی داشته باشد در صورتی‌که از فیلدهای جدول یعنی صفات موجودیت مرتبط نتوانیم کلید را مشخص کنیم، ایجاد فیلد جدیدی از این نوع به‌عنوان فیلد کلید اصلی، می‌تواند مفید باشد.
<b>Yes/No</b>	این نوع می‌تواند مقادیر دوازده‌گانه <b>True/False</b> ، <b>Yes/No</b> ، <b>On/Off</b> و یا دیگر مقادیر را در بر گیرد.
<b>OLE Object</b> <sup>1</sup>	این نوع می‌تواند صدا، تصویر و یا دیگر اشیاء را دربر گیرد. مثل فایل <b>Word</b> و یا <b>Excel</b> که به‌وسیله‌ی برنامه‌های دیگر تولید می‌شوند.

<sup>1</sup> - Object Linking and Embedding

Hyperlink	
	<p>نوعی است که به آدرس وب سایت یا محل بانک دیگر وصل می‌شود. داده‌های این نوع شامل حداکثر چهار بخش است که به وسیله علامت # جدا می‌شوند :</p>
	<p><b>Displaytext#Address#Subaddress#Screentip</b>          به خاطر بسپارید که قسمت Address تنها بخشی است که حتما باید وارد شود. نمونه‌هایی از این نوع داده را می‌بینید :</p>
	<p>❖ لینک اینترنتی :</p>
	<p><b>HomePage#http://www.microsoft.com#</b>          صفحه‌ی اول سایت مایکروسافت که با عنوان "HomePage" روی مرورگر باز می‌شود.          ❖ لینک به پایگاه داده :</p>
	<p><b>Students##Form Student#Click to view and edit students information</b>          فرم Students که در پایگاه داده‌ی جاری قرار دارد. و پیغام Click... را نمایش می‌دهد.          ❖ لینک به پایگاه داده دیگر:</p>
	<p><b>#c:\Microsoft Office\Northwind.mdb#Form Customers</b>          فرم Customers که در پایگاه داده‌ی Northwind قرار دارد را نشان می‌دهد.          ❖ لینک به برنامه‌ی دیگر:</p>
	<p><b>#c:\windows\personal\student.ppt#13</b>          اسلاید 13 ارائه‌ی power point در آدرس c:\... را نشان می‌دهد.</p>
	<p><b>#c:\windows\personal\student.xls#Sheet1!A2</b>          سلول A2 را در Sheet1 صفحه‌گسترده‌ی Excel در آدرس c:\... را نشان می‌دهد.</p>

### 3-1-2- پانل سایر مشخصات فیلد : Field Properties

یک فیلد علاوه بر نام و نوع داده‌ای، مشخصه‌های دیگری نیز دارد که در این قسمت در اختیار قرار می‌گیرند. این مشخصه‌ها نحوه‌ی ذخیره‌سازی، دستکاری و نمایش مقدار داده‌ی ذخیره‌شده در آن فیلد را کنترل و تنظیم می‌کنند. با انتساب این مشخصه‌ها زمان زیادی در مرحله‌ی ساخت فرم، گزارش و اشیاء دیگر پایگاه داده صرفه‌جویی می‌شود یعنی زمان کمتری در ایجاد آن‌ها صرف خواهد شد. پنجره‌ی مشخصه‌های فیلد، بسته به نوع داده‌ای که برای آن فیلد در نظر گرفته‌اید تغییر می‌کند.

## برخی از مشخصه‌های این پنجره عبارت است از :

مشخصه	توضیحات
<b>Field Size</b>	اندازه‌ی نوع فیلد text را به تعداد مشخصی کاراکتر محدود می‌کند و یا نوع فیلد Number را به یک دامنه اعداد محدود می‌کند.
<b>Format</b>	قالب نمایش داده در نمای Datasheet است.
<b>Decimal places</b>	تعداد ارقام سمت راست نقطه اعشار را تعریف می‌کند. این مشخصه در نوع داده‌ی Number ، در صورتی‌که Format مقدار خالی و یا مقدار General باشد اثری ندارد و در سایر موارد می‌تواند Format تعیین شده را تغییر دهد.
<b>Input Mask</b>	قالبی است که تعیین می‌کند داده در زمان ورود اطلاعات به چه شکلی دریافت شود. مثلاً برای تلفن موبایل تعریف 0####\##### نمونه داده : 0912-1234567 را دریافت می‌کند. یعنی ابتدای آن صفر و پس از چهارمین رقم ، علامت - قرار گیرد و کلاً 11 رقم باشد.
<b>Caption</b>	برچسبی است که به‌جای نام فیلد ، در جدول‌ها ، فرم‌ها و گزارش‌ها نمایش داده می‌شود.
<b>Default Value</b>	مقدار پیش‌فرضی است که در زمان ایجاد یک رکورد جدید اگر مقداری به فیلد مربوطه اختصاص داده نشده باشد ، در آن ذخیره می‌شود.
<b>Validation Rule</b>	شرطی است که داده‌های وارد شده را محدود می‌کند. مثلاً این‌که داده‌ی وارد شده بیش از 10000 نباشد.
<b>Validation Text</b>	متنی است که در زمان عدم احراز شرایط Validation Rule به عنوان پیغام خطا یا اخطار نمایش داده می‌شود.
<b>Required</b>	مشخص می‌کند در زمان ورود داده ، آن فیلد نمی‌تواند خالی رها شود بلکه حتماً باید مقداری به آن نسبت داده شود.
<b>Allow Zero Length</b>	مشخص می‌کند آیا می‌توان مقدار خالی یعنی "" را به فیلد نسبت داد یا خیر.
<b>Indexed</b>	بازیافت داده از فیلد را تسریع می‌کند. تمام فیلدهای کلید اصلی به‌طور پیش‌فرض Indexed هستند. زمانی که مقدار آن yes انتخاب شود ، می‌توانیم تعیین کنیم مقدار تکراری در آن فیلد قرار گیرد یا خیر.

**کنجاوي :**  
 نحوه ي شرکت مقدار عددي در جملات محاسباتي با نوع Field Size متفاوت چگونه است؟  
 راهنمائي :  
 نتیجه ي 13/2 مي تواند به دوجواب برسد . برحسب آن که مقدارها Integer فرض شوند 6 و با فرض مقدارها به نوع نقطه اعشار 6/5 خواهد بود.

### مقادير مشخصه ي Format نوع داده ي اعداد و نوع داده ي تاريخ/زمان :

- انتسابات قالب عددي (Format Number) : خصوصيت Format نحوه ي نمايش داده را در صفحه نمايش Datasheet مشخص مي کند . اين مشخصه بر نحوه ي ذخيره شدن داده در جدول و يا بر نحوه ي شرکت مقدار عددي ذخيره شده ي آن در محاسبات تائيري ندارد .

#### مشخصه Format براي نوع داده Number :

عدد را به همان صورتی که وارد شده است ، نمايش مي دهد . قالب پيش فرض است .	<b>General</b>
از کما به عنوان جداساز هزارتايي استفاده مي کند و علامت ارز را نيز نشان مي دهد . اعداد منفي در پرانتز نمايش مي يابند و مقدار پيش فرض $Decimal\ Place=2$ است .	<b>Currency</b>
حداقل يك رقم را نشان مي دهد و مقدار پيش فرض $Decimal\ Place=2$ است . اعداد حول تعداد ارقام راست نقطه اعشار گرد <sup>1</sup> مي شوند . بانك از کما براي نمايش جداساز هزارتايي استفاده مي کند .	<b>Fixed</b>
عدد وارد شده را در 100 ضرب مي کند و علامت درصد را به انتهاي آن اضافه مي کند . مقدار پيش فرض $Decimal\ Place=2$ است .	<b>Percent</b>
اعداد را به صورت علمي توان 10 و با نماد E نشان مي دهد .	<b>Scientific</b>

<sup>1</sup> - Round

- انتسابات تاریخ / زمان (Date/Time Format) : قالب نمایش تاریخ و یا زمان را تغییر می‌دهد.

**مشخصه Format برای نوع داده Date/Time :**

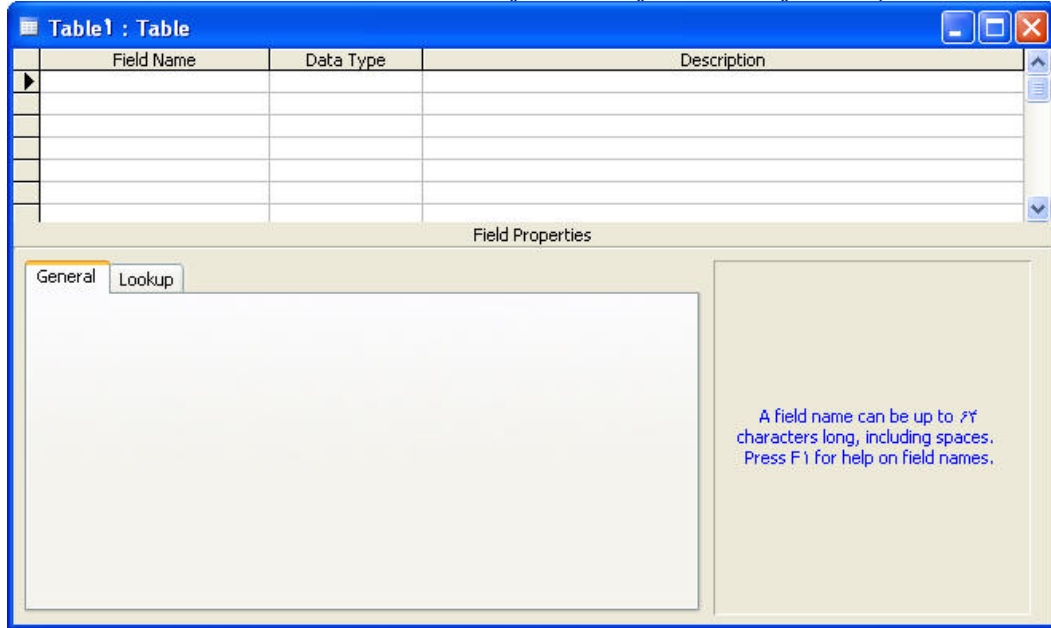
<b>General Date</b>	اگر مقدار فقط تاریخ بدون زمان باشد و یا اگر مقدار زمان بدون تاریخ باشد ، از آن استفاده می‌شود.
<b>Long Date</b>	اسامی روز و ماه به صورت متنی نوشته می‌شود. مثل Tuesday, July 4, 1995
<b>Medium Date</b>	نام ماه به صورت مخفف نشان داده می‌شود و نام روز حذف می‌شود. مثل : 04-Jul-95
<b>Short Date</b>	تاریخ به صورت رقم که با علامت "/" جدا شده اند نمایش داده می‌شود. مثل : 7/4/95
<b>Long Time</b>	زمان به صورت ساعت ، دقیقه و ثانیه که با علامت ":" جدا شده اند ، نمایش می‌یابد به همراه علامت AM برای صبح و PM برای بعد از ظهر. مثل : 6:30:15 PM
<b>Medium Time</b>	مشابه قالب Long Time است جز آنکه ثانیه نمایش نمی‌یابد مثل : 06:15 PM
<b>Short Time</b>	زمان به صورت 24 ساعته بدون ثانیه نمایش داده می‌شود. مثل : 18:30



### مثال 3-1 :

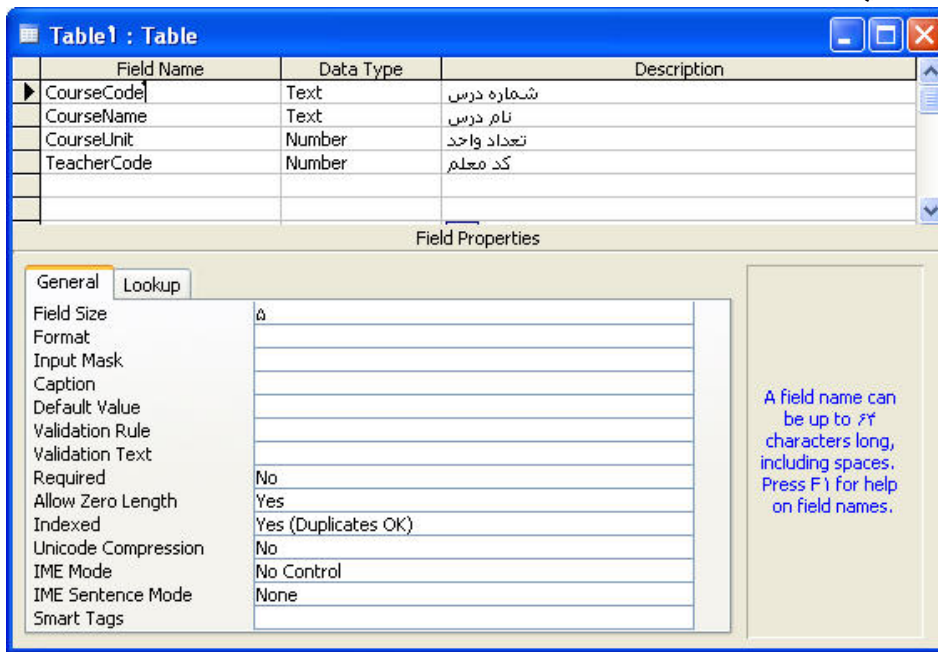
#### مراحل ایجاد جدول Tbl\_Course به وسیله Design :

- از لیست ایجاد جدول در پنجره‌ی اشیاء جدول ، Create table in Design view را انتخاب کنید.
- پنجره‌ی طراحی جدول خالی نمایش می‌یابد. مطابق شکل زیر :



### ادامه‌ی مثال 3-1 :

- در سلول نام فیلد ردیف اول CourseCode را تایپ کنید.
- با فشردن کلید [Tab] در ستون دوم نوع فیلد (Data type) ، نوع داده را انتخاب کنید که به‌طور پیش‌فرض text است. [با کلیدهای جهتی (Arrow Key) صفحه کلید ، انواع دیگر داده را می‌توانید انتخاب کنید]. همان‌طور که در فرض مثال می‌بینید نوع داده‌ی کد درس متنی و از نوع text انتخاب شده است.
- با کلید [Tab] در ستون توضیحات قرار بگیرید . هرچند الزامی به تکمیل آن نیست ولی بهتر است توضیحاتی در مورد آن فیلد به صورت فارسی در آن تایپ شود.
- با کلید [Tab] در سطر بعد قرار بگیرید . سایر فیلدها را به همین روش تعریف کنید.



- سایر مشخصات فیلدهای CourseName ، CourseUnit و TeacherCode را مطابق

#### CourseName:

شکل‌های زیر تنظیم کنید :

Field Size	۲۰
Format	
Input Mask	
Caption	
Default Value	
Validation Rule	
Validation Text	
Required	No
Allow Zero Length	Yes
Indexed	No
Unicode Compression	Yes
IME Mode	No Control
IME Sentence Mode	None
Smart Tags	

### ادامه‌ی مثال 3-1 :

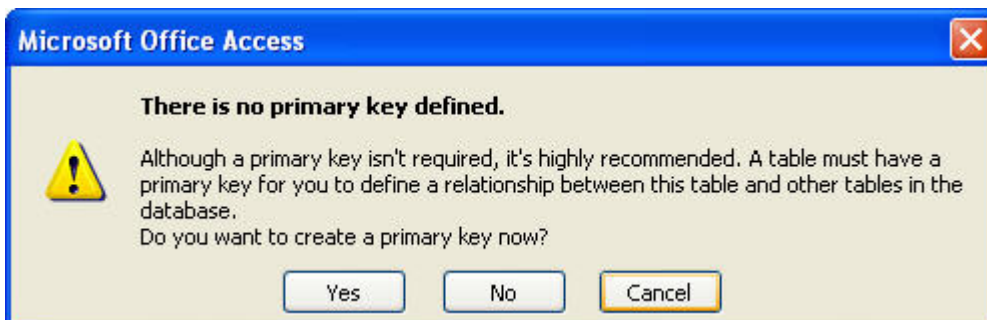
#### CourseUnit :

Field Size	Byte
Format	
Decimal Places	Auto
Input Mask	
Caption	
Default Value	*
Validation Rule	
Validation Text	
Required	No
Indexed	No
Smart Tags	


#### TeacherCode :

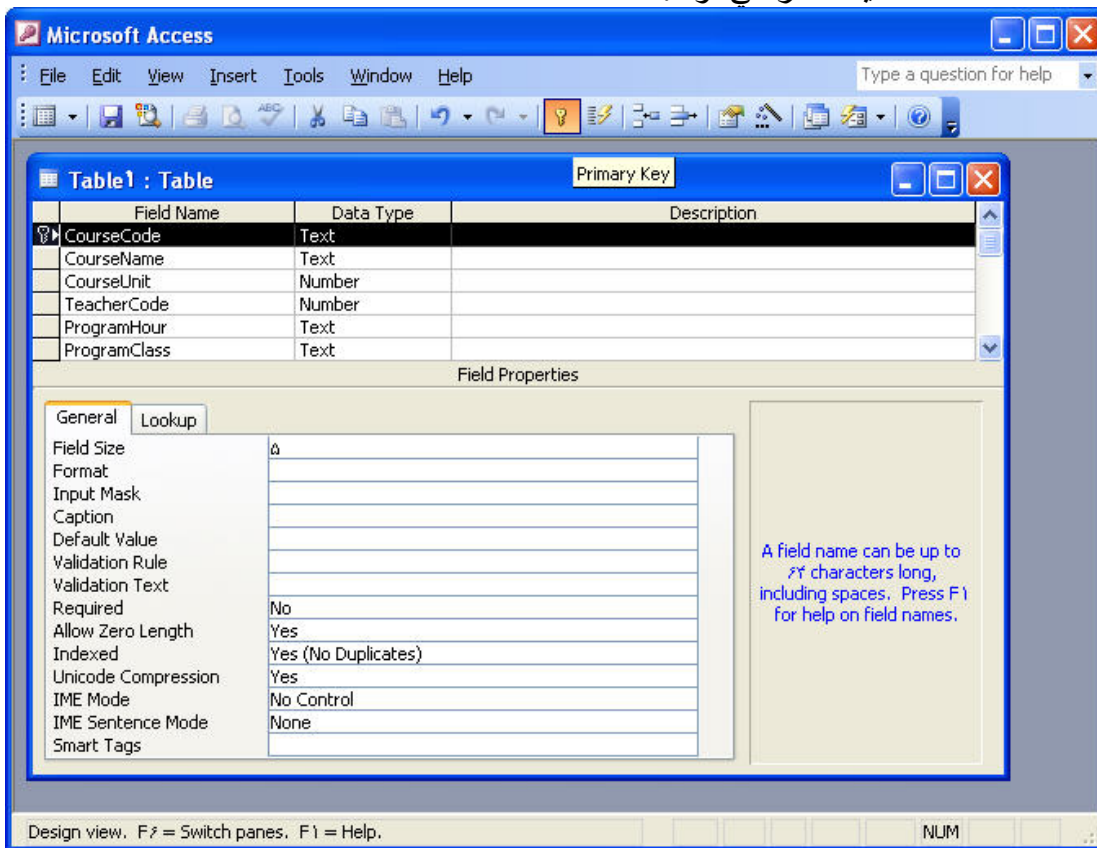
Field Size	Integer
Format	
Decimal Places	Auto
Input Mask	
Caption	
Default Value	*
Validation Rule	
Validation Text	
Required	No
Indexed	Yes (Duplicates OK)
Smart Tags	

- پس از تعریف تمام فیلدها ، دکمه‌ی Save از نوار ابزار یا گزینه‌ی Save از منوی File را انتخاب کنید تا پنجره‌ی ذخیره‌سازی جدول ظاهر شود.
- نام آن را Tbl\_Course قرار دهید و Ok را کلیک کنید. چون هنوز فیلد کلیداصلي را مشخص نکرده‌ایم ، پیغام زیر نمایش داده می‌شود.



### ادامه‌ی مثال 3-1 :

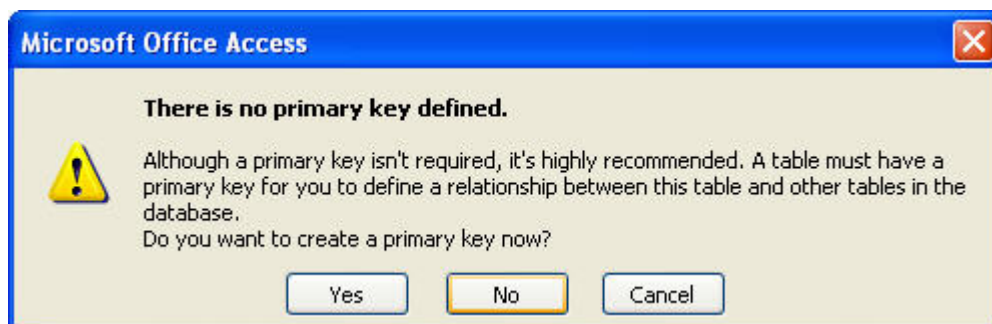
- در صورتی که جدول ، فیلد کلید اصلی تعریف شده‌ای نداشته باشد می‌توان از قابلیت Auto Number نرم افزار Access استفاده کرد. در این صورت با انتخاب دکمه‌ی Yes فیلد جدیدی به نام ID و از نوع Auto Number (خودشمارنده) ایجاد می‌شود. در اینجا کلید جدول ما در طراحی مشخص شده است بنابراین دکمه‌ی Cancel را انتخاب و کلید را به صورت زیر تعریف می‌کنیم.
- مطابق شکل زیر ، فیلد مورد نظر برای کلید را در پنجره‌ی طراحی جدول انتخاب کرده، از نوار ابزار دکمه‌ی علامت کلید  را کلیک کنید. در کنار نام فیلد علامت کلید ظاهر می‌شود.



- در این حال به هنگام ذخیره‌سازی پنجره‌ی نام جدول مجدداً ظاهر می‌شود. نام جدول را Tbl\_Course قرار دهید و آن را ذخیره نمایید. توجه کنید که پیغام قبلی ظاهر نمی‌گردد و جدول ذخیره می‌شود.

**کنجکاوې :**

گزینه یې No در پیغام زیر چگونه عمل می‌کند؟



**تمرین 3-1 :**

جدول Tbl\_RegisterItem را به این روش ایجاد کنید.

### **3-2- ایجاد جدول به وسیله ی ویزارد Table Wizard**

کاربر می‌تواند با استفاده از جدول‌ها و فیلدهای پایگاه داده‌ی از پیش‌تعریف‌شده و با پاسخ دادن به یک سری سوالات درباره داده‌ای که می‌خواهد ذخیره شود، جدول را ایجاد کند.

استفاده از ویزارد مزایا و معایبی دارد که از آن جمله عبارت است از :

جدول‌های پایگاه داده بدون درک طراحی و شناخت نوع فیلد، ایجاد می‌شوند، ولی کنترل کمی به وسیله‌ی کاربر می‌تواند در آن لحاظ شود.

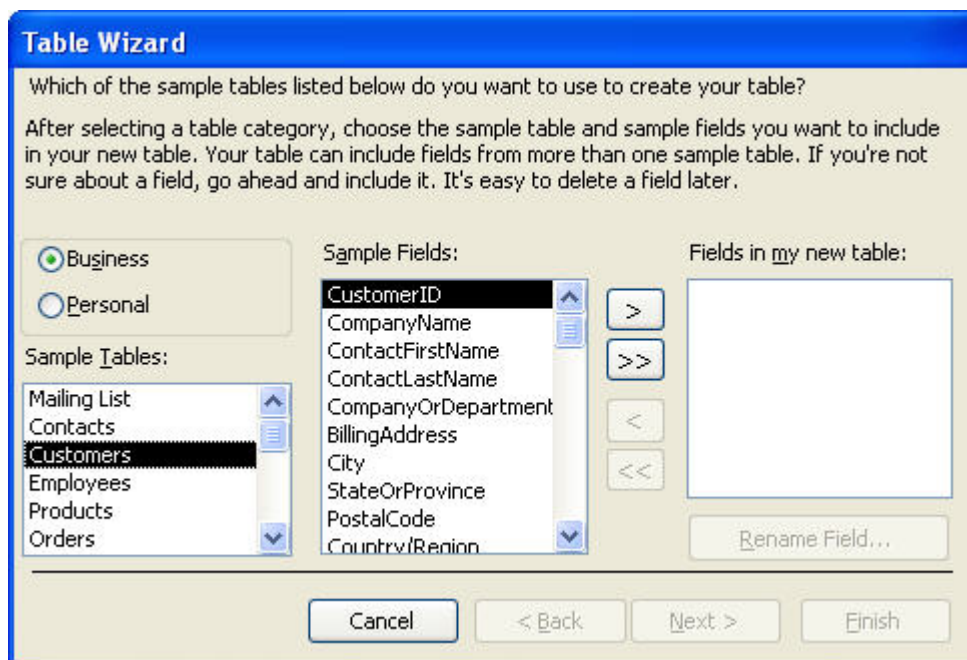
**کنجکاوې :**

مزایا و معایب استفاده از ویزارد را بررسی کنید؟

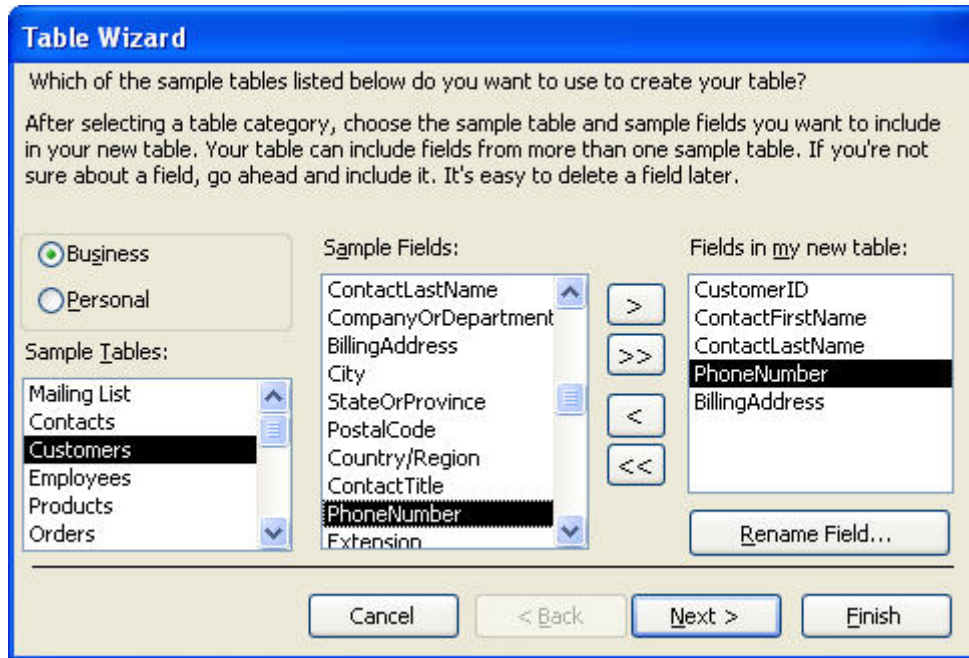
### مثال 3-2 :

مراحل ایجاد جدول Tbl\_Teacher به وسیله ی ویزارد :

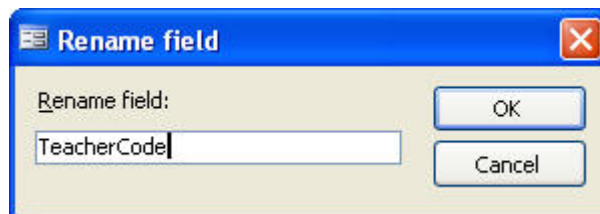
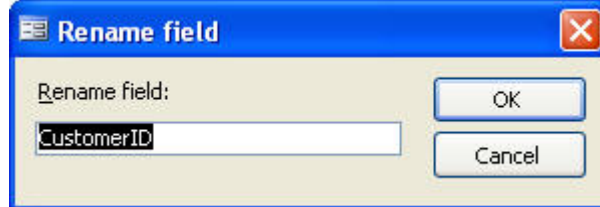
- از لیست ایجاد جدول در پنجره ی اشیاء جدول ، Create table by using wizard را انتخاب کنید.
- مطابق شکل ، از لیست Sample Table روی Customers کلیک کنید.



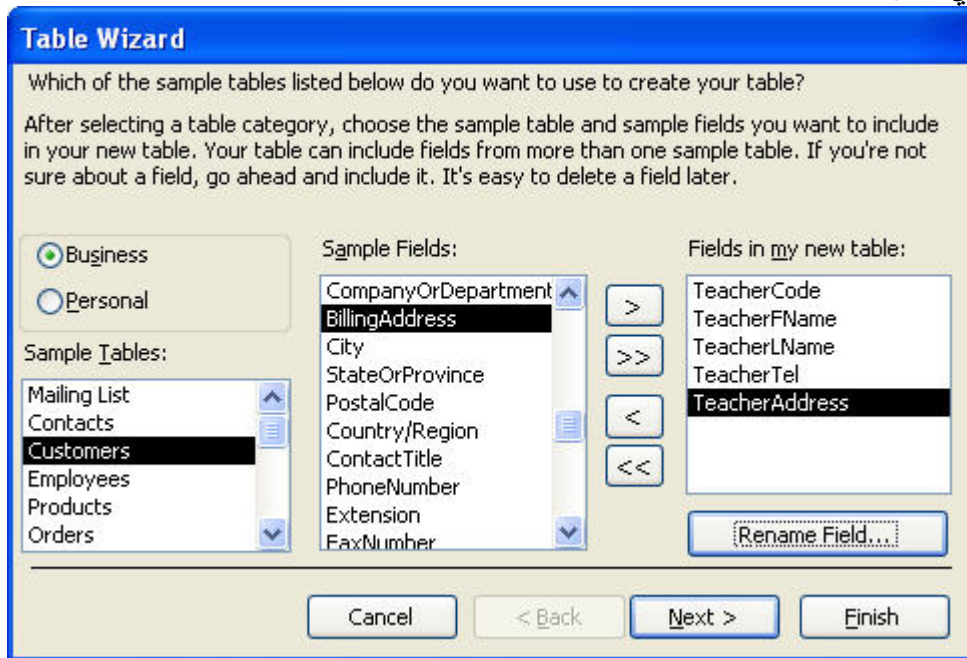
- از Sample Fields ، فیلد CustomerId را انتخاب و دکمه > را بزنید تا به لیست Fields in my new table اضافه گردد.
- برای باقی فیلدها نیز ، فیلد CustomerId را به جای TeacherCode و به همین ترتیب فیلدهای CustomerFirstName ، CustomerLastName ، BillingAddress و PhoneNumber را به جای فیلدهای TeacherFName ، TeacherLName ، TeacherTel و TeacherAddress اضافه کنید. مطابق شکل زیر :



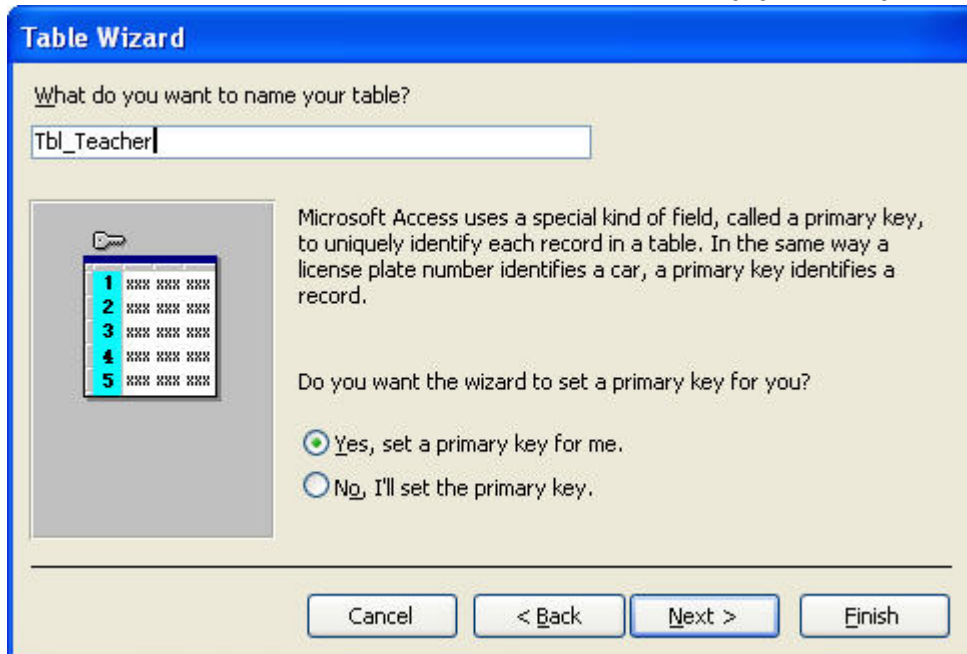
- روی CustomerId در قسمت Fields in my new table کلیک کرده و دکمه‌ی Rename Field... را کلیک کنید.
- در پنجره‌ی باز شده مانند شکل زیر آن را به TeacherCode تغییر نام دهید.



- به همین صورت برای سایر فیلدها عمل کنید و نام هر یک از فیلدها را متناسب تغییر دهید. مانند شکل زیر :

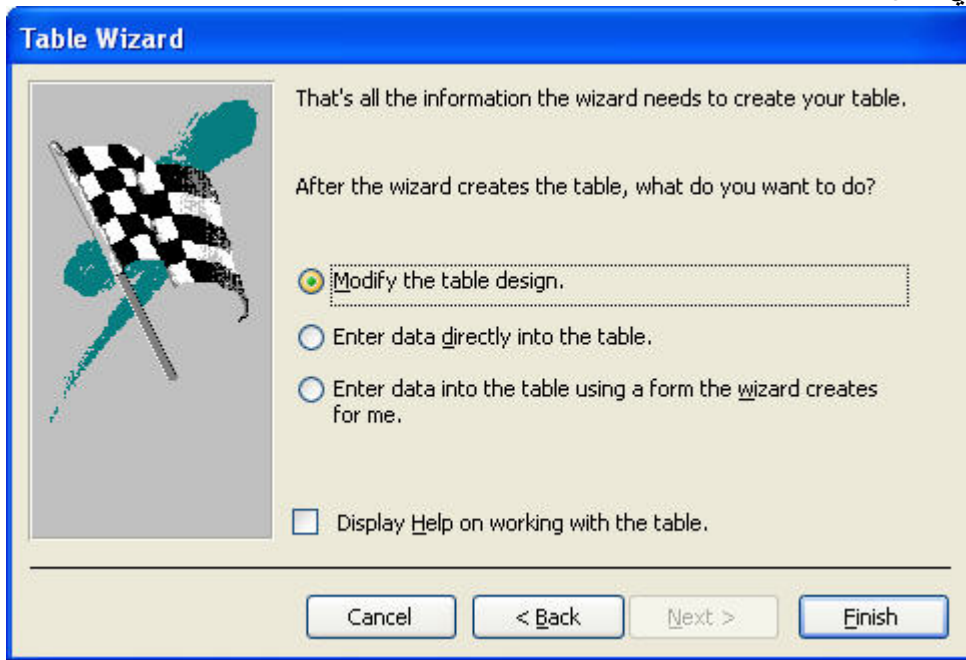


- روی دکمه Next کلیک کنید و نام جدول موردتقاضا را Tbl\_Teacher بگذارید. شکل زیر:

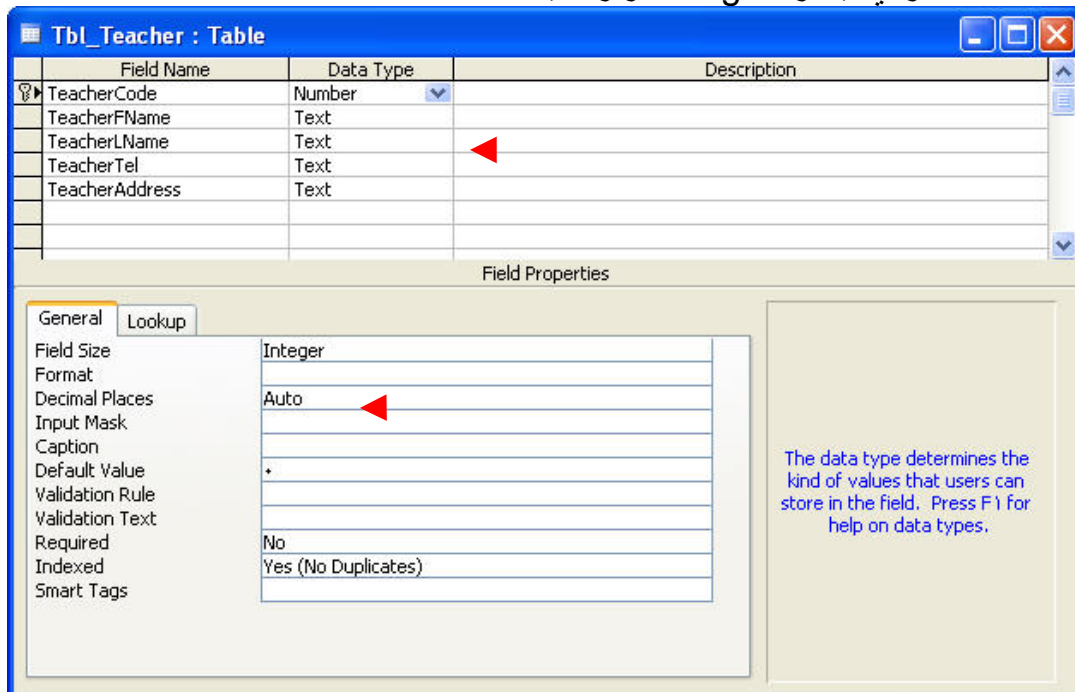


- روی دکمه Next کلیک کنید. در پنجره‌ی باز شده مطابق شکل زیر گزینه‌ی Modify Table Design را انتخاب و دکمه Finish را کلیک کنید.





- با انتخاب گزینه‌ی بالا پنجره‌ی طراحی جدول نمایش داده می‌شود. مطابق شکل زیر و بر اساس مشخصه‌های فیلد تمرین ، نوع داده را به Number و اندازه‌ی فیلد را Integer قرار دهید.



### تمرین 3-2 :

جدول Tbl\_Student را به این روش ایجاد کنید.

### 3-3- ایجاد جدول به وسیله نماي Datasheet

در این حالت اطلاعات در سلول ها وارد می‌شوند بدون آنکه فیلدها از قبل تعریف شده باشند. با ذخیره اطلاعات وارد شده ، Access بر حسب مقادیر داده‌ی وارد شده در هر ستون ، نوع فیلد را حدس زده و نوع داده‌ی فیلد را تعیین می‌کند . بدین ترتیب جدول ایجاد می‌شود .

### مثال 3-3 :

#### مراحل ایجاد جدول Tbl\_Register به وسیله نماي Datasheet :

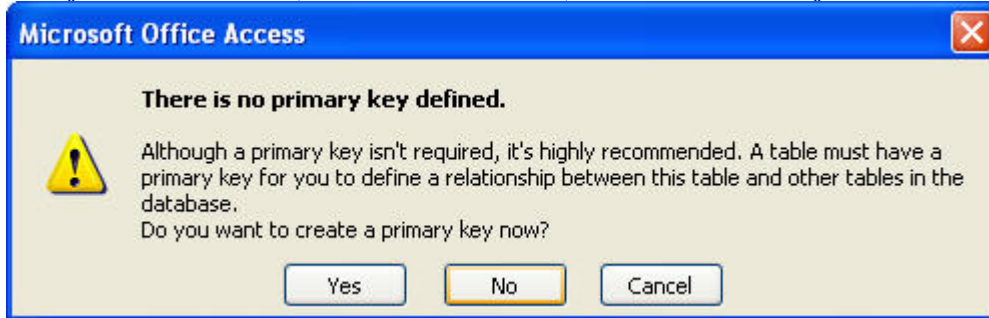
- از لیست ایجاد جدول در پنجره‌ی اشیاء جدول ، Create table by entering data را انتخاب کنید.
- مطابق شکل ، در پنجره‌ی باز شده ، اطلاعات را وارد نمایید.

	Field1	Field2	Field3	Field4
10	۱۰	۳۴۵۶	۱۳۸۵/۰۶/۲۰	
11	۱۱	۴۰۵۴۶	۱۳۸۵/۰۶/۲۰	
13	۱۳	۴۵۰۵۵	۱۳۸۵/۰۶/۲۲	
15	۱۵	۴۴۳۵۰	۱۳۸۵/۰۶/۲۳	
17	۱۷	۲۴۵۹۰	۱۳۸۵/۰۶/۲۳	
22	۲۲	۲۰۰۵۴	۱۳۸۵/۰۶/۲۴	

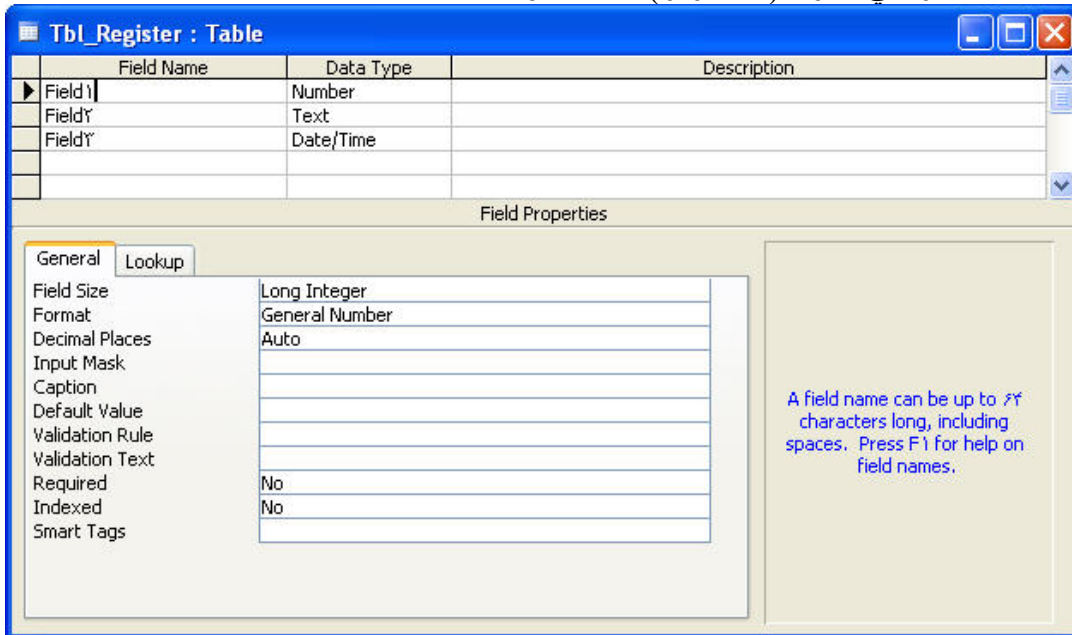
Record: 22 of 21

### ادامه‌ی مثال 3-3 :

- پس از آن ، دکمه‌ی Save از نوار ابزار یا گزینه‌ی Save از منوی File را انتخاب کنید تا پنجره‌ی ذخیره‌سازی جدول ظاهر شود.
- نام آن را Tbl\_Register قرار دهید و Ok را کلیک کنید. چون هنوز فیلد کلید اصلی را مشخص نکرده‌ایم ، مانند مثال 2-3 پیغام زیر نمایش داده می‌شود.

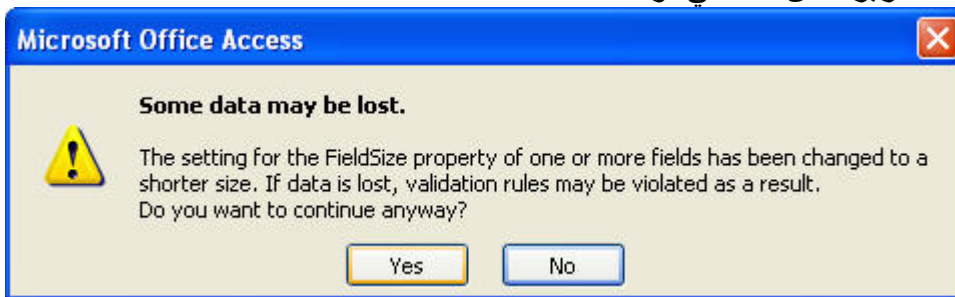


- دکمه‌ی No را انتخاب کنید.
- پنجره‌ی ورود داده را ببندید. در Tbl\_Register در اشیاء جدول بانک ثبت شده است.
- از نوار ابزار پنجره‌ی بانک ، دکمه‌ی  Design را انتخاب کنید. تا پنجره‌ی طراحی جدول (شکل زیر) نمایان شود.



### ادامه‌ی مثال 3-3 :

- نام فیلد و نوع فیلد را مطابق مشخصات داده شده تصحیح نمایید.
- کلید اصلی جدول را مشابه مثال 2-3 انتخاب کنید و تغییرات جدول را ذخیره نمایید.
- به دلیل آن‌که اندازه‌ی فیلد StudentCode را از 50 به 5 تغییر داده‌ایم ، پیغام زیر نشان داده می‌شود.



- بدون هیچ ترسی دکمه‌ی Yes را انتخاب کنید! اندازه‌ی داده‌های وارد شده در این فیلد از 5 کاراکتر بیشتر نیست و داده‌ای حذف نخواهد شد. در سایر مواقع کاراکترهای بیش از 5 از بین می‌روند.
- جدول اصلاح شده ذخیره می‌شود.

### 3-4- ایجاد جدول به وسیله‌ی Import Table و Link Table

با انتخاب گزینه‌های Import یا Link در پنجره‌ی New Table ، می‌توان از جدول‌های پایگاه داده‌ی موجود ، جدول‌هایی مشابه ایجاد کرد.

در حالت دریافت Import ، هیچ‌گونه ارتباطی بین پایگاه داده فعلی و بانک قبلی وجود نخواهد داشت و طراحی جدول به همراه داده‌های موجود در آن به بانک جدید منتقل می‌شود.

ولی در حالت Link ، جدول مبدا در بانک مربوطه اش قرار دارد و جدول افزوده شده به آن متصل شده است و در صورتی‌که محل فیزیکی بانک مبدا تغییر کند ، لینک جدول افزوده شده بی‌اثر خواهد شد . داده‌هایی که در بانک اضافه می‌شوند، در جدول لینک شده نمایش می‌یابند و بالعکس یعنی داده‌های وارد شده به جدول لینک شده ، در جدول مبدا اضافه می‌شود. در این حالت ، طراحی جدول

لينك شده را در بانك جديد نمي توان تغيير داد ، بلکه بايد در بانك مبدا تغييرات اعمال شود.

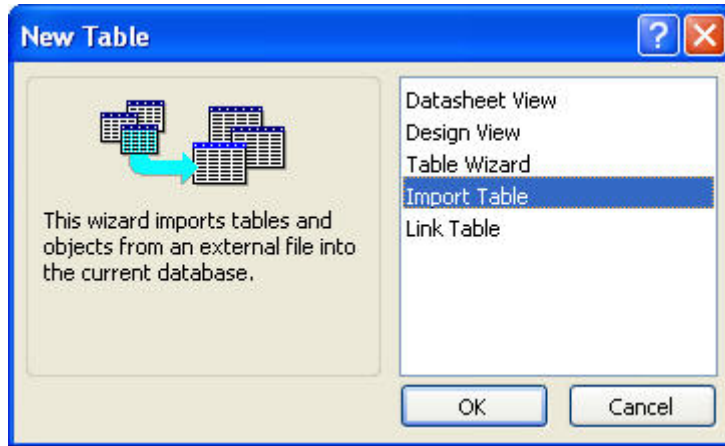
با انتخاب هر يك از اين گزينه ها ، ابتدا نام و مسير بانك موجود و موردنظر را انتخاب مي كنيم ، سپس از پنجره ي باز شده جدول يا جدول ها موردنظر را انتخاب و Add مي كنيم .

#### **كنجاوي :**

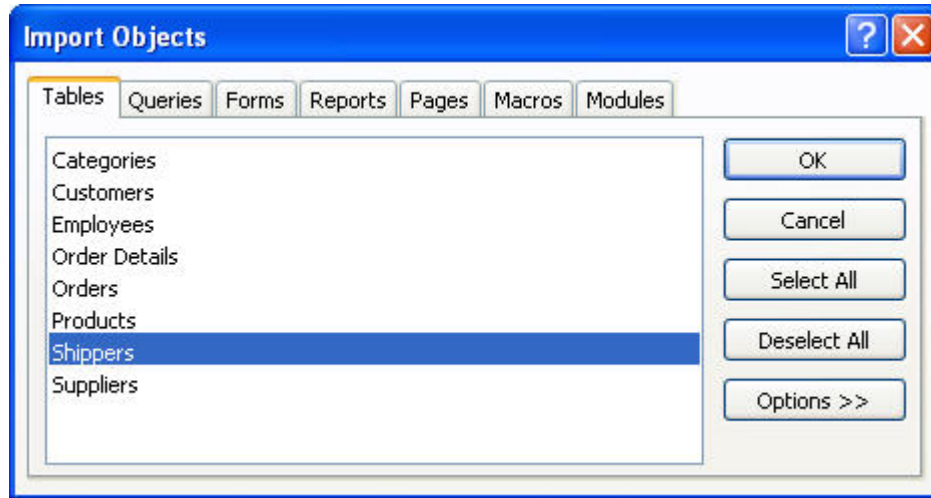
1. آيا مي توان با روش Import و Link ساير اشياء بانك داده را نيز منتقل كرد؟ اين كار چگونه صورت مي گيرد؟
2. کدام اشياء قابل انتقال نيستند؟
3. آيا شي خاصي وجود دارد كه فقط به يكي از اين دو روش منتقل شود؟

مثال 3-4 :

- مراحل ایجاد جدول رشته‌های تحصیلی (Tbl\_Major) به وسیله Import :
- از منوی insert ، Table را انتخاب و از لیست New Table ، Import Table را انتخاب کنید. مطابق شکل زیر :



- در پنجره‌ی باز شده مسیر بانک Northwind.mdb را انتخاب کنید و کلیک نمایید.
- از پنجره‌ی باز شده مطابق شکل زیر ، Shippers را انتخاب و کلید OK را انتخاب کنید.



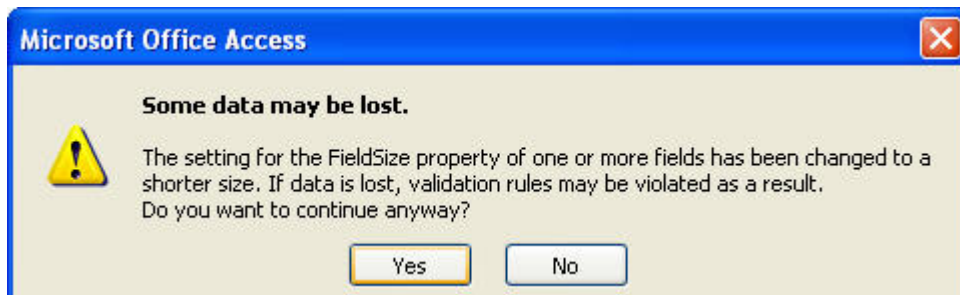
- جدول Shippers به لیست اشیاء جدولی بانک اضافه می‌شود. آن را انتخاب و گزینه‌ی Design را از نوار ابزار پنجره‌ی بانک انتخاب کنید و تغییرات را مشابه مشخصات بانک اعمال کنید. یعنی تغییر نام ShipperID به MajorCode و تغییر مشخصات فیلد آن به Number و نوع Integer. تغییر نام CompanyName به MajorName و تغییر اندازه‌ی آن به عدد 20. همچنین ردیف فیلد Phone را انتخاب و کلید Delete را کلیک کنید تا حذف شود.
- مشخصه‌ی caption فیلدها را نیز پاک کنید و یا تصحیح نمایید.
- با کلیک دکمه‌ی Delete در ردیف Phone، پیغام زیر ظاهر می‌شود. Yes را انتخاب کنید.



- داده‌های ستون Description را پاک کنید یا متناسب با تعریف فیلد جدید اصلاح نمایید.

### ادامه‌ی مثال 3-4 :

- به دلیل کاهش اندازه‌ی فیلد ، هنگام ذخیره‌ی جدول ، پیغام زیر نمایش داده می‌شود، دکمه‌ی Yes را انتخاب کنید.



- جدول Shippers به لیست اشیاء جدول اضافه شده است. آن را به Tbl\_Major تغییر نام دهید. این کار با روش‌های مشابه تغییر نام فایل در محیط ویندوز قابل انجام است.

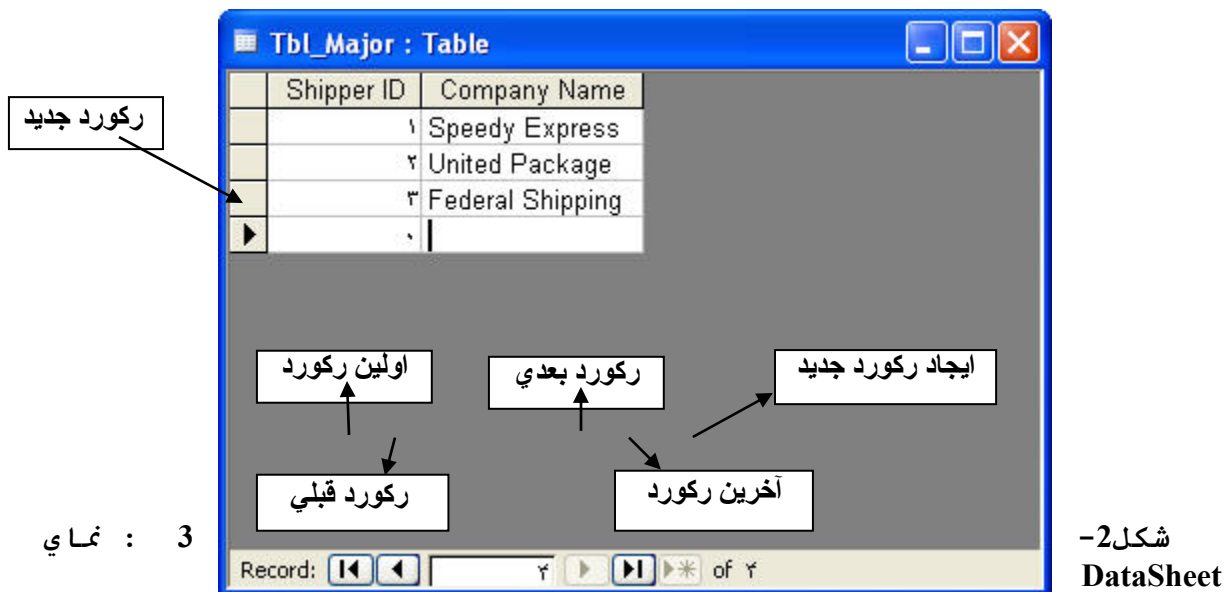
### کنجکاو ی :

آیا روش ایجاد Link مشابه روش Import است؟ دلایل استفاده از روش Link را توضیح دهید و در این محیط عملیاتی برای کدام جدول/جدول‌ها ، این اتصال معنی‌دار است؟



### 3-5- ورود داده ها به جدولها

برای وارد کردن مقادیر داده‌ها در جدول، ابتدا جدول را از پنجره‌ی اشیاء بانک، انتخاب کنید. سپس روی نام جدول موردنظر دوبار کلیک کنید و یا از دکمه‌ی های فرمان، Open را کلیک نمایید. نمایی مشابه Datasheet و به صورت زیر (شکل 2-3) نمایش داده می‌شود. جدول انتخاب شده Tbl\_Major است.



3 : نمایی

شکل 2-  
DataSheet

#### • افزودن رکوردها :

با تایپ مقادیر فیلدها در ردیف آخر رکوردها (علامت \*) یک رکورد جدید ایجاد می‌شود. همچنین می‌توانید دکمه New Record را در پایین Datasheet کلیک کنید و مقادیر را وارد نمایید. Access برای ذخیره‌ی مقادیر داده پیغامی نمایش نمی‌دهد و همزمان با تایپ کردن داده‌ها، ذخیره‌سازی انجام می‌شود.

#### تمرین 3-3 :

مقدار Test a Major را با کد درس 10 ایجاد کنید.

#### • ویرایش رکوردها :

برای ویرایش رکورد به سادگی نشانگر را روی رکورد ببرید و تغییرات موردنظر را اعمال کنید. با استفاده از کلیدهای جهت می‌توانید از یک فیلد به دیگری جابه‌جا شوید. همچنین می‌توانید از دکمه‌های قبلی (Prev) ، بعدی (Next) ، اولین (First) و آخرین (Last) در پایین Datasheet نیز برای این‌کار استفاده کنید. مشابه افزودن داده ، برای ویرایش نیز پیغامی نمایش نمی‌دهد و هم‌زمان با تایپ ، تمام تغییرات لحاظ می‌شود. گرچه می‌توان با کمک Undo Typing از منوی Edit ، تغییرات را به حالت قبل برگرداند ولی با بستن پنجره‌ی جدول ، تغییرات قطعی می‌شود.

#### تمرین 3-4 :

در جدول Tbl\_Major ، مقدار Test a Major را به مقدار "ریاضی فیزیک" تغییر نام دهید.

#### • حذف رکوردها :

با انتخاب رکورد موردنظر و انتخاب Delete Record از منوی Edit و یا کلیک گزینه‌ی حذف از نوار ابزار محیط کار ، می‌توان آن رکورد را حذف کرد. ابتدا پیغامی برای تایید حذف ، نمایش می‌یابد و در صورت تایید ، رکورد حذف می‌شود.

#### تمرین 3-5 :

در جدول Tbl\_Major ، رکوردهای Speedy Express ، United Package و Federal Shipping را حذف کنید.

#### • تغییراندازه سطرها و ستون‌ها :

ارتفاع سطرها در نمای Datasheet با کشیدن خط نوارخاکستری بین ردیف‌ها که با جابه‌جایی ماوس به شکل صلیب (+) درمی‌آید قابل انجام است. با تغییر ارتفاع یک ردیف تمام ردیف‌ها در نمای Datasheet تغییر می‌کند.

عرض ستون‌ها به شکلی مشابه ردیف قابل تغییر است. با دوبار کلیک روی خط نیز به‌طور خودکار اندازه ، مناسب با بزرگترین مقدار واردشده در ستون ،

تنظیم می‌شود. برخلاف ردیف‌ها ، هر ستون می‌تواند اندازه‌ی متفاوتی داشته باشد. مقادیر دقیق‌تر به وسیله‌ی انتخاب Row Height و Column Width از منوی Format و وارد کردن مقدار طول و عرض قابل تنظیم است.

#### • ثابت کردن ستون ها :

مشابه حالت Freezing در Excel می‌توان ، ستون‌های جدول را در Access ثابت کرد. در این صورت ستون ، در ابتدایی‌ترین قسمت نما قرار می‌گیرد. برای ثابت‌سازی ، ستون موردنظر را انتخاب و Freeze Columns را از منوی Format انتخاب می‌کنیم. برای برگشت ثابت‌سازی ، ستونی را انتخاب و UnFreeze All Columns را از منوی Format انتخاب می‌کنیم.

#### • پنهان‌سازی ستون ها :

ستون‌ها می‌توانند - گرچه در بانک وجود دارند - از دید کاربر پنهان باشند. برای پنهان‌سازی ، ستون موردنظر و یا ستون‌های هم‌جوار موردنظر را انتخاب و از منوی Format ، Hide Columns را انتخاب می‌کنیم. برای نمایش مجدد ستون ، از منوی Format ، Unhide Columns را انتخاب می‌کنیم. پنجره‌ای باز می‌شود که تمام فیلدها و درکنار هر یک ، یک Check box دیده می‌شود. با انتخاب Check box کنار فیلدهای موردنظر و بستن پنجره ، ستون‌ها بر حسب انتخاب یا عدم انتخاب Check box ها ، ظاهر و یا پنهان می‌شوند.

#### • جستجو و یافتن داده در جدول :

می‌توان داده/اطلاع خاصی را در جدول جستجو کرد. برای این کار به صورت زیر عمل کنید :

1. باز کردن جدول در نمای Datasheet.
2. قراردادن نشانگر در ستونی که می‌خواهید جستجو شود و انتخاب Find... از منوی Edit.

3. پنجره **Find and Replace** محاوره‌ای **Find** نمایش داده خواهد شد. مقدار شرط را در جعبه متن **Find** **What:** وارد کنید.
4. در لیست بازشونده **Look In:** ناحیه‌ی مورد جستجو را تعیین کنید. کل جدول یا فیلد مشخص شده.
5. معیار تطابق را در لیست بازشونده **Match** اختیار کنید. سه انتخاب موجود است:
- a. Any Part of Field** : زیرمجموعه کاراکترهای هر فیلد را مطابقت می‌دهد.
- b. Whole Field** : کل فیلد را مطابقت می‌دهد.
- c. Start of Field** : شروع فیلد را مطابقت می‌دهد.
6. از لیست بازشونده **Search** ، می‌توانید **All** را برای تمام ستون‌ها ، **Up** را برای جستجو به سمت ابتدای قبل از محل رکوردی که در آن هستیم و **Down** را برای جستجو به سمت انتهای بعد از محل قرارگرفتن رکورد انتخاب کنید.
7. پس از تنظیم کل معیارها ، روی دکمه **Find** **Next** کلیک نمایید. اگر بیش از یک رکورد در آن شرط صدق کند ، کلیک روی دکمه **Find Next** را تا رسیدن سایر رکوردهای دارای شرط ادامه دهید.

### تمرین 3-6 :

آیا رکوردی با مقدار "ریاضی فیزیک" در جدول **Tbl\_Major** وجود دارد؟

### • جایگزین کردن داده :

با تابع **Replace** ، می‌توان مقدار خاص داده را با مقدار جدید جایگزین کرد.

1. از منوی **Edit** ، **Replace** را انتخاب کنید. یا

زبان‌های **Replace** را در پنجره محاوره‌ای **Find and Replace** انتخاب کنید.




2. برای جستجو و انتخاب داده‌های موردنظر، به

روش جستجوی فوق عمل کنید و معیارها را تنظیم نمایید. مقدار جدید را در جعبه متن **Replace With:** تایپ کنید.

3. با کلیک Find Next و سپس کلیک دکمه Replace ، مقدار یافته شده با مقدار جدید جایگزین می شود.
4. با کلیک Replace All ، تمام مقادیر دارای شرط به یکباره جایگزین می شوند.

### • مرتب سازی و فیلتر کردن داده ها :

با امکان مرتب سازی می توان ترتیب نمایش رکوردها را تعریف کرد و با فیلتر کردن می توان فقط رکوردهای مطلوب و منطبق با معیار را مشاهده کرد<sup>1</sup>.

- **مرتب سازی :** فیلد مورد نظر برای مرتب سازی را انتخاب کنید ، از منوی Sort -> Records ، گزینه Sort Ascending یا Sort Descending را انتخاب کنید ، یا دکمه  یا  را در نوار ابزار کلیک کنید. برای مرتب سازی با بیش از یک فیلد ، ابتدا ستون های مورد نظر را انتخاب کنید ، سپس به صورت قبل عمل کنید.
- **فیلتر کردن :** رکوردهایی که دارای شرایط مورد نظر باشند را نمایش می دهد. فیلد مورد نظر را انتخاب کنید ، روی رکورد حاوی مقدار مطلوب قرار بگیرید ، از منوی Filter ، گزینه Filter By selection را انتخاب کنید یا روی نوار ابزار دکمه  را انتخاب کنید.
- **حذف فیلتر :** دکمه Remove Filter را در نوار ابزار کلیک کنید. تمام رکوردهای جدول دوباره نمایش داده می شود.

<sup>1</sup>- با مفهوم فیلتر در Excel آشنا شده اید.

### مثال 3-5 :

مقادیر زیر را در جدول Tbl\_Major ، در بانک Proj.mdb که ایجاد کرده‌اید اضافه کنید.



MajorCode	MajorName
۱	ریاضی و فیزیک
۲	علوم تجربی
۳	علوم انسانی
۴	فنی حرفه‌ای
۵	کار دانش
۶	

### • حذف و اضافه ستون ها :

بهترین راه افزودن و یا حذف فیلد از جدول ، استفاده از نمای Table است چراکه انتخاب‌های متعددی در اختیار هست . در نمای Datasheet هم می‌توان این کار را انجام داد که مشابه حذف و اضافه ستون در نرم افزار Excel است.

### یادآوری :

ستونی را که می‌خواهید ، ستون جدید در سمت چپ آن افزوده شود انتخاب کنید سپس با انتخاب Insert Column از منوی Edit ستون جدید افزوده می‌شود. برای حذف می‌توان ستون موردنظر را انتخاب و با گزینه Delete Column از منوی Edit آن را حذف کرد. زمانی که ستونی حذف یا اضافه شود و جدول ذخیره گردد تغییرات قطعی می‌شوند و قابل برگشت به وسیله Undo نخواهند بود.

### تمرین 3-7 :

در جدول Tbl\_Course ، فیلدهای ساعت تشکیل کلاس ProgramHour و شماره کلاس ClassName را اضافه کنید. آن را ذخیره نمایید. در نمای Design این فیلدهای جدید چه خصوصیتی دارند؟

### 3-6- ایجاد ارتباط بین جدولها

انواع ارتباطها بین جدولها را دیده ایم. در نظریه‌ی بانک‌های اطلاعاتی رابطه‌ای، می‌توانیم یکپارچگی ارتباطها را تعریف کنیم که به **Referential Integrity** معروف است. در **Access** می‌توانیم ارتباطهای بین جدولها را معرفی کنیم بدون آن که کنترل یکپارچگی به وسیله‌ی بانک اعمال گردد، ولی حسن تعریف ارتباط آن است که **Access** کنترل‌هایی را روی عملیات داده‌ها اعمال می‌کند و با تعریف مجموعه‌ی قواعد هم از صحت ارتباط بین رکوردهای دو جدول مرتبط شده اطمینان می‌یابیم، هم **Access** سازگاری و یکپارچگی اطلاعات بانک را حفظ می‌کند. به‌طور مثال، هنگام حذف رکورد از جدولی که ارتباط یک-به-چند با جدولی دیگر دارد، باید رکوردهای جدول جزئیات نیز حذف گردد چراکه درغیراین‌صورت چون اطلاعات اصلی (**master**) حذف شده است، رکوردهای جدول دوم (**detail**) بلامتکلیف می‌مانند. برای جلوگیری از این‌گونه ناسازگاری‌های داده، زمان معرفی ارتباط در **Access**، یکپارچگی را نیز تعریف می‌کنیم.

**برای ایجاد یکپارچگی شرایط زیر باید وجود داشته باشد :**

- فیلد برقرارکننده‌ی ارتباط در جدول پدر (**master**) کلید اصلی باشد و یا دارای اندیس منحصر به فرد (**Unique**) باشد.
- فیلدهای برقرارکننده در هر دو جدول، نوع داده‌ی (**Data Type**) یکسانی داشته باشند<sup>1</sup>.
- هر دو جدول در یک بانک اطلاعاتی قرار داشته باشند.

هنگامی که یکپارچگی ارتباط فعال گردد باید موارد زیر را مدنظر قرار دهیم :

<sup>1</sup>- برای مطالعه‌ی بیشتر به مفهوم میدان در کتاب‌های بانک اطلاعاتی مراجعه شود.

- نمی‌توانیم در فیلد کلیدخارجی (فیلد مشترك در جدول فرزند (detail)) مقداری را وارد نماییم که در جدول پدر موجود نباشد.
- نمی‌توانیم از جدول پدر رکوردی را حذف کنیم که رکوردهای متناظر در جدول فرزند وجود داشته باشند. مگر آن‌که در تعریف یکپارچگی رابطه به Access مجوز برای حذف رکوردهای متناظر در جدول فرزند داده باشیم. در این صورت خودبه‌خود رکوردهای متناظر در جدول فرزند حذف می‌شوند.
- نمی‌توانیم مقدار فیلد کلیداصلي را در جدول پدر تغییر دهیم و ویرایش نماییم ، اگر رکوردهای متناظری در جدول فرزند وجود داشته باشند.

#### کنجکاوِي :

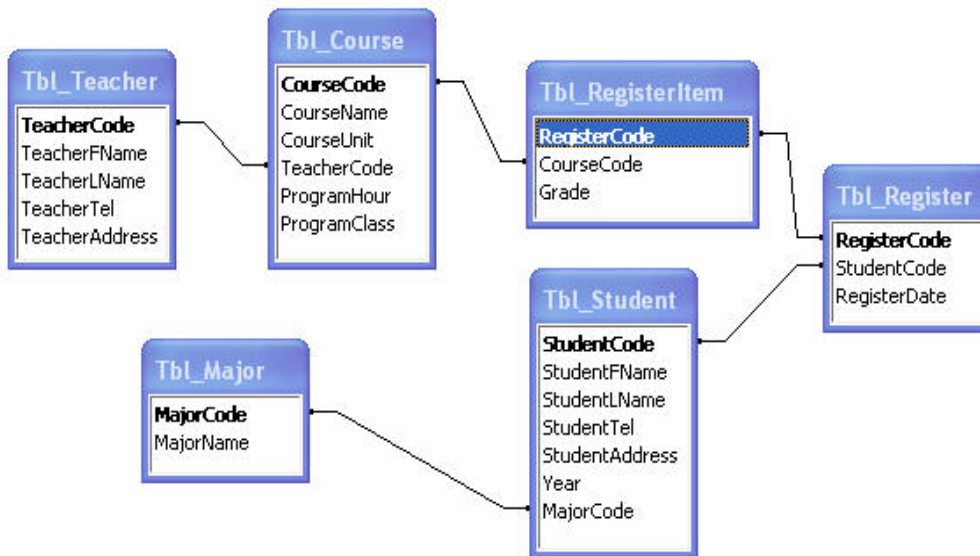
- 1) با تعریف چه نوع ارتباطی ، با حذف رکوردی در جدول پدر ، رکوردهای متناظر در جدول فرزند خودبه‌خود حذف می‌گردد؟
- 2) چگونه با تغییر مقدار فیلداصلي در جدول پدر ، فیلد متناظر در جدول فرزند نیز ویرایش و به‌هنگام می‌شود؟

برای آن که کنترل یکپارچگی و صحت روابط بین جدول‌ها انجام گیرد ، زمان معرفی ارتباط باید گزینه‌ی Enforce Referential Integrity را انتخاب کرده ، در پنجره‌ی ظاهر شده ، شرایط موردنظر را تنظیم کنیم. در تمرین زیر چگونگی ایجاد رابطه و اعمال یکپارچگی را مشاهده خواهید کرد.





### مثال 3-6 :

ارتباطهاي زير را در بانك خود اضافه كنيد.

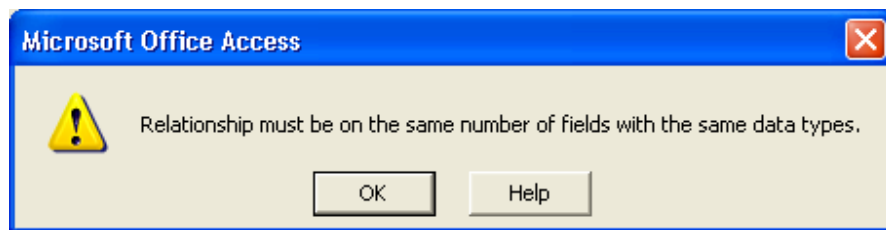


مراحل انجام كار :

- از منوي نوار ابزار، گزینه‌ي Relationships  را انتخاب نماييد. در پنجره‌ي باز شده، جدول‌هاي موردنظر را با كليك روي  اضافه نماييد.
- مانند شكل از فيلد مشخص شده‌ي يك جدول به فيلد متناظر در جدول ديگر درگ كنيد تا ارتباط ايجاد شود.
- در پنجره‌ي باز شده بدون هيچ تغييری، Create را انتخاب كنيد. دقت كنيد كه يکپارچگي روابط در همين پنجره تعيين مي‌شوند، گزینه‌ي Enforce Referential Integrity که ما آن را بدون تغيير گذاشته‌ايم.

همان‌طور که در تبديل موجوديت و ارتباطات به جدول‌ها در فصل اول ديديد، فيلد كليداصلي از يك جدول در ديگري اضافه شده است. اين ارتباطات در زمان طراحي بانك مشخص مي‌شوند. در نوع ارتباط يك-به-چند، فيلدكليداصلي جدول طرف "يك" در جدول طرف "چند" اضافه مي‌شود. و در جدول دوم اصطلاحا كليد خارجي نام دارد. نام فيلد كليد خارجي مي‌تواند متفاوت از نامي باشد که در جدول اول استفاده شده است ولي نوع آن‌ها

الزاما يکي است ، درغیراین صورت در ایجاد یکپارچگی رابطه Referential Integrity خطاي زیر(شکل 3-3) رخ می دهد.



شکل 3-3 : پیغام خطاي نوع فیلد

### کنجکاوې :

زمان ایجاد رابطه در محیط Relationships ، درگ کردن از فیلد کلیداصلي جدول پدر به فیلد کلیدخارجي جدول فرزند یا عکس این عمل یعنی درگ کردن از جدول فرزند به جدول پدر ، آیا تفاوتی ایجاد می کند؟

## خلاصه ي فصل

شيء جدول ، داده هاي بانك را نگه مي دارد. جدول هاي ترسيم شده روي كاغذ بايد در پايگاه داده تعريف و ايجاد شوند. سه روش اصلي براي ايجاد جدول وجود دارد : **Design View** ، **Table Wizard** ، **Datasheet View** و همچنين يك روش انتقال به وسيله **Import Table** و يا **Link Table**.

پس از ايجاد جدول و تعريف خصوصيات فيلدهاي متناسب با طراحي انجام شده ، مي توان داده ها را در آن ثبت كرد. يكي از راه هاي ورود داده ها به جدول به كمك نماي **Datasheet** است كه با امكانات موجود در اين نما مي توان ركوردها را اضافه كرد ، داده هاي آن را ويرايش كرد ، حذف كرد و يا طرح نمايش را با تغيير اندازه ي سطروستونها و يا تغيير رنگ ، نوع قلم ، ثابتسازي ستون ، پنهانسازي ستون و ... تغيير داد. مي توان در اين نما داده هاي موردنظر را جستجو كرد ، مقدار داده را با يك مقدار جديد جايگزين نمود و يا ترتيب نمايش ركوردها را با مرتبسازي تغيير داد و يا ركوردها را برحسب شرايط خاص فیلتر كرد. دیدیم عموماً جدول هاي پايگاه داده با يكدیگر ارتباط دارند. ايجاد رابطه (**Relation**) كه در طراحي مشخص شده است با انتقال كليداصلي به جدول ديگر به عنوان كليدخارجي صورت مي گيرد. در **Access** ارتباط موجود مي تواند به وسيله ي قواعدسازگاري كنترل شود تا ناسازگاري داده اي در بانك به وجود نيايد. اينكار با تعريف **Referential Integrity** در محيط **Relationships** ايجاد مي شود.

## خود آزمائي

1. روش هاي ايجاد جدول را با هم مقايسه كنيد.
2. مشخصه هاي انواع داده ي فيلدها را بيان كنيد.
3. شرايط ايجاد يکپارچگي ارتباط در جدول چيست؟
4. قواعد سازگاري چه مواردی را كنترل مي كند؟

5. تفاوت انتقال جدول از بانک دیگر به کمک  
Import و Link چیست؟

## 4- فصل چهارم : ایجاد و کار با فرم ها

فرم ، واسطی است بین کاربر و پایگاه داده . در این فصل ابتدا با مفاهیم اولیه فرم و اجزای آن که به آن ها کنترل (Control) می‌گوییم آشنا شده ، سپس ایجاد فرم و عملیات روی آن را بررسی می‌کنیم . فرم ها اطلاعات بانک را در قالبی جذاب نشان می‌دهند. با ایجاد فرم ، کاربر، داده ها را آسان تر وارد می‌کند و امکان اشتباه نیز کاهش می‌یابد.

### در پایان این فصل انتظار می‌رود که هنرجو بتواند :

- کنترل‌های روی فرم را ایجاد کند.
- بتواند طراحی کنترل‌های روی فرم را تغییر دهد.
- به وسیله‌ی گزینه‌ی **Form Wizard** فرمی را ایجاد و ازگزینه‌های این ابزار استفاده کند.
- در نمای **Design View** کنترل‌ها را به انواع دیگر تبدیل کند و مشخصه‌های آن‌ها را تغییر دهد.
- بتواند در نمای **Form View** فرم را ایجاد و رکوردهایی را ثبت ، ویرایش یا حذف نماید.

## 4-1- مفاهیم اولیه فرم و اشیاء آن

### 4-1-1- انواع "نوع کنترل" در ایجاد فرم ها :

- مقید<sup>1</sup> : داده‌ها از منبع رکوردهای متصل شده به فرم نمایش می‌یابند ، می‌توان رکوردها را ویرایش کرد ، حذف کرد و یا اضافه نمود.
- نامقید<sup>2</sup> : به رکوردها متصل نیستند و محتوای آن‌ها با جابه‌جایی از رکوردی به دیگری تغییر نمی‌کند.

### 4-1-2- انواع کنترل‌ها : نمونه‌هایی از این کنترل‌ها را در

شکل‌های (4-1) و (4-2) می‌بینید.

کنترل	نوع	مفهوم
<b>Label</b>	<b>Unbound</b>	توضیحی در خصوص متن هم‌جوارش ارائه می‌کند.
<b>Text Box</b>	<b>Unbound</b>	داده‌ها را از منبع رکورد (Record Source) نمایش می‌دهد ، می‌توان آن را ویرایش نمود و یا داده‌ی جدیدی را وارد کرد.

<sup>1</sup> - Bound

<sup>2</sup> - Unbound

لیست ورودی‌های ممکن را نمایش می‌دهد.	<b>Bound</b>	<b>List Box</b>
لیست ورودی‌ها را نشان می‌دهد ، همچنین محلی متنی برای ورود متن داده به وسیله صفحه کلید نیز فراهم می‌کند. ترکیبی است از List box و Text box.	<b>Bound</b>	<b>Combo Box</b>
جنبه‌ی سه بُعدی به فرم می‌افزاید.	<b>Unbound</b>	<b>Tab Control</b>
انتخاب بله/خیر را برای یک فیلد فراهم می‌کند. اگر علامت <input checked="" type="checkbox"/> داشته باشد ، بله است.	<b>Bound</b>	<b>Check Box</b>
انتخاب بله/خیر را برای یک فیلد فراهم می‌کند. اگر فشرده شده باشد ، بله است.	<b>Bound</b>	<b>Toggle Button</b>
انتخاب را برای یک فیلد فراهم می‌کند.	<b>Bound</b>	<b>Option Button</b>
انتخاب‌های یک فیلد را دسته‌بندی می‌کند و نمایش می‌دهد.	<b>Bound</b>	<b>Option Group</b>
داده‌ی OLE <sup>1</sup> مثل تصاویر را نمایش می‌دهد.	<b>Bound</b>	<b>Bound Object Frame</b>
نمایش تصویر و یا یک فیلم کوتاه که به رکوردی متصل نباشد.	<b>Unbound</b>	<b>Unbound Object Frame</b>
با آن می‌توان خطوط و مربع یا مستطیل‌ها را در فرم رسم کرد.	<b>Unbound</b>	<b>Line and Rectangle</b>
ساده‌ترین راه اجرای یک فرمان یا یک ماکرو است. با کلیک روی آن برنامه‌ی نوشته شده اجرا می‌شود.	<b>Unbound</b>	<b>Command Button</b>

جدول شماره 4-1 : انواع کنترل‌ها

شکل 4-1 : کنترل‌های یک فرم نمونه





1 - Object linking and embedding



شکل 2-4 : نمونه‌ای از کنترل‌های یک فرم

### 3-1-4- روش تغییر اندازه و جابه‌جایی کنترل‌ها :

با جابه‌جایی ماوس روی کنترل ، اشکال زیر ظاهر می‌شوند که هر یک عمل متفاوتی انجام می‌دهد :

عمل مورد انتظار	چه زمانی ظاهر می‌شود؟	شکل
با کلیک می‌توان کنترل مورد نظر را انتخاب کرد.	زمانی که به کنترل انتخاب‌نشده اشاره کنید، نمایش داده می‌شود. (همان وضعیت پیش‌فرض ماوس است)	
با درگ ، می‌توان کنترل انتخاب‌شده را جابه‌جا کرد.	زمانی که به لبه‌ی کنترل انتخاب‌شده اشاره کنید ، ظاهر می‌شود. (البته نه در زمان اشاره به مربعات تغییر اندازه)	
با درگ ، می‌توان آن کنترل را جابه‌جا کرد نه به همراه سایر کنترل‌های انتخاب‌شده.	زمانی که به مربع بزرگ تغییر اندازه‌ی سمت چپ-بالای کنترل انتخاب‌شده اشاره کنید ، نمایش داده می‌شود.	
با درگ می‌توان اندازه‌ی کنترل را تغییر داد.	زمانی که به مربع تغییر اندازه اشاره کنید. (البته نه مربع چپ-بالا)	

جدول شماره 2-4 : روش تغییر اندازه و جابه‌جایی کنترل

#### 4-1-4- انتخاب بیش از یک کنترل :

- کلیک روی اولین کنترل (انتخاب یک کنترل) و انتخاب سایرین با **Shift+click** روی هر یک از آنها.
- درگ **selection box** (کلیک یک نقطه و درگ روی کنترل، به طوری که داخل مربع نقطه چین نمایش یافته قرار گیرد).

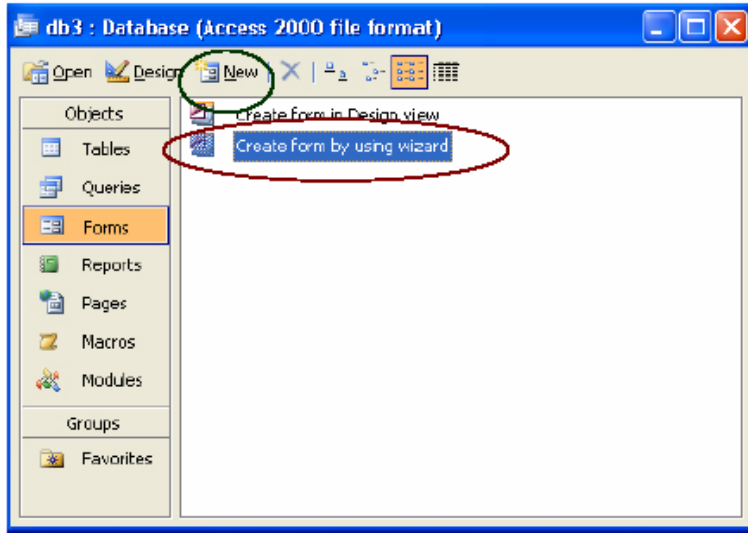
#### 4-2- ایجاد فرم

راه‌های مختلفی برای ایجاد فرم وجود دارد. دو راه ایجاد فرم که در اینجا بررسی می‌کنیم عبارت است از : **Auto Form Wizard** و **Design View**.

#### 4-2-1- ایجاد فرم به وسیله **Auto form Wizard**

##### ایجاد فرم **Frm\_Course** :

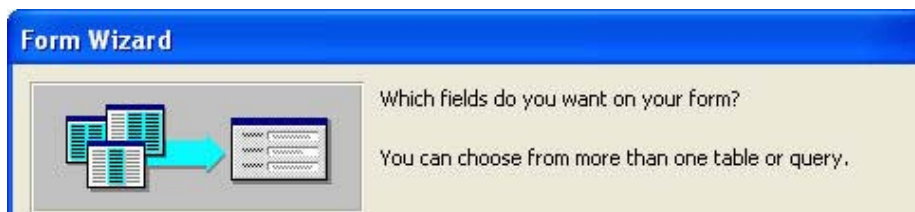
- کلیک روی **Forms** در نوار اشیاء .
- کلیک روی دکمه **New** روی نوار دکمه‌های فرمان و انتخاب **Form Wizard** از پنجره‌ی ایجاد فرم ظاهر شده . یا می‌توانید **Create form by using wizard** از پنجره‌ی اشیاء را کلیک کنید (شکل 3-4) .



4 : ایجاد  
به وسیله‌ی

شکل 3-  
فرم  
Wizard

- جدول یا پرسوجوی موردنظر را از لیست بازشونده **Tables/Queries** انتخاب کنید. جدول یا پرسوجو باید تمام اطلاعات موردنیاز نمایش روی فرم را دربرداشته باشد. در اینجا انتخاب **Tbl\_Course** (شکل 4-4) .





لیست بازشونده‌ی  
Tables/Queries

فیلدهای انتخاب‌شده  
برای نمایش

شکل 4-4 : انتخاب فیلدهای موردنیاز برای نمایش روی فرم  
دکمه‌های انتقال

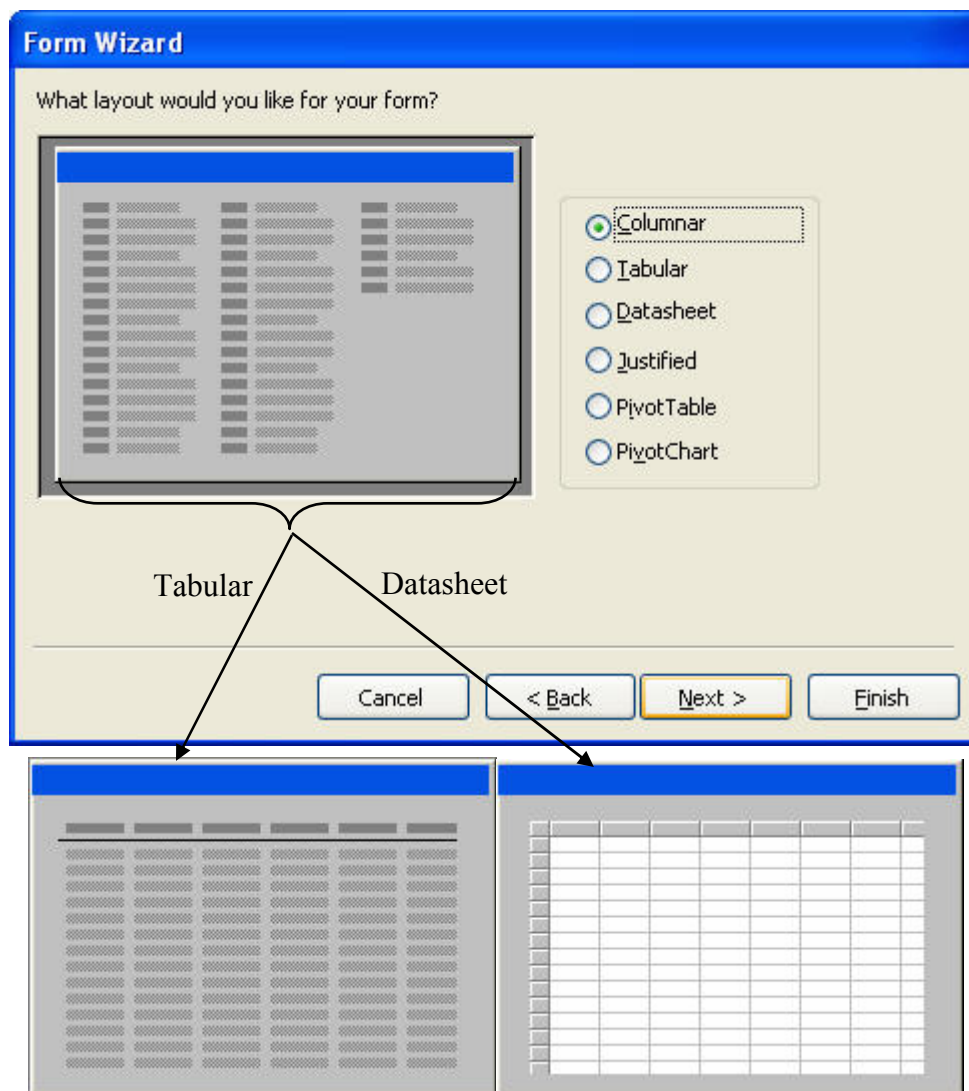
- بر روی دکمه‌ی >> کلیک کنید تا تمام فیلدهای جدول به لیست سمت راست با عنوان Selected Fields افزوده شود. یا با کلیک روی دکمه > فیلدهای موردنظر خود را یکی یکی اضافه کنید. در صورتی که فیلدی به اشتباه در لیست راست اضافه شده باشد با دکمه < آن را از لیست انتخابی خود حذف کنید.

- با کلیک دکمه‌ی Next پنجره‌ی انتخاب قالب فرم نمایش داده می‌شود که پیش‌فرض آن Columnar است. (شکل 4-5).

- Columnar : هر رکورد با برچسب‌ها و فیلدهای فرم نمایش داده می‌شود. شبیه یک فرم معمولی است.

- Tabular : چندین رکورد در صفحه لیست می‌شوند. فیلدها به صورت ستونی و رکوردها در سطرها نمایش می‌یابند.

- Datasheet : رکوردها به صورت نمای datasheet جدول نشان داده می‌شوند.

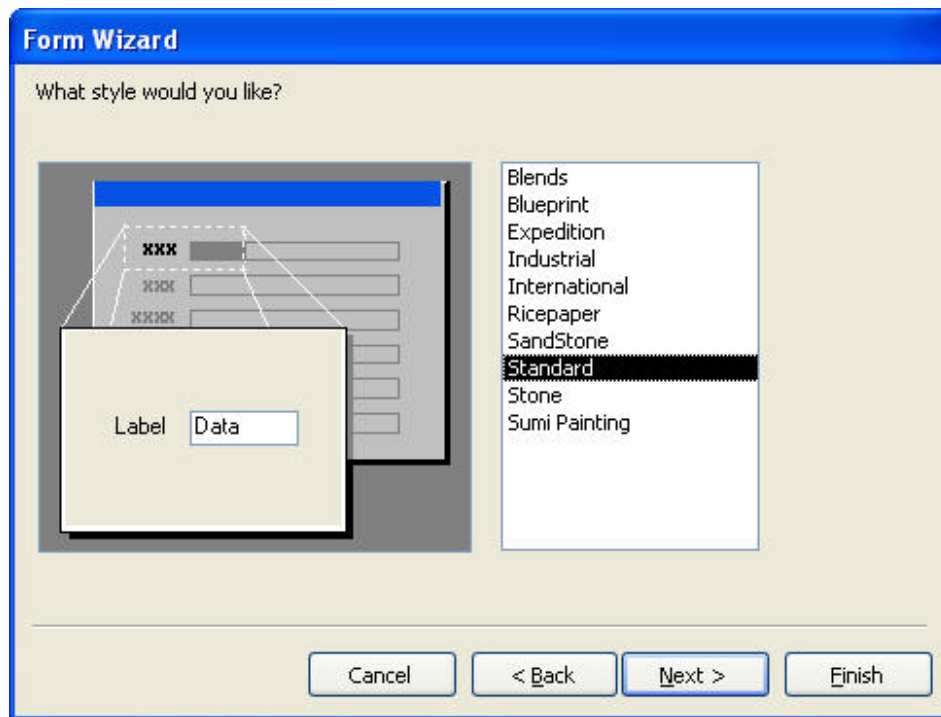


شکل 4-5 : پنجره‌ی انتخاب قالب فرم

#### کنجکاو :

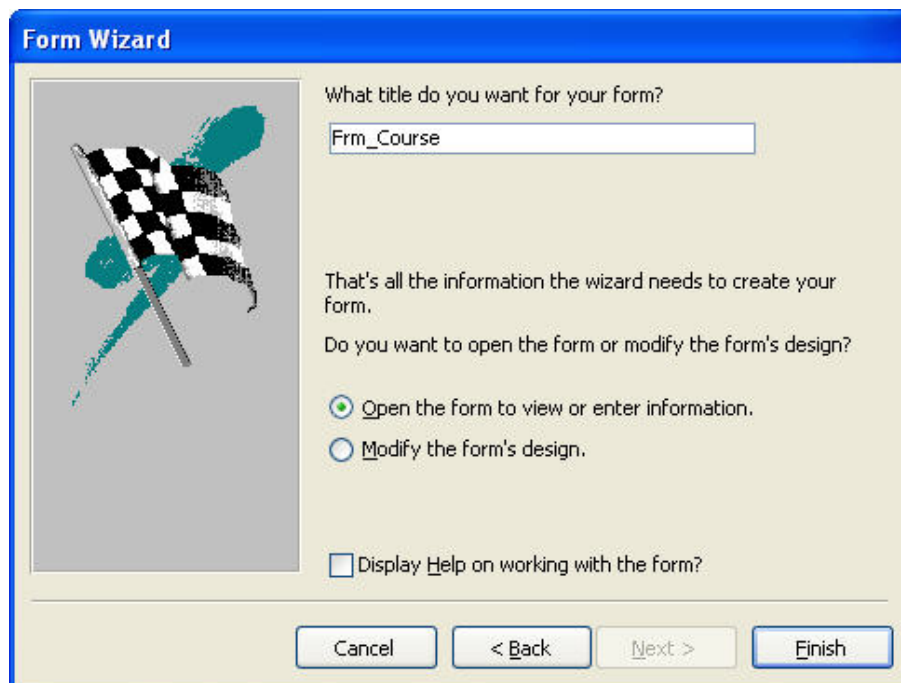
- (1) گزینه‌های Pivot Table و Pivot Chart ، چه قالبی را برای نمایش فرم در نظر می‌گیرند؟
- (2) تفاوت قالب فرم Columnnar و Justified چیست؟

- در پنجره‌های متوالی بعدی ، پیش‌فرض Standard را قبول کنید (شکل 4-6) و در پنجره‌ی آخر (شکل 4-8) نام موردنظر برای فرم را وارد نمایید و دکمه Finish را کلیک کنید. فرم ساخته شده در نمای نمایشی نشان داده می‌شود.

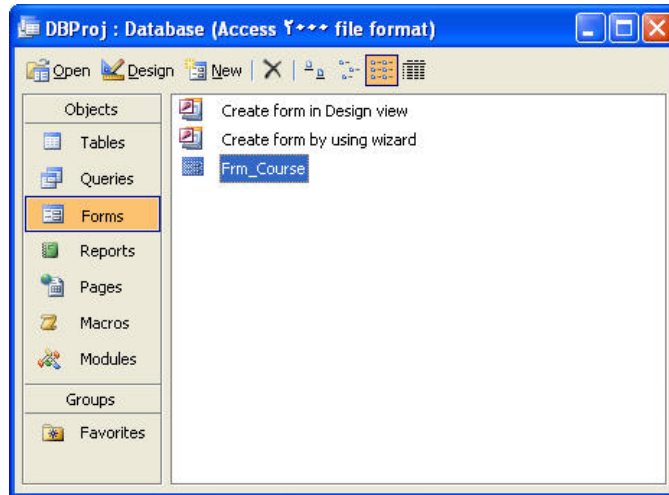


شکل 4-6 : پنجره‌ی انتخاب طرح

- با دکمه‌ی Close فرم را ببندید. در این صورت فرم ایجاد شده را در پنجره‌ی اشیاء تحت Forms مشاهده خواهید کرد (شکل 4-8).



شکل 4-7 : پنجره‌ی آخر طراحی فرم به روش Wizard



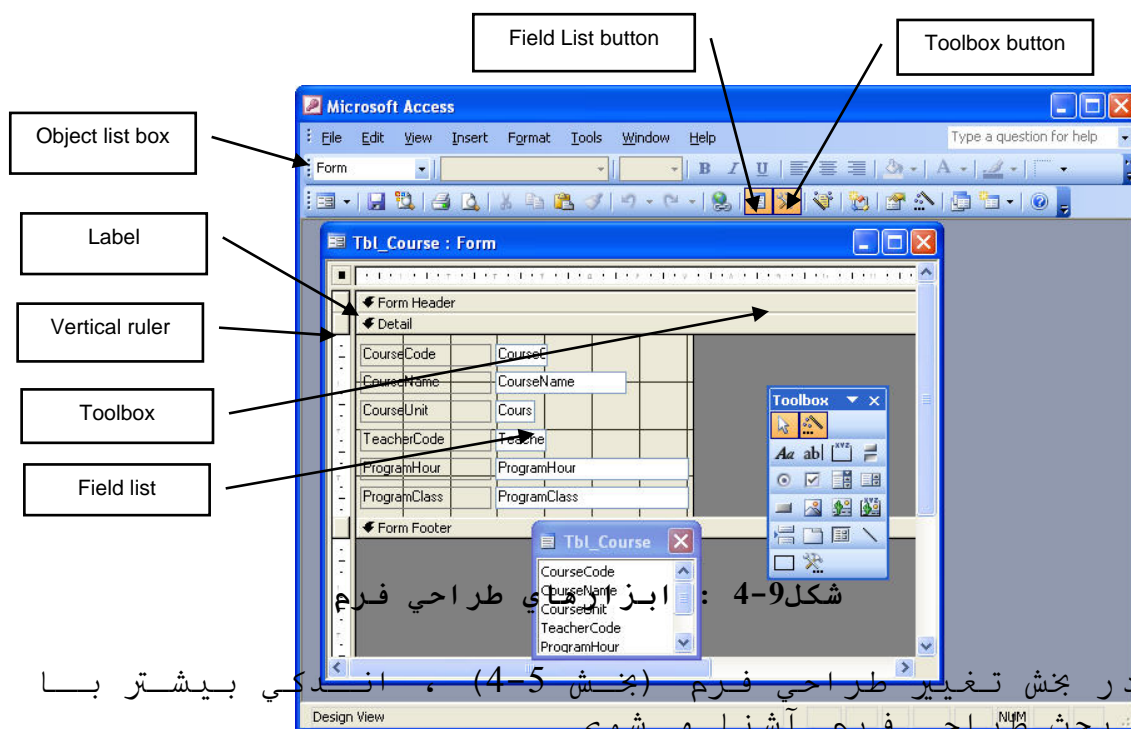
شکل 8-4 : پنجره ی اشیاء فرم

#### تمرین 4-1 :

به وسیله ی ویزارد برای جدول Tbl\_Teacher فرم Frm\_Teacher را ایجاد نمایید.  
 همچنین به کمک ویزارد ، فرمی برای جدول Tbl\_Student ایجاد کنید. نام آن را  
 Frm\_Student قرار دهید و کنترل های آن را مانند زیر تنظیم نمایید.

## 4-2-2- ایجاد فرم به وسیله Design View


در این حالت باید با استفاده از ابزارهای طراحی فرم که در شکل (4-9) می‌بینید، فرم را ایجاد کنید. کنترل متناظر و مناسب برای هر فیلد جدول را در فرم افزوده و خصوصیات آن را تغییر می‌دهید. در صورت نیاز برنامه‌نویسی نیز لازم است.

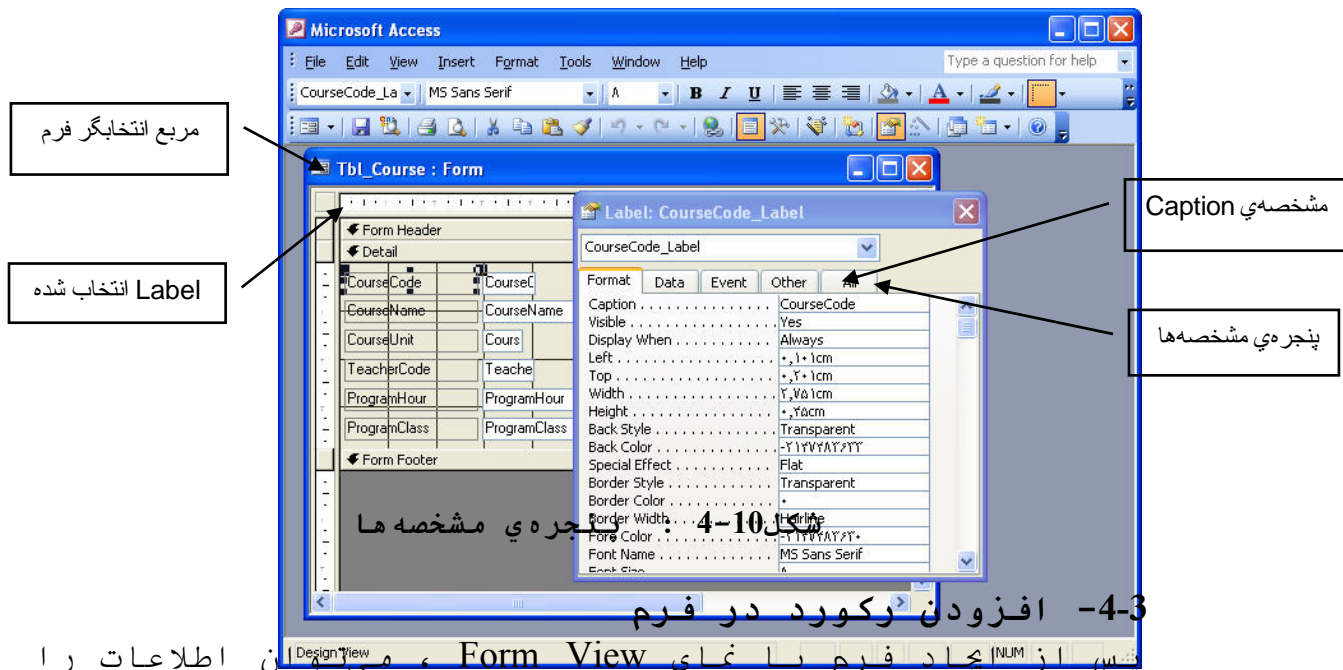


شکل 4-9: ابزارهای طراحی فرم

در بخش تغییر طراحی فرم (بخش 4-5)، از آنجایی که بیشتر با مبحث طراحی فرم آشنا می‌شویم.

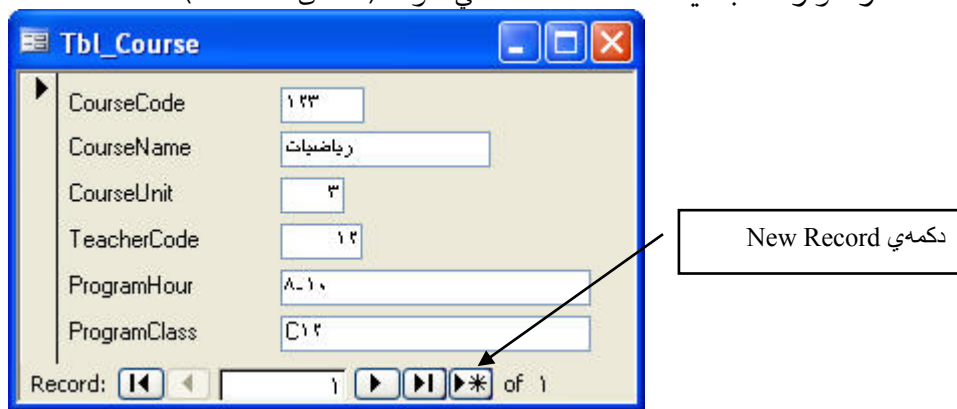
### تغییر مشخصات کنترل‌ها :

با پنجره‌ی مشخصه‌ها می‌توان مثلاً متن نمایشی در یک Label فرم را - مشخصه‌ی Caption - همان‌طور که در شکل (4-10) ملاحظه می‌شود، تغییر داد. در صورتی که پنجره‌ی مشخصه‌ها ظاهر نباشد آن را به وسیله‌ی دکمه‌ی  در نوار ابزار مشاهده کنید.



4-3- افزودن رکورد در فرم  
 پس از ایجاد فرم با نگاهی Form View ، می توان اطلاعات را در جدول متناظر فرم ثبت کرد.

با تکمیل اطلاعات بر روی کنترل های فرم ، اطلاعات به صورت یک رکورد به جدول اضافه می شود. با کلید [Tab] از کنترلی به دیگری جابه جا می شوید. زمانی که آخرین رکورد نمایش داده می شود، در صورتی که روی آخرین کنترل فرم قرار داشته باشید ، با کلید [Tab] کنترل های فرم خالی شده و اطلاعات قبلی در یک رکورد ذخیره می شود. با کلیک دکمه New Record در پایین فرم نیز کنترل های فرم خالی شده و منتظر تکمیل می ماند و با تکمیل فیلدها ، رکورد جدید اضافه می شود (شکل 4-11).



شکل 4-11 : نمونه ای از فرم

#### مثال 4-1 :

به وسیله‌ی نمای ورود اطلاعات فرم Frm\_Teacher ، اطلاعات زیر را در جدول زیر ثبت می‌کنیم.

Tbl\_Teacher :

TeacherCode	TeacherFName	TeacherLName	TeacherTel	TeacherAddress
۱۰۴۵	حامد رضا	سعدی نژاد	۰۹۱۲-۱۴۳۹۸۰	
۱۰۴۶	محمد	مهدي فر		
۱۰۴۷	علیرضا	سلجوقیان	۰۲۱-۷۷۷۶۵۵۴۰	
۱۰۴۸	کامبیز	هدایي فر	۰۹۱۲-۵۵۵۴۳۴۳	
۱۰۴۹	امیرعلی	ساسانیان		

– برای این منظور فرم Frm\_Teacher را انتخاب و دوبار کلیک نمایید. فرم به حالت ورود اطلاعات شکل زیر باز می‌شود.

- یک ردیف از اطلاعات جدول بالا را وارد کنید. دکمه‌ی را کلیک کنید. با این کار اطلاعات ثبت می‌گردد و کنترل‌های فرم خالی شود.
- ردیف‌های دیگر را وارد و همین‌گونه ادامه دهید تا تمام داده‌ها ثبت گردند.

#### 4-4- ویرایش فرم‌ها

همان‌طور که در بسته‌های نرم‌افزاری 1 و 2 دیدید در این نرم‌افزار نیز باید بتوانیم کنترل‌های روی فرم را جابه‌جا نماییم ، تغییر اندازه دهیم ، برجسب متناظر به کنترل را اصلاح کنیم ، ظاهر نمایی فرم را تنظیم کنیم ، سربرگ و پانوشت برای فرم‌ها ایجاد کنیم و برای سهولت اعمال تغییرات محیط ویرایش فرم را تنظیم نماییم . در محیط Access به راحتی می‌توانیم انواع کنترل را به یکدیگر تبدیل کنیم که در محیط‌های برنامه‌نویسی دیگر به این سهولت قابل انجام نیست.

### • تنظیمات محیط ویرایشی فرم :

- **تنظیم شبکه‌ی نقطه‌چین Grid lines** : به طور پیش‌فرض ، در حالت نمای Design روی فرم یک سری نقاط و خطوط به عنوان راهنما نشان داده می‌شود تا محل قرار گرفتن عناصر روی فرم به آسانی تنظیم شود. برای نمایش یا عدم نمایش این خصوصیت بر روی فرم از منوی View ، گزینه‌ی Grid را انتخاب کنید و برای تنظیم فاصله‌ی نمایش شبکه‌ی نقطه‌چین در حالت نمای Design فرم ، روی مربع انتخابگر فرم دوبار کلیک نمایید تا پنجره‌ی مشخصه‌های فرم نمایان شود و یا با روش قبلی این پنجره را آشکار کنید. در زبانه‌ی Format گزینه‌های GridX و GridY را تنظیم کنید.

#### کنجکاو ی :

گزینه‌ی GridX و GridY چه مقیاسی دارند و هر یک کدام بُعد مربع یعنی فاصلهی عمودی یا افقی نمایش شبکه‌ی نقطه‌چین را تغییر می‌دهند؟

- **گزینه‌ی Snap to Grid** : با انتخاب این گزینه ، کنترل‌های روی فرم حتماً باید با گوشه‌های شبکه‌ی نقطه‌چین تنظیم شوند. در این صورت زمان جابه‌جایی کنترل ، حرکت آن به صورت پرشی خواهد شد. با عدم انتخاب این گزینه ، آزادانه می‌توانید اشیاء فرم را در کوچکترین فاصله و بین فاصله‌ی مربع‌های شبکه‌ی نقطه‌چین جابه‌جا نمایید.

### • روش‌های ویرایش طراحی فرم :

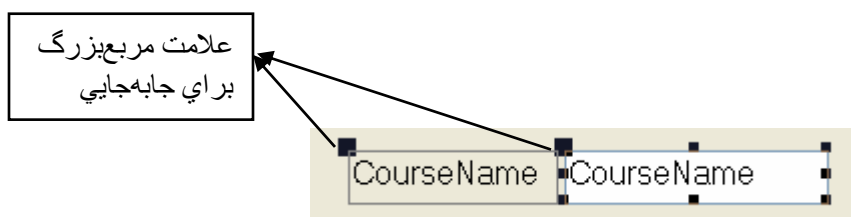
- **تغییر اندازه‌ی کنترل‌های روی فرم** : با درگ کردن به وسیله‌ی ماوس در گوشه و یا ضلع‌های یک شیء می‌توان اندازه آن را تغییر داد.

- **تغییر نوع کنترل** : به آسانی می‌توان نوع کنترل‌های فرم را بدون نیاز به حذف و ایجاد مجدد، تغییر داد. مثلاً از Text box به Combo box

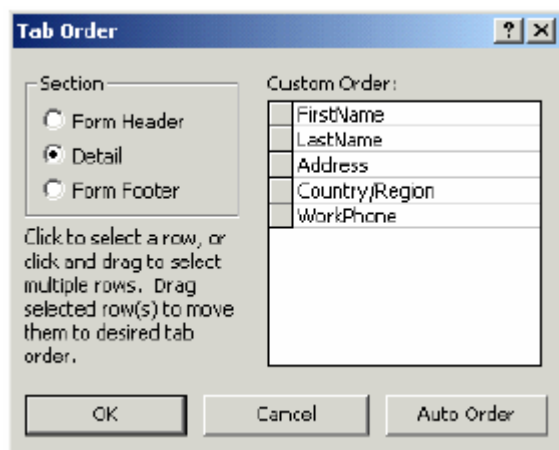


تبدیل کرد. روی شیء کلیک راست کرده و گزینه‌ی Change To را انتخاب کنید و از لیست انواع مختلف کنترل‌ها، نوع دیگری از کنترل را انتخاب کنید.

- **جابه‌جایی برچسب و کنترل و تغییر اندازه‌ی برچسب:** در Access هر شیء فرم و برچسب متناظر آن به هم متصل هستند و با جابه‌جایی هر یک، دیگری نیز جابه‌جا می‌شود. برای تغییر محل شیء و برچسب آن به صورت مجزا، روی علامت مربع بزرگ در گوشه بالای چپ کنترل/برچسب کلیک و به محل موردنظر درگ کنید.



- **ترتیب حرکت روی کنترل‌ها در زمان نمایش فرم Tab order:** با انتخاب منوی View و گزینه‌ی Tab Order می‌توانید، ترتیب جابه‌جایی روی کنترل‌ها به وسیله‌ی کلید [TAB] را در نمای ورود اطلاعات فرم مشخص کنید. در پنجره‌ی ظاهر شده با درگ کردن لیست نام کنترل‌ها و جابه‌جایی آن‌ها ترتیب مناسب را تنظیم کنید (شکل 4-12).



ترتیب  
کنترل‌ها

شکل 4-12: تغییر  
جابه‌جایی روی

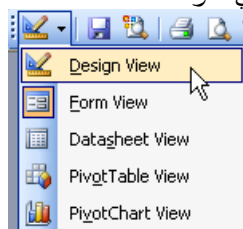
به وسیله‌ی کلید [TAB]

- **ظاهر فرم<sup>1</sup>** : رنگ زمينه ي فرم با كليك دكمه Fill/Back color در نوار ابزار Formatting و انتخاب يكي از گزينه هاي جعبه رنگ ، قابل تنظيم است. رنگ هر يك از كنترل هاي فرم به طور جداگانه به وسيله ي انتخاب رنگ موردنظر از جعبه رنگ Font/Fore Color در نوار ابزار Formatting قابل تغيير است. نوع قلم و اندازه ي آن ، ساير موارد مربوط به نوع قلم و خطوط اطراف شيء به وسيله ي نوار ابزار Formatting قابل تغيير و تنظيم است.

- **سربگ و پانوشت فرم<sup>2</sup>** : سربگ و پانوشت فقط در زمان چاپ فرم ظاهر ميشوند. به وسيله ي منوي View و گزينه ي Page Header/Footer ميتوانيد به اين بخش دسترسي داشته باشيد. نمايش شماره صفحه نيز ميتواند به اين بخشها اضافه شود ، اينكار به وسيله ي منوي Insert و گزينه ي Page Numbers... انجام ميشود. تاريخ و ساعت نيز از همين منو و به وسيله ي گزينه ي Date and Time اضافه ميشوند. براي پنهان ساختن اين بخشها در نماي Design گزينه ي Page Header/Footer را مجدداً انتخاب كنيد.

#### 4-5- تغيير طراحي فرم (Design)

براي تغيير طراحي فرم ، ابتدا فرم موردنظر را انتخاب کرده و سپس از نوار ابزار نماها (شکل 4-13) ، گزينه ي Design View را انتخاب كنيد. پنجره ي نماي طراحي نمايش داده ميشود.



شکل 4-13 : نوار ابزار نماها

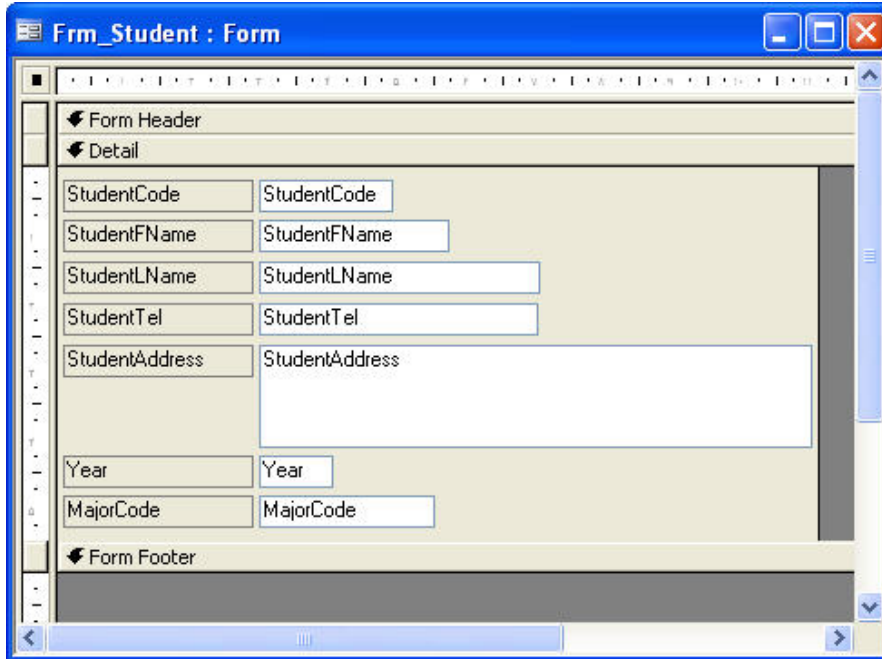
در ادامه با يك مثال طراحي فرم را تغيير مي دهيم .

<sup>1</sup> - Form Appearance

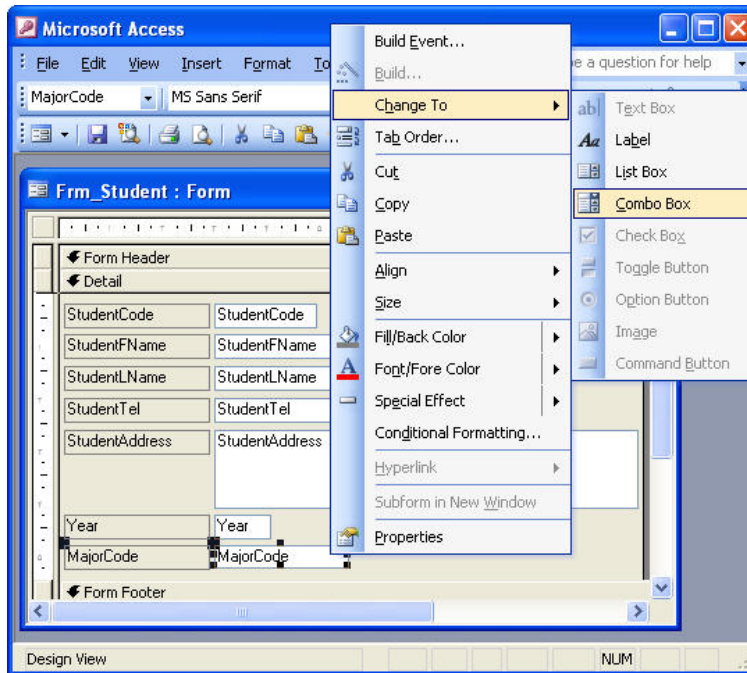
<sup>2</sup> - Page Header and Footer

## مثال 4-2 :


فرم Frm\_Student ایجاد شده به وسیله ویزارد را در نظر بگیرید. می‌خواهیم طراحی آن را بهبود دهیم. می‌خواهیم Text box مربوط به MajorCode را به Combo box تبدیل کنیم.



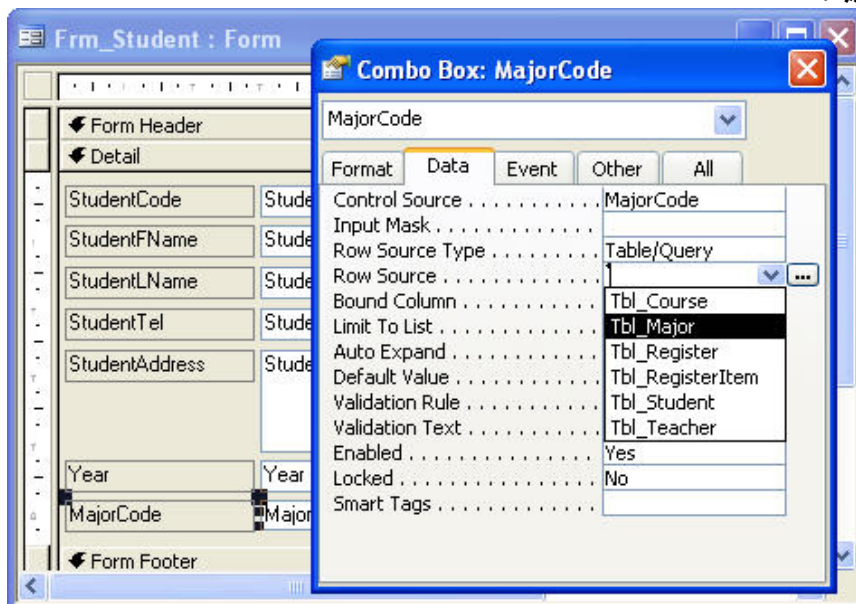
– ابتدا جعبه‌های متنی مربوط به MajorCode را انتخاب ، کلیک سمت راست ماوس را فشار دهید و گزینه‌های Combo Box را از منوی Change to انتخاب کنید (شکل زیر).



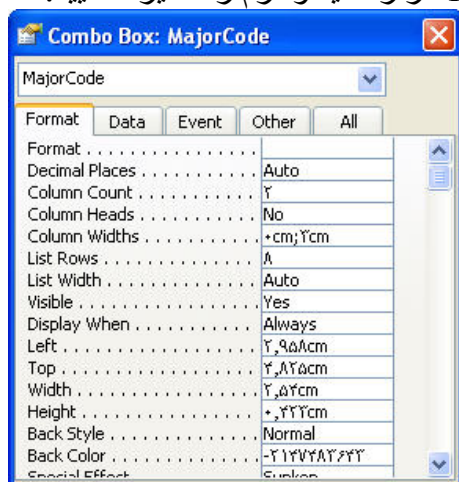
## ادامه‌ی مثال 2-4 :

– حال دوباره روی کنترل کلیک سمت راست کرده ، گزینه‌ی Properties را انتخاب کنید و یا از نوار ابزار گزینه‌ی  را کلیک کنید تا پنجره‌ی مشخصه‌ها باز شود.

– از زبانه‌ی Data در پنجره‌ی مشخصه‌ها ، مقدار Row Source را Tbl\_Degree قرار دهید (شکل زیر) . برای این کار ، می‌توانید از علامت فلش سمت راست استفاده کرده و از لیست نمایش‌یافته جدول موردنظر را انتخاب نمایید.



– در زبانه‌ی Format ، مشخصه‌ی Column Count را 2 و خصوصیت Column Width را مقدار 0cm;3cm قرار دهید . در زمان اجرا برای انتخاب فقط مقادیر موجود در این جدول‌ها خصوصیت Limit to list را در زبانه‌ی Data ، Yes قرار دهید و فرم را ذخیره نمایید.



## تمرین 2-4 :

به کمک فرم ایجاد شده اطلاعات زیر را در جدول ذخیره نمایید. البته MajorCode به جای MajorName در جدول ثبت می‌شود.

Tbl\_Student :

StudentCode	StudentFName	StudentLName	StudentTel	StudentAddress	MajorName
۸۵۰۰۱	سعید	هدایتی	۰۲۱-۸۸۹۷۵۴۵۴	خیابان دکتر فاطمی ، خیابان شهید صارمی ، پلاک ۱۴	ریاضی و فیزیک
۸۵۰۰۲	سید کلام	مهاجری	۰۲۱-۵۵۶۴۴۳۳۹	خوابان اسلام ، شهرک قدس ، آیرنمان شماره ۳۰ ، واحد ۶ شمالی	ریاضی و فیزیک
۸۵۰۰۳	رضا	میرزایی	۰۲۱-۹۹۵۴۴۵۶۵	خیابان ولیعصر ، جازم ، کوچه ناهید غربی ، پلاک ۶۹/۲	ریاضی و فیزیک
۸۵۰۰۴	محمدرضا	بخشایی	۰۲۱-۷۷۹۰۶۸۵۷	خیابان ولیعصر ، منبریه ، پلاک ۱۲۳/۴ ، طبقه ۳ ، واحد ۴	فنی حرفه ای

## تمرین 3-4 :

به کمک ویزارد فرم درس Frm\_Course را ایجاد کنید و به روش مشابه بالا ، جعبه متن TeacherCode را به Combo Box تبدیل کنید.

## مثال 3-4 :

نمای طراحی فرم دانش‌آموز ایجاد شده را به صورت فارسی تبدیل کنید. راهنمایی :

- عنوان برچسب‌ها را به فارسی تبدیل کنید. ابتدا برچسب موردنظر را انتخاب کنید و دوباره روی آن کلیک نمایید تا در حالت تغییر متن قرار گیرید و بتوانید عنوان فارسی را تایپ کنید و یا از گزینه‌ی Caption در زبانه‌ی Format پنجره‌ی مشخصه‌ها استفاده کنید.
- با کمک مربع بزرگ جابجایی برچسب ، آن‌ها را به سمت راست فرم منتقل کنید.

### ادامه‌ی مثال 3-4 :

– با کمک مربع بزرگ جابجایی متن ، مناسب با محل برچسب آن‌ها را تنظیم نمایید.

– چیدمان متن را با کمک دکمه‌های  در نوار ابزار راست‌چین قرار دهید. یا از گزینه‌ی Text Align در زبانه‌ی Format پنجره‌ی مشخصه‌ها این‌کار را انجام دهید.

– کنترل‌های فرم را با استفاده از گزینه‌ی Select All منوی Edit انتخاب و با کمک کلیدهای جهتی ، آن‌ها را به سمت راست فرم جابجا نمایید و فاصله‌ی ایجادشده را تنظیم کنید.

– فرم را

ادامه‌ی مثال 3-4 :  
 - اندازه‌ی فرم را تنظیم کنید.

#### 4-6- ایجاد فرم جزئیات در فرم اصلی (Sub Form)

تمرین : فرم `Frm_Register` و `Frm_RegisterItem` را با کمک Wizard ایجاد کنید.

در پنجره‌ی Wizard ، ابتدا در قسمت `Tables/Queries` ، جدول `Tbl_Register` را انتخاب کرده ، با کلیک دکمه‌ی `>>` تمام فیلدها را به لیست انتخابی منتقل کنید. سپس در قسمت `Tables/Queries` جدول `Tbl_RegisterItem` را انتخاب و مجدداً با دکمه‌ی `>>` تمام فیلدهای آن را انتخاب کنید. دکمه‌ی `Next` را بزنید. مجدداً `Next` و از پنجره‌ی بازشده `Tabular` را انتخاب کنید. در آخرین مرحله نام دو فرم را مقادیر `Frm_Register` و `FrmSub_RegisterItem` قرار دهید. با دکمه‌ی `Finish` شکل زیر ظاهر می‌شود.

RegisterCode: 1  
 StudentCode: A5001  
 RegisterDate: 1385/07/10  
 FmSub\_RegisterItem  
 RegisterCode CourseCode Grade  
 1 . .  
 Record: 1 of 1  
 Record: 1 of 2

همان‌طور که مشاهده می‌شود ، این نوع فرم شامل زیرفرم است. مانند تمرین‌های قبل Text Box مربوط به StudentCode و CourseCode را به Combo Box تبدیل کنید (شکل زیر).

RegisterCode: 1  
 StudentCode: سعید هدایتی  
 RegisterDate: 1385/07/10  
 FmSub\_RegisterItem  
 RegisterCode CourseCode Grade  
 1 ریاضی ۱۷  
 1 حسابان ۱۹,۵  
 1 . .  
 Record: 3 of 3  
 Record: 1 of 2

و با اجرای فرم داده‌های زیر را ثبت نمایید.

RegisterCode	StudentName	RegisterDate	CourseName	Grade
3	رضا میرزایی	۱۳۸۵/۰۷/۰۹	مکتبک	۱۹,۵
1	سعید هدایتی	۱۳۸۵/۰۷/۱۰	ریاضی ۲	۱۷
1	سعید هدایتی	۱۳۸۵/۰۷/۱۰	حسابان	۱۹,۵
2	سید کلثم مهاجری	۱۳۸۵/۰۷/۱۱	معلومات دینی	۰

با کمک همین امکانات ذکرشده ، می‌توانید فرم‌های موردنظر خود را ایجاد کنید. به‌خاطر داشته باشید که برای ایجاد فرم‌های پیچیده‌تر و هوشمندتر که اطلاعات واردشده‌ی کاربر را کنترل کنند ، داده‌ها را برای ذخیره آماده سازند و ... باید برنامه‌نویسی کنید. در این صورت زبان برنامه‌نویسی Visual Basic به شما کمک خواهد کرد ، چرا که بسیاری از دستورات آن در محیط برنامه‌نویسی Access نیز قابل اجرا و استفاده است.





## خلاصه ي فصل

فرم يکي از ابزارهاي سودمند براي ثبت داده ها در جدول است. فرم ها پنجره هايي شامل کنترل ها هستند که کاربر مي تواند در آن ها مقدار فيلد را تايپ کند يا از ليست نمايش يافته مورد ي را انتخاب کند و يا از کنترل هاي ديگر به آساني داده ها را وارد نمايد تا به صورت يك رکورد در جدول متناظر فرم ثبت شود.

با کمک ابزار جادويي Form Wizard مي توان يك فرم ايجاد کرد. طراحي اين فرم در نماي Design View قابل تغيير است. اگر با زبان برنامه نويسي Visual Basic و محيط آن آشنا باشيد ، ايجاد يك فرم در اين دو محيط مشابه است. کنترل هاي دو محيط مشخصه هاي متناظري دارند که ما از ذکر آن ها در اينجا خودداري مي کنيم.

پس از ايجاد فرم با کمک امکانات نماي Form View مي توانيم رکوردهايي را در جدول متناظر ثبت ، ويرايش و يا حذف نماييم.

## خود آزمايي

1. کنترل مقيد و نامقيد چه تفاوت ي دارند و چگونه تعريف مي شوند؟
2. مراحل ايجاد فرم به وسيله ي Form Wizard را توضيح دهيد؟
3. امکانات نماي Design View (نوار ابزار ، جعبه ابزار ، ليست فيلدها و ...) را به اختصار توضيح دهيد.
4. نکات مفيد در ويرايش فرم ها را به اختصار بيان کنيد.
5. در صورتي که نخواهيد از Form Wizard استفاده کنيد ، آيا مي توانيد Combo box اي روي فرم ايجاد کنيد که مقادير جدول Tbl\_Degree را نشان دهد؟

(راهنمایی : مشابه همان تغییراتی که در تبدیل نوع کنترل از Text box به Combo box دیدید)

## 5- فصل پنجم : ایجاد و کار با پرسوجوها

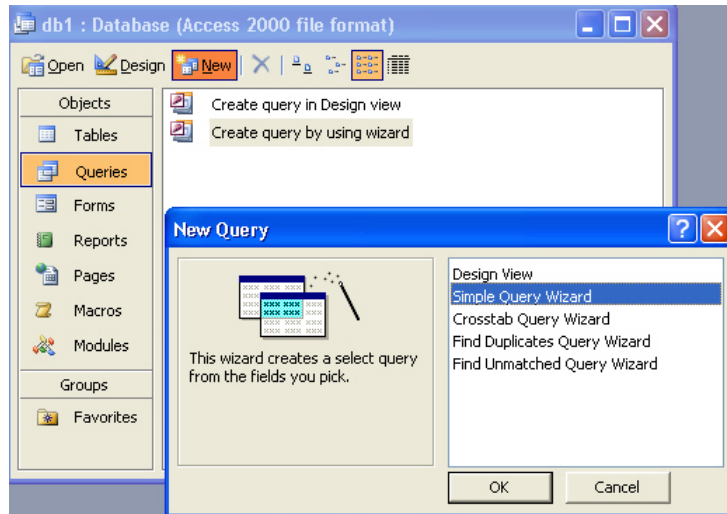
در Access جدول ، قسمت اصلی نگهدارنده داده ها است و پرسوجو برای بازیافت اطلاعات از این جدولها دارای اهمیت است. پرسوجوها انواع مختلف دارند ولی معمولترین آن ، پرسوجویی است که برای یافتن زیرمجموعه ای از رکوردهای دارای شرط یا شرایط خاص ، استفاده میشود. به این نوع پرسوجو Select گفته میشود. پرسوجوهای دیگر ، برای انجام عملیات روی مقادیر داده ها استفاده میشوند. همانطور که پیشتر گفتیم ، برای نمایش داده ها در فرمها و گزارشها معمولا از پرسوجوهای از نوع Select استفاده میشود.

### در پایان این فصل انتظار میرود که هنرجو بتواند :

- به وسیله ی گزینه ی Query Wizard یک پرسوجوی بازیافت اطلاعات Select ایجاد کند.
- به وسیله ی Design View ، پرسوجو ایجاد کند و یا پرسوجوی موجود را تغییر دهد.
- نتیجه ی پرسوجو را مرتب کند.
- پرسوجوی شرطی ایجاد کند. برای محدود کردن رکوردهای نتیجه شرط اعمال کند.
- فیلدهای غیرجدولی (محاسباتی) به پرسوجو اضافه کند و از Expression Builder برای این منظور استفاده کند.
- با پرسوجوهای عملیاتی آشنا شده و آنها را ایجاد کند.
- با زبان ساختیافته SQL آشنا میشود.

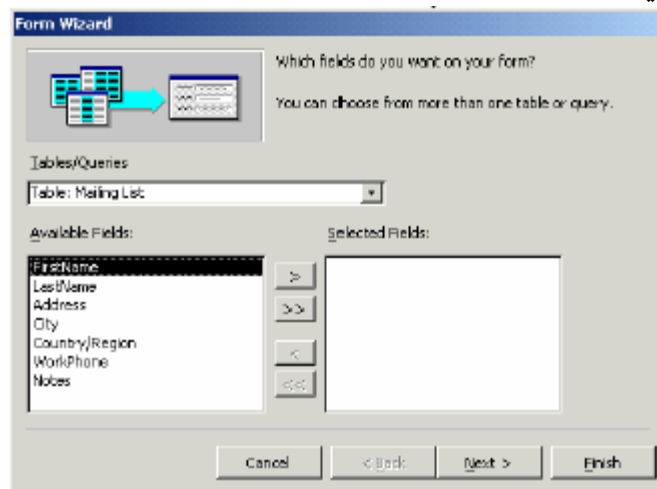
### 5-1- ایجاد پرسوجو به وسیله ی Simple Query Wizard

- گزینه ی Queries را از لیست اشیاء در پنجره ی پایگاه داده انتخاب کرده ، سپس Create Query by using wizard را دوبار کلیک نمایید (شکل 5-1).



شکل 1-5 : پنجره انتخاب ایجاد پرسوجو

مشابه پنجره انتخاب فیلد فرم است. در لیست Tables/Queries این پنجره جدول موردنظر را انتخاب کنید و از لیست فیلدهای در دسترس در: Available Fields ، فیلد موردنظر خود برای ایجاد پرسوجو را با دکمه‌های >> یا > به فیلدهای انتخاب شده: Selected Fields اضافه کنید. می‌توانید جدول دیگری را انتخاب و فیلدهای مربوط از آن را نیز به لیست فیلدهای انتخابی اضافه کنید. پس از آن با کلیک Next ، نام پرسوجو را وارد کرده ، Finish نمایید.

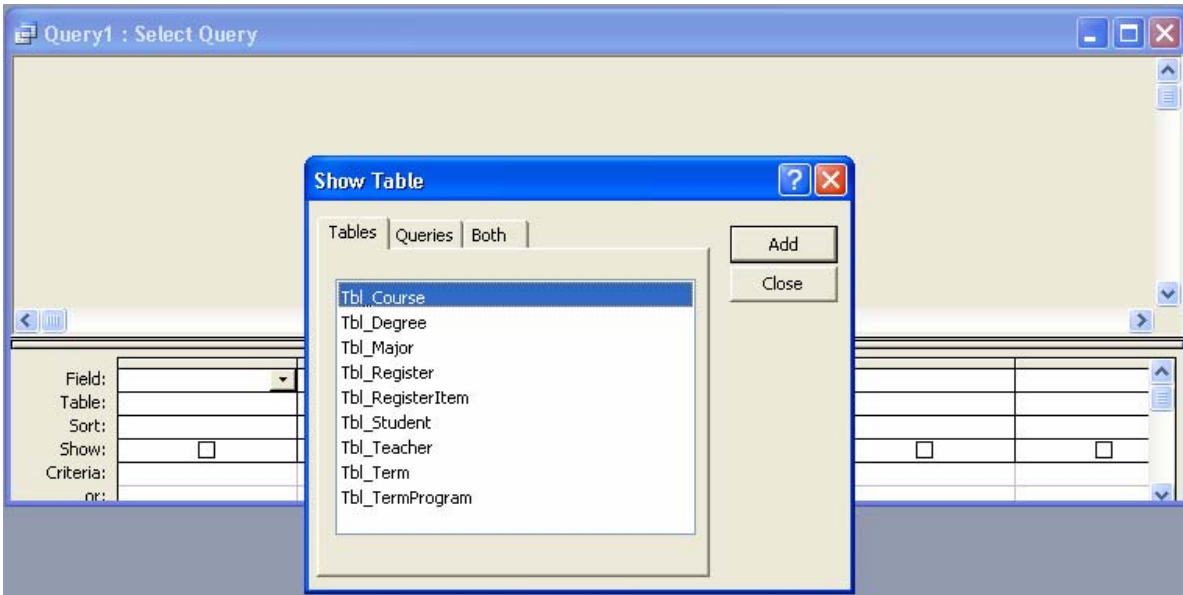


انتخاب فیلدهای  
برای نمایش در

شکل 2-5 :  
موردنیاز  
پرسوجو

## 5-2- ایجاد پرسوجو به وسیله Design View

- گزینه Queries را از لیست اشیاء در پنجره پایگاه داده انتخاب کرده ، سپس دکمه New Command را در نوار دکمه های فرمان کلیک کنید و گزینه Design View را انتخاب نمایید. پنجره زیر باز می شود (شکل 5-3). با انجام مراحل بعدی می توانید جدول ها ،



فیلدها و شرایط مورد نظر پرسوجو را ایجاد نمایید.

شکل 5-3 : پنجره انتخاب جدول در طراحی پرسوجو

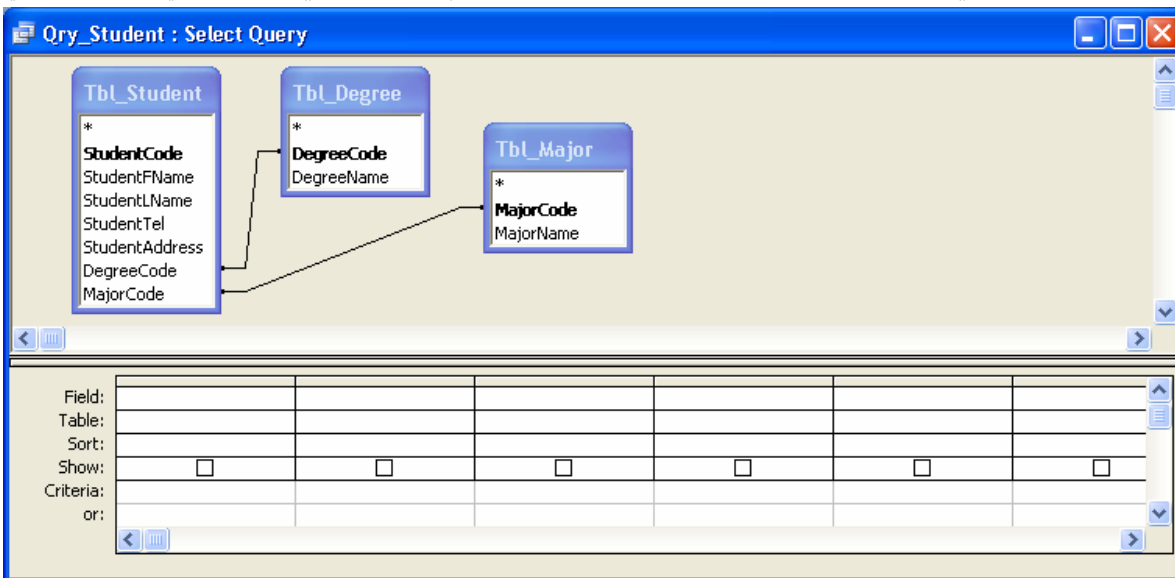
### کنجکاو :

غیر از دو گزینه Design View و Query Wizard سایر گزینه های شکل 5-1 چه عملی انجام می دهند؟

## 5-3- استفاده از جدول در پرسوجو


با انتخاب Design ، کادر محاوره ای با عنوان Show Table نمایان می شود (شکل 5-3). در این کادر امکان انتخاب جدول ، پرسوجو های ایجاد شده و یا ترکیبی از آن ها وجود دارد. در واقع می توانید از جدول های تعریف شده و یا سایر پرسوجو های موجود برای ایجاد یک پرسوجوی جدید استفاده کنید.

با دو بار کلیک روی نام جدول یا انتخاب جدول و کلیک دکمه **Add** ، آن را به پرسوجو اضافه می‌کنید. با استفاده از **Shift** می‌توانید جدول‌های مجاور و با استفاده از **Ctrl** می‌توانید جدول‌های غیرهم‌جوار را همزمان انتخاب نموده و با کلیک **Add** آن‌ها را به پرسوجو اضافه کنید. پس از افزودن جدول‌ها و پرسوجوهای موردنظر پنجره‌ی محاوره‌ای را با **Close** ببندید تا پنجره‌ی اصلی طراحی



پرسوجو (Query Design) (شکل 4-5) ظاهر شود.  
شکل 4-5 : پنجره‌ی اصلی طراحی پرسوجو

#### کنجکاو ی :

آیا پس از بستن پنجره‌ی انتخاب جدول ، راهی برای افزودن جدول جدید به پرسوجو وجود دارد؟  
راهنمایی : پرسوجو را در حالت **Design** باز کرده و با دکمه‌ی  جدول‌های جدید موردنیاز را اضافه کنید.

#### 4-5- انتخاب فیلدهای جدول‌ها/پرسوجوهای اضافه‌شده موردنیاز پرسوجوی جدید

پس از افزودن جدول یا جدول‌های موردنظر و یا پرسوجوهای قبلی ، باید فیلدهای موردنیاز را معین کنیم .

همان‌طور که در شکل 4-5 مشاهده می‌شود. دو قسمت در پنجره‌ی طراحی پرسوجو قابل توجه است :

- نیمه بالایی : در این قسمت جدول‌ها یا پرسوجوهای نمایش داده می‌شوند که در مرحله قبل اضافه کرده‌ایم


و می‌خواهیم از آن‌ها در ایجاد پرسوجوی جدید استفاده کنیم .

• نیمه پایینی : فیلدهایی که می‌خواهیم در خروجی پرسوجو مشاهده شوند در این قسمت معین می‌شود.  
برای انتخاب فیلدها از یکی از راه‌های زیر استفاده می‌کنیم :

- 1- دوبار کلیک روی نام فیلد در نیمه بالایی.
- 2- درگ کردن فیلد از نیمه بالایی به نیمه پایینی.
- 3- استفاده از Shift یا Ctrl.
- 4- دوبار کلیک روی علامت \* در لیست فیلدها برای انتخاب همه‌ی فیلدها .

#### کنجکاوئ :

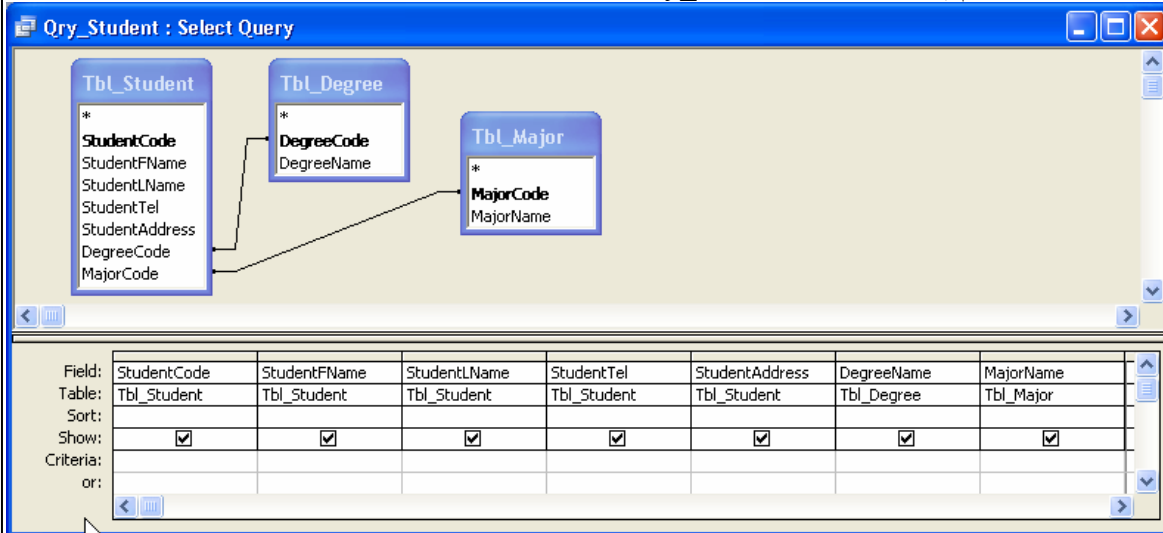
1. دوفیلد نمونه از یک جدول را با دو بار کلیک به نیمه‌ی پایینی اضافه کنید. پس از آن علامت \* را انتخاب کنید. به‌وسیله‌ی \* تمام فیلدها در خروجی ظاهر می‌شوند . دوفیلد قبلی نیز فیلدهای مشابهی در خروجی دارند ، این فیلدها چگونه و با چه نامی نمایش داده می‌شوند؟
2. چگونه از Shift یا Ctrl برای انتخاب فیلدها استفاده می‌شود ؟

پس از ایجاد پرسوجو ، آن را با یک نام ذخیره کنید. یکی از ساده‌ترین راه‌های اجرای پرسوجو استفاده از علامت  در نوار ابزار است (دکمه‌ی Run). نتیجه‌ی پرسوجو به صورت جدولی نمایش داده می‌شود.



### مثال 5-1 :

سه جدول Tbl\_Student ، Tbl\_Degree و Tbl\_Major را به پرسوجو اضافه کنید. فیلدهای زیر را انتخاب نمایید و نام پرسوجو را Qry\_Student بگذارید.



### ادامه‌ی مثال 5-1 :

با اجرای آن ، نتیجه را به صورت شکل زیر خواهید دید.

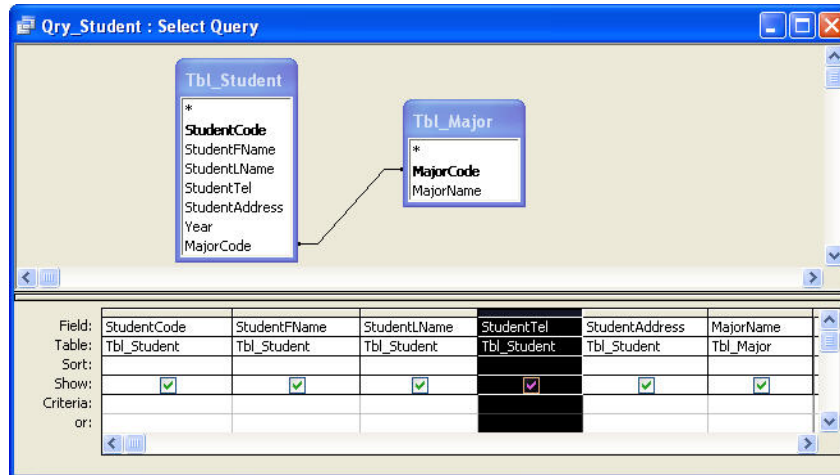
StudentCode	StudentFName	StudentLName	StudentTel	StudentAddress	DegreeName	MajorName
۸۵۰۰۱	سعید	هدایتی	۰۲۱-۸۸۹۷۵۴۵۴	۱۴ بلاک ۱۴ خیلان شهید صرامی ، بلاک ۱۴	اول دبیرستان خیلان دکتر فلسفی ، خیلان شهید صرامی ، بلاک ۱۴	ریاضی و فیزیک
۸۵۰۰۲	سید کاظم	مهاجری	۰۲۱-۵۵۶۴۴۳۳۹	۳۰ واحد ۶ شمالی	اول دبیرستان شهرک قدس ، آیرنشان شماره ۳۰ ، واحد ۶ شمالی	علوم تجربی
۸۵۰۰۳	رضا	میرزایی	۰۲۱-۹۹۵۴۴۵۶۵	۶۹/۲	اول دبیرستان ولیعصر ، جام جم ، کوچه ناهید غربی ، بلاک ۶۹/۲	علوم انسانی
۸۵۰۰۴	محمدرضا	بنمایی	۰۲۱-۷۷۹۰۶۸۵۷	۴	اول دبیرستان ولیعصر ، منزیه ، بلاک ۱۲۳/۴ ، طبقه ۳ ، واحد ۴	فنی حرفه ای
۸۵۰۰۵	محمد علی	بنمایی	۰۲۱-۷۷۹۰۶۸۵۷			فنی حرفه ای

### 5-5- حذف فیلد از پرسوجو

برای حذف فیلد از پرسوجو ، ستون متناظر را از نیمه پایینی پنجره‌ی طراحی پرسوجو انتخاب و سپس دکمه‌ی Delete را می‌زنیم. به این ترتیب فیلد مذکور از نتیجه‌ی پرسوجو حذف می‌شود.

## مثال 2-5 :

در پرسوجوي Qry\_Student ، فيلد شماره تلفن را حذف كنيد .  
- مطابق شكل زير ستون StudentTel را انتخاب كنيد ، سپس دكمه delete صفحه كليك را كليك كنيد .



## كنجكاوي :

بدون حذف فيلد به روش بالا نيز مي توان كاري كرد كه فيلد در نتيجه ي خروجي نمايش داده نشود ، به چه صورتي اين كار انجام مي شود؟  
راهنمايي : علامت Checkbox ستون موردنظر در نيمه ي پاييني را حذف كنيد يا ستون موردنظر را در پنجره ي نتيجه و خروجي . پنهان كنيد .

## نكته :

- 1) هنگامي كه در نماي Design هستيم ، مي توانيم با انتخاب يك ستون و درگ كردن آن محل قرار گرفتن فيلد در پنجره ي خروجي نتيجه را تغيير دهيم .
- 2) براي ذخيره سازي پرسوجو از Save استفاده مي كنيم . Access به صورت پيش فرض نام هاي Query1 ، Query2 و ... را پيشنهاد مي كند . بهتر است نام مناسبی براي پرسوجو انتخاب شود . مثلاً ابتداي نام پرسوجو را با Qry شروع كنيد .

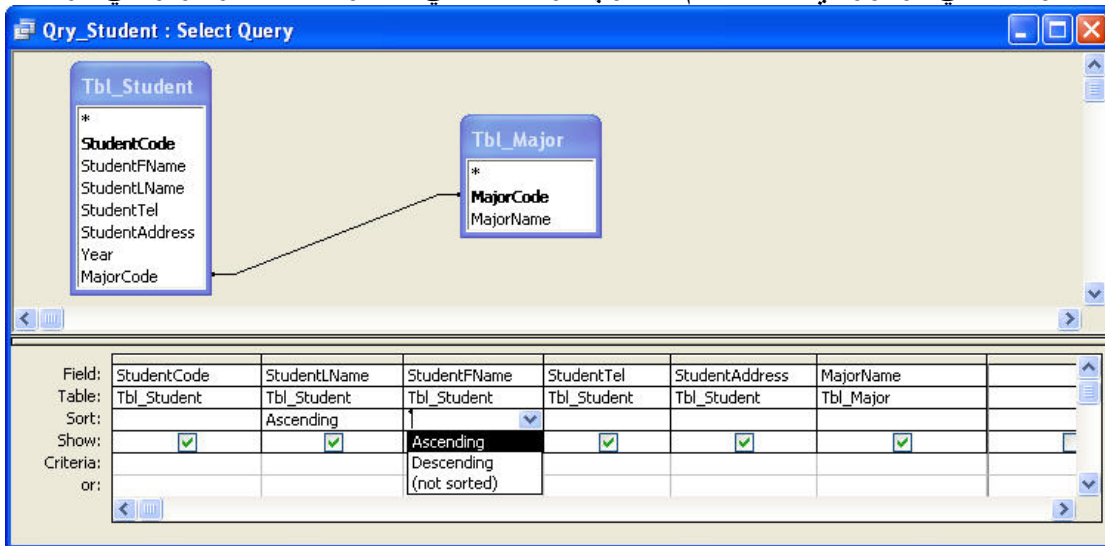
## 5-6- مرتب سازي نتيجه ي پرسوجو

پس از اجرای پرسوجو ، رکوردهای نتیجه به ترتیب معینی ظاهر نمی‌شوند. اگر بخواهیم نتیجه بر اساس صعودی یا نزولی مقادیر یک فیلد ظاهر شود در ستون فیلد موردنظر درنیمه پایینی و درسطر **Sort** مقدار **Ascending** (صعودی) و یا **Descending** (نزولی) را انتخاب کنید.

### مثال 3-5 :

در پرسوجوی Qry\_Student ، نتیجه‌ی پرسوجو بر اساس صعودی مقادیر نام خانوادگی و در صورت تساوی نام خانوادگی بر اساس صعودی نام دانش‌آموز مرتب کنید.

- برای این کار می‌توانیم مطابق شکل زیر در ستون فیلد StudentLName در سطر Sort مقدار Ascending را برگزینیم و در ستون فیلد StudentFName در سطر Sort مقدار Ascending را انتخاب می‌کنیم. در این صورت همان‌طور که در نتیجه شکل زیر می‌بینید ، بین دو 'یغمایی' رکوردی که فیلد نام آن کوچکتر باشد یعنی "علیرضا" بالاتر قرار می‌گیرد.



- نتیجه به صورت زیر خواهد بود :

StudentCode	StudentLName	StudentFName	StudentTel	StudentAddress	MajorName
85002	مهاجری	سید کلتم	021-55644339	آیرئمان شماره 30 ، واحد 6 شمالی	ریلنسی و فیزیک
85003	میرزایی	رضا	021-99544565	خیابان ولیعصر ، جام جم ، کوچه ناهید غربی ، بلاک 2/69	ریلنسی و فیزیک
85001	هدایتی	سعید	021-88975454	خیابان دکتر فاطمی ، خیابان شهید صارمی ، بلاک 14	ریلنسی و فیزیک
85005	یغمایی	علیرضا	021-55902232	فلکه نوم صلاویه ، بلاک 123/4 ، طبقه اول	فنی حرفه ای
85004	یغمایی	محمدرضا	021-77906857	خیابان ولیعصر ، منیریه ، بلاک 123/4 ، طبقه 3 ، واحد 4	فنی حرفه ای

- اگر ستون فیلد StudentFName سمت چپ ستون فیلد StudentLName قرار دهیم ، نتیجه به صورت زیر خواهد بود :

StudentCode	StudentFName	StudentLName	StudentTel	StudentAddress	MajorName
85003	رضا	میرزایی	021-99544565	خیابان ولیعصر ، جام جم ، کوچه ناهید غربی ، بلاک 2/69	ریلنسی و فیزیک
85001	سعید	هدایتی	021-88975454	خیابان دکتر فاطمی ، خیابان شهید صارمی ، بلاک 14	ریلنسی و فیزیک
85002	سید کلتم	مهاجری	021-55644339	آیرئمان شماره 30 ، واحد 6 شمالی	ریلنسی و فیزیک
85005	علیرضا	یغمایی	021-55902232	فلکه نوم صلاویه ، بلاک 123/4 ، طبقه اول	فنی حرفه ای
85004	محمدرضا	یغمایی	021-77906857	خیابان ولیعصر ، منیریه ، بلاک 123/4 ، طبقه 3 ، واحد 4	فنی حرفه ای

**نکته :**

برای مرتب‌سازی با بیش از یک ستون باید ابتدا ستون‌های موردنظر را در نیمه‌ی پایینی پنجره‌ی طراحی از چپ به راست و بر اساس نظم موردنظر قرار دهیم. مثلاً در مثال بالا ستون فیلد StudentLName سمت چپ ستون فیلد StudentFName قرار گرفته باشد. همانطور که قبلاً گفتیم، محل قرارگرفتن ستون فیلد با درگ کردن جابه‌جا می‌شود. همیشه روش مرتب‌سازی به ترتیب قرارگرفتن فیلدها، از سمت چپ به راست است.

**5-7- معیارها<sup>1</sup> در پرسوجو**

با استفاده از ردیف Criteria در نیمه‌ی پایینی، می‌توانیم شرط‌هایی را به پرسوجو اعمال کنیم. در شرط می‌توان از عملگرهای مختلفی استفاده کرد که در جدول شماره (5-1) قابل مشاهده است.

مثال	مفهوم	عملگر
= "یغمایی"	مساوی	=
< 20	کوچکتر از	<
> 20	بزرگتر از	>
<= 20	کوچکتر یا مساوی	<=
>= 20	بزرگتر یا مساوی	>=
<> 20	نا مساوی	<>
"هدایتی" and "یغمایی"	و (شرطها باید درست باشند)	And
"هدایتی" or "یغمایی"	یا (حداقل یکی از شرطها باید درست باشد)	Or
Like "محمد"	مانند	Like
Between 5 and 20	بین [دو مقدار]	Between
In "یغمایی", "هدایتی"	در (مانند Or)	In

جدول 5-1 : عملگرهای شرطی

**5-8- استفاده از تاریخ در معیارها**

<sup>1</sup> - Criteria

همان طور که دیدیم می‌توانیم از عملگرها در پرسوجو استفاده کنیم. به غیر از عملگرها از توابع مثل تابع‌های مربوط به تاریخ نیز می‌توان استفاده کرد. در جدول شماره (2-5) عبارتهای مجاز تاریخ را می‌توانید مشاهده کنید.

عبارت	مفهوم	مثال
Date	تاریخ جاری سیستم به میلادی	<> Date() مقدار فیلد مثلا ثبتنام برابر تاریخ امروز نباشد.
Day (Date)	روز	Day(RegDate) >= 15 تاریخ ثبتنام از نیمه ماه گذشته باشد.
Month(Date)	ماه	>= Month(Date)-2 مقدار فیلد مثلا ثبتنام از دو ماه قبل به اینطرف باشد.
Year(Date)	سال	Year(RegDate) = 1385 سال ثبتنام 1385 باشد.
Weekday(Date)	روز هفته	WeekDay(RegDate)=1 روز ثبتنام ، شنبه باشد.
Between Date And Date	محدوده ای از تاریخ	Between #1/1/98# and #12/31/98# تاریخ ثبتنام بین این دو باشد.
DatePart(Interval,Date)	قسمت معینی از تاریخ مقادیر Interval عبارت است از :	DatePart("m",RegDate) = 5 ماه ثبتنام برابر ماه مه میلادی (حدود اردیبهشت ماه) باشد.

مقدار	توضیحات
yyyy	سال
q	فصل
m	ماه
y	تعداد روزهای
d	روز
w	روز هفته
ww	هفته
h	ساعت
n	دقیقه
s	ثانیه

جدول 2-5 : تابع‌های مربوط به تاریخ

#### مثال 5-4 :

دانش آموزان ثبت نام شده در رشته‌ی ریاضی فیزیک را به دست آورید. پرس و جوی Qry\_Student را باز کنید. این پرس و جو دانش آموزان ثبت نام شده را بر می گرداند. اگر بخواهیم فقط دانش آموزان رشته‌ی "ریاضی فیزیک" را ببینیم، شرط MajorCode=1 را به صورت شکل زیر در ستون MajorCode اعمال می کنیم. با انتخاب گزینه‌ی Save As... از منوی File، آن را به نام Qry\_StudentMath ذخیره کنید.

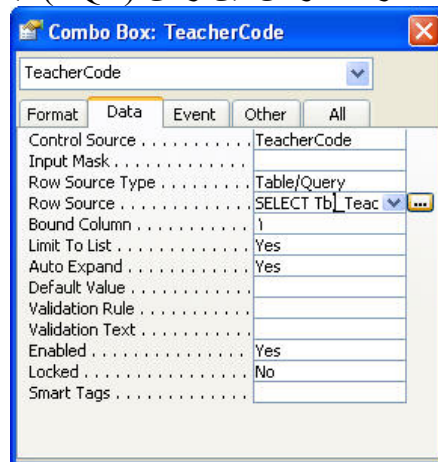
Field:	StudentCode	StudentFName	StudentLName	StudentTel	StudentAddr	DegreeName	MajorName	MajorCode
Table:	Tbl_Student	Tbl_Student	Tbl_Student	Tbl_Student	Tbl_Student	Tbl_Degree	Tbl_Major	Tbl_Major
Sort:								
Show:	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Criteria:								=1
or:								

#### نکته :

همان طور که در شکل بالا می بینید، مربع بالای شرط انتخاب نشده است، چرا؟ راهنمایی: پرس و جو را اجرا کنید، آیا در نتیجه‌ی پرس و جو، کد رشته مشاهده می شود؟

#### مثال 5-5 :

در فرم Frm\_Course، Combo Box نام معلم را به گونه‌ای تغییر دهید که نام و نام خانوادگی به صورت یک متن نمایش یابد و به نظر نرسد دو فیلد مجزا برای نگهداری آن‌ها وجود دارد. در پنجره‌ی مشخصه‌ها، در زبانه‌ی Data، گزینه‌ی Row Source به جای انتخاب Tbl\_Tecaher، عبارت زیر را که زبان استاندارد ساختیافته‌ی کار با پایگاه داده است تایپ نمایید: در پایان این فصل در خصوص این زبان (SQL) بیشتر صحبت می کنیم.

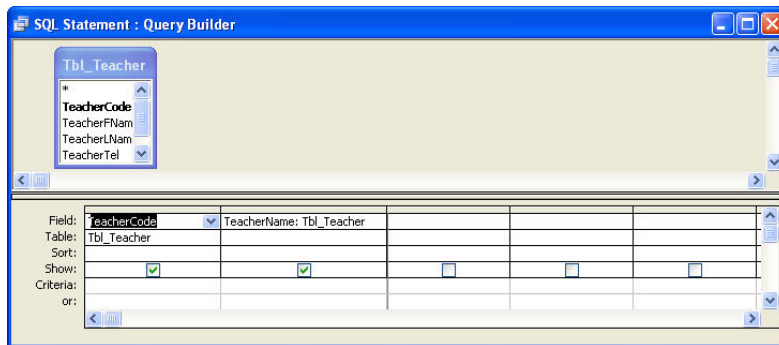



## ادامه‌ی مثال 5-5 :

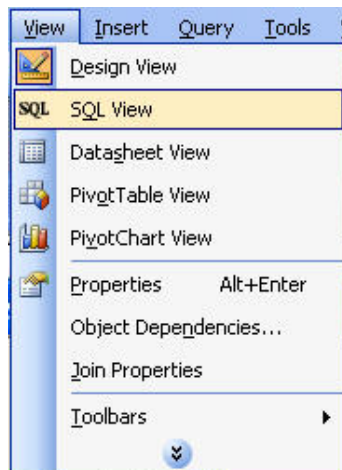
```
SELECT.....TecaheCode, [TecaheFName] + '.....' + [TecaheLName]..... AS.....
TecaheName, TecaheLName.....FROM.....Tbl_Tecahe;
```

☆ دقت: هر يك ..... از فاصله‌ها در عبارت فوق ، حداقل يك كاراكثر Blank است. ساير فاصله‌ها براي خوانايي است و مي‌تواند وجود نداشته باشد.

**دقت :** به‌جاي تايپ عبارت SQL مي‌توانيد علامت سه نقطه‌ي کنار آن را كليك كنيد ، محيطي مشابه محيط طراحي پرسوجو ظاهر مي‌شود (شكل زير) كه مي‌توانيد جدول‌هاي Tbl\_Course و Tbl\_Teacher را اضافه كنيد و مانند محيط پرسوجو ، فيلدهاي موردنظر يعني TeacherCode ، TeacherFName و TeacherLName را انتخاب كنيد كه در نيمه‌ي پاييني پنجره‌ي طراحي مشاهده شوند.



از دكمه‌ي View  نوارابزار (شكل زير) ، مي‌توان SQL View را انتخاب كرد و Select مربوطه را مشاهده كرد كه به صورت زير است :





ادامه‌ی مثال 5-5 :

```
SQL Statement : Query Builder
SELECT Tbl_Teacher.TeacherCode, Tbl_Teacher.TeacherFName, Tbl_Teacher.TeacherLName
FROM Tbl_Teacher;
```

حال آن را به صورت زیر تصحیح کنید :

```
SQL Statement : Query Builder
SELECT Tbl_Teacher.TeacherCode, Tbl_Teacher.TeacherFName, Tbl_Teacher.TeacherLName
FROM Tbl_Teacher;
```

**دقت:** SELECT Tbl\_Teacher.TeacherCode, Tbl\_Teacher.TeacherFName + ' ' + Tbl\_Teacher.TeacherLName AS TeacherName FROM Tbl\_Teacher ;

با بستن پنجره‌ی پرسوجو ، SQL مربوط به آن در Row Source قرار می‌گیرد . سپس با اجرای فرم ، داده‌های زیر را به کمک آن ذخیره نمایید :

Query۲ : Select Query

CourseCode	CourseName	CourseUnit	TeacherName	ProgramHour	ProgramClass
۴۰۳	حسابان	۳	حامدرضا سعیدی نژاد	۱۰:۳۰-۱۲:۰۰	کلاس C۱۰
۵۰۱	فران	۲	حامدرضا سعیدی نژاد	۷:۳۰-۹:۰۰	کلاس D۱۵
۴۰۲	ریاضی ۲	۲	علیرضا میرزاپور	۹:۰۰-۱۰:۳۰	کلاس C۱۲
۶۰۱	فیزیک	۴	علیرضا میرزاپور	۱۵:۰۰-۱۶:۳۰	کلاس C۱۲
۷۰۲	تربیت بدنی ۲	۲	علیرضا میرزاپور	۱۳:۳۰-۱۵:۰۰	کلاس C۱۰
۵۰۲	معلومات دینی	۳	علیرضا میرزاپور	۱۳:۳۰-۱۵:۰۰	کلاس C۱۱
۴۰۱	ریاضی ۱	۲	کامبیز هدایتی فر	۹:۰۰-۱۰:۳۰	کلاس C۱۰
۶۰۲	مکانیک	۲	کامبیز هدایتی فر	۷:۳۰-۹:۰۰	کلاس A۱۰
۷۰۱	تربیت بدنی ۱	۲	امیرعلی حسینی	۹:۰۰-۱۰:۳۰	کلاس B۱۰
۶۰۳	حرارت و سیالات	۲	امیرعلی حسینی	۱۰:۳۰-۱۲:۰۰	کارگاه
۲۰۱	شیمی ۱	۲	امیرعلی حسینی	۱۰:۳۰-۱۲:۰۰	آزمایشگاه
*					

Record: 1 of 11

**کنجکاو:**

علت استفاده از عبارت پرسوجو به جای نام جدول در تمرین‌های فوق چیست؟

## 9-5- فیلدهای محاسباتی

می‌توان در پرسوجوها ، نتیجه‌ی عبارتهای محاسباتی را در یک ستون مشاهده کرد. این ستون مشابه یک فیلد عمل می‌کند. این ویژگی یکی از مزایای پرسوجوهاست. به عنوان مثال فرض کنید، می‌خواهیم حاصل ضرب مقادیر دو فیلد Price و Quantity را مشاهده کنیم و در واقع یک فیلد محاسباتی ایجاد کنیم.

### مثال 5-6 :

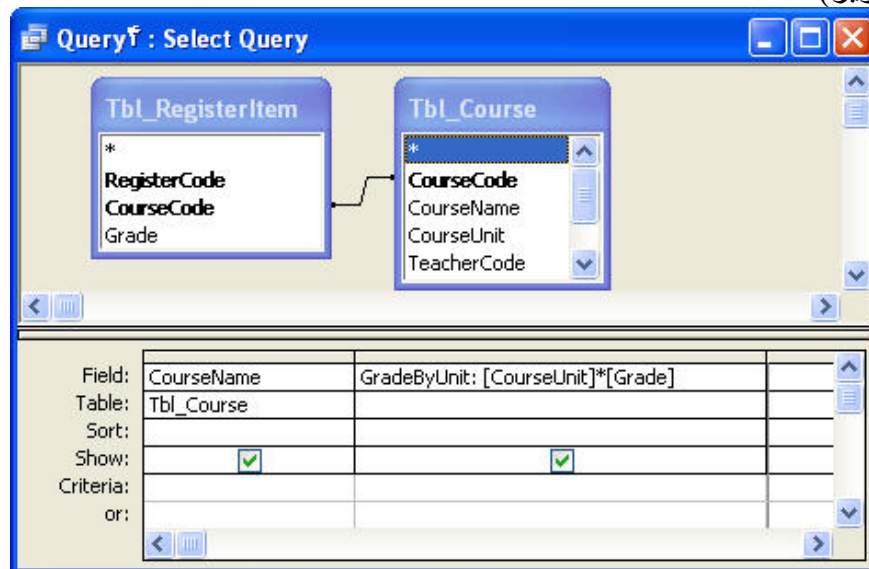
می‌خواهیم حاصل ضرب نمره‌ی درس و واحدهای درسی یک دانش‌آموز را به دست آوریم. Grade در جدول Tbl\_RegisterItem و UnitCourse در جدول Tbl\_Course و این مقدار در فیلد GradeByUnit قرار گیرد.

مراحل انجام کار :

– در نمای طراحی پرسوجو ، جدول‌های Tbl\_RegisterItem و Tbl\_Course را انتخاب می‌کنیم.

– از جدول Tbl\_Course ، فیلد CourseName را به پرسوجو اضافه می‌کنیم.

– در نیمه‌ی پایینی پنجره‌ی طراحی پرسوجو ، ستون خالی بعدی را در نظر می‌گیریم و در سطر Field آن ، عبارت GradeByUnit: Grade\*CourseUnit را تایپ می‌کنیم. (شکل زیر)



### ادامه‌ی مثال 5-6 :


– پرس‌وجو را اجرا می‌کنیم. فیلد موردنظر در انتهای جواب مشاهده می‌شود. (شکل زیر)

CourseName	GradeByUnit
ریاضی ۲	۳۴
ریاضی ۲	۳۰
ریاضی ۲	۲۶
حسابان	۵۷
حسابان	۶۰
حسابان	۲۷
معلومات دینی	۴۲
معلومات دینی	۴۶,۵
معلومات دینی	۵۱
معلومات دینی	۴۲
مکانیک	۱۹

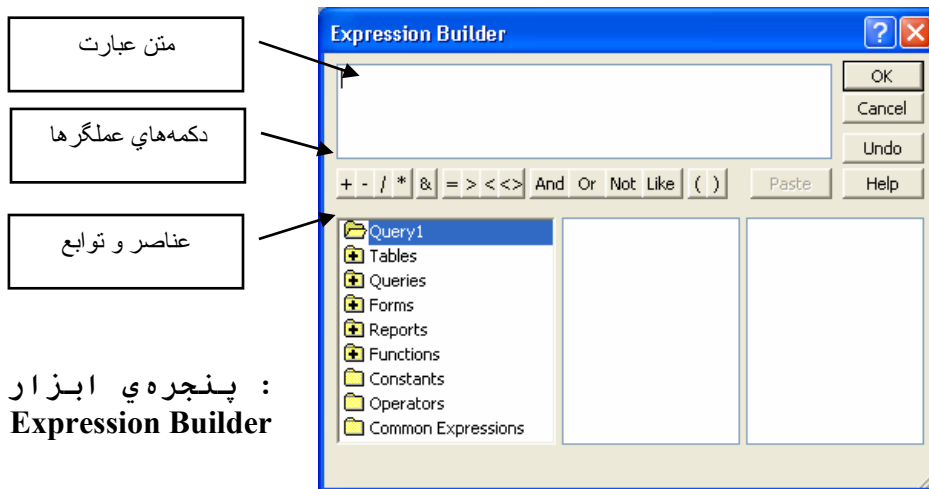
### کنجکاو ی :

عبارت `GradeByUnit` که قبل از `Grade*CourseUnit` قرار گرفته است، چیست؟  
راهنمایی: در عنوان فیلد در نتیجه‌ی خروجی دقت کنید.

### 5-9-1- استفاده از Expression Builder

ابزار مفیدی برای ایجاد عبارت در پرس‌وجو است. با کلیک روی سلول `Field` و انتخاب گزینه‌ی  (Build) از نوار ابزار، پنجره‌ی آن ظاهر می‌شود. (شکل 5-5)

با کمک `Expression Builder` فیلدها و توابع موردنظر را انتخاب می‌کنید و بدین‌ترتیب احتمال خطا و اشتباه تایی می‌کاهش می‌یابد. با استفاده از قسمت `Functions` می‌توانیم توابع تعریف‌شده به وسیله‌ی کاربر ( `User Defined` ) و یا از پیش‌ساخته‌شده در `Access` ( `Build in` ) را استفاده کنیم.



پنجره ی ابزار  
Expression Builder

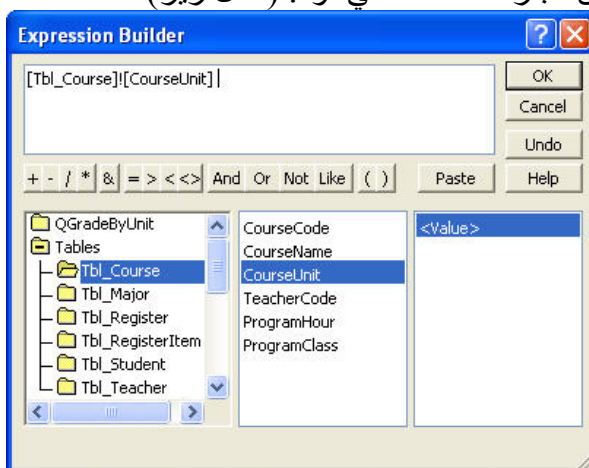
شکل 4-5

### مثال 5-7 :

در مثال حاصل ضرب نمره ی درس در واحدهای درسی یک دانش آموز را با کمک Expression Builder ایجاد کنید.

مراحل انجام کار :

- از قسمت عناصر و توابع ، گزینه ی Tables را انتخاب کرده و Tbl\_Course را انتخاب می کنیم. از لیست وسط فیلد CourseUnit را انتخاب و دوبار کلیک نمایید. این فیلد در قسمت متن عبارت مشاهده می شود. (شکل زیر)



### ادامه‌ی مثال 5-7 :

– سپس از دکمه‌های عملگرها ، \* را کلیک می‌کنیم تا علامت ضربدر به متن عبارت اضافه شود. (شکل زیر)

[Tbl\_Course]![CourseUnit] \*

– سپس از گزینه‌ی Tables ، جدول Tbl\_RegisterItem را انتخاب می‌کنیم . فیلد Grade را انتخاب و دکمه‌ی Paste را از دکمه‌های عملگرها کلیک می‌کنیم ، تا به متن عبارت اضافه شود.

[ [Tbl\_Course]![CourseUnit] \* [Tbl\_RegisterItem]![Grade

– در متن عبارت در ابتدای متن قرار گرفته و GradeByUnit: را تایپ می‌کنیم.  
[GradeByUnit: [Tbl\_Course]![CourseUnit] \* [Tbl\_RegisterItem]![Grade

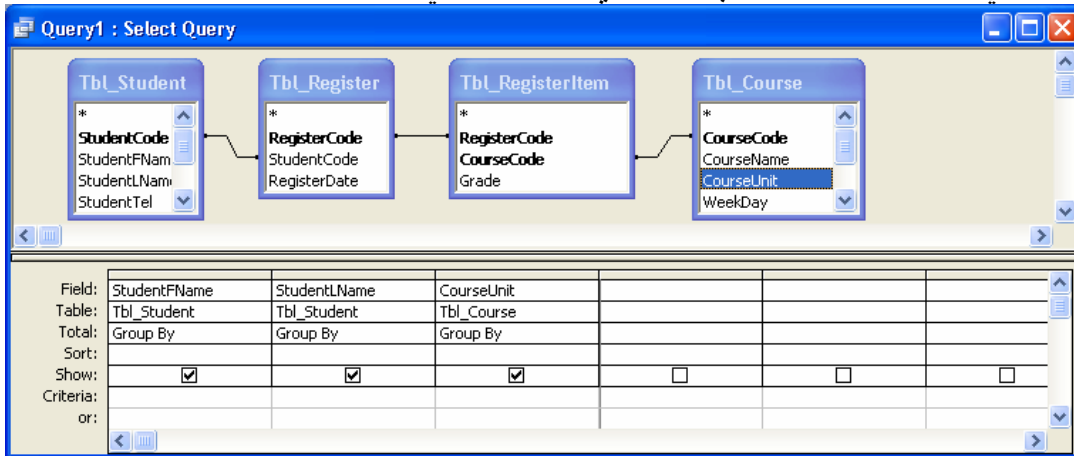
– دکمه‌ی Ok پنجره را کلیک می‌کنیم. با این کار به محیط طراحی پرس‌وجو بر می‌گردیم و عبارت ایجادشده در قسمت Field ستون موردنظر اضافه می‌شود. مانند حالتی که عبارت را تایپ کرده‌ایم.

### 5-10- فیلدهای محاسباتی تجمعی ( توابع Aggregate )

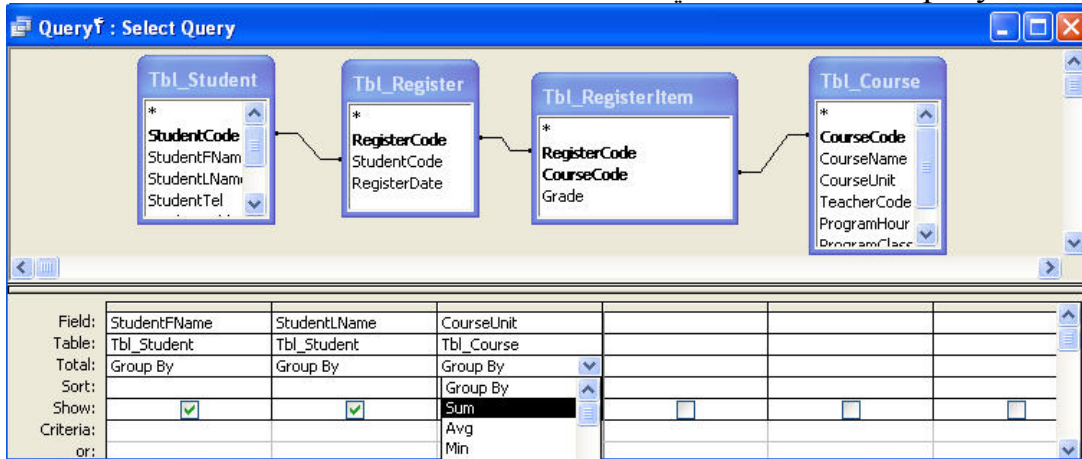
گاهی نیاز به محاسبه‌ی جمع کل ، تعداد کل یک فیلد از یک جدول برحسب مقادیر فیلدی دیگر داریم. برای این محاسبات کاری فراتر از فیلد محاسباتی لازم است. برای این‌کار باید از توابع تجمعی (Aggregate) استفاده کنیم. در پنجره‌ی طراحی پرس‌وجو با انتخاب دکمه‌ی  $\Sigma$  از نوارابزار ، ردیف Total در شبکه‌ی نیمه‌ی پایینی پنجره‌ی طراحی ظاهر می‌شود. این ردیف شامل مقادیر Group by ، Sum ، Count ، Avg ، Min ، Max و ... است. در ستون فیلدی که می‌خواهیم مثلاً جمع کل آن حساب شود ، در ردیف Total ، Sum را انتخاب می‌نماییم. فیلدی که می‌خواهیم جمع کل مقدار برحسب آن لیست شود را به صورت Group by انتخاب می‌کنیم.

### مثال 5-8 :

جمع واحدهاي انتخاب شده ي هر دانش آموز را حساب كنيد.  
 راهنمائي : شكل زير در ايجاد پرس و جوي اوليه كمك مي كند.



فيلد StudentFName و StudentLName و CourseUnit را انتخاب كنيد. دكمه  $\Sigma$  را كليك كنيد. همانطور كه در شكل بالا مي بينيد در رديف Total در ستون Group By براي تمام ستون ها Group By نمايش داده مي شود. در ستون CourseUnit ، Sum را انتخاب كنيد.



با اجراي پرس و جو ، نتيجه را به صورت زير مشاهده مي كنيد.

StudentFName	StudentLName	SumOfCourseU
رضا	ميرزايي	5
سعيد	هدايتي	16
سعيد كلثوم	مهاجري	8

### مثال 5-9 :

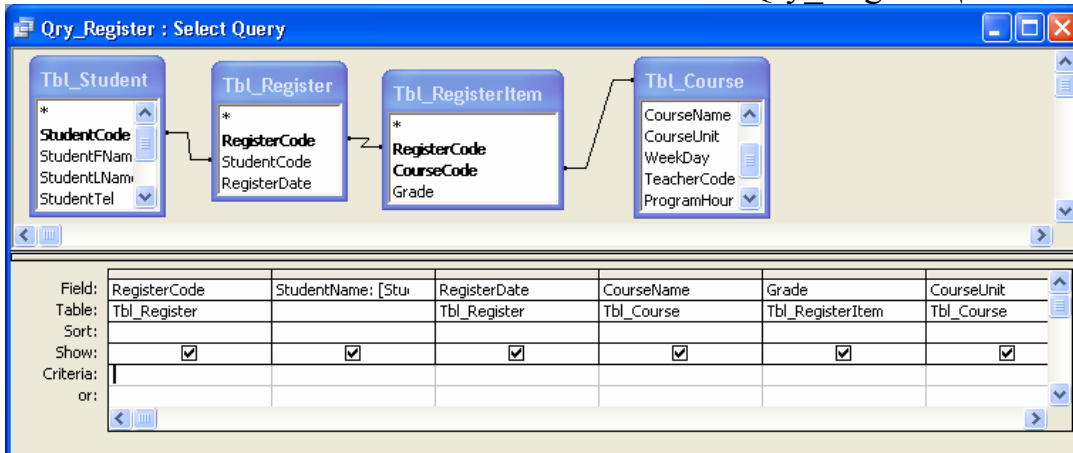
معدل هر دانش آموز را حساب کنید.

راهنمایی :

برای محاسبه‌ی معدل ، باید نمره در تعداد واحدهای درسی ضرب شده و بر مجموع تعداد واحدها تقسیم گردد.

مراحل انجام کار :

- ابتدا پرسوجویی برای Tbl\_Course ایجاد کنید. شکل زیر به شما کمک می‌کند. آن را با نام Qry\_Register ذخیره کنید.



- حال فیلد دیگری به آن اضافه کنید که حاصل ضرب Grade \* CourseUnit را برگرداند. نام فیلد جدید را GradeByUnit قرار دهید و ذخیره کنید.

- با کمک این پرسوجو ، پرسوجویی می‌سازیم که از توابع Aggregate استفاده کرده و مجموع تعداد واحدها و مجموع ضرب حاصل شده را برگرداند. تا بتوانیم معدل را بدین طریق محاسبه کنیم.

- پرسوجویی جدیدی باز کنید و Qry\_Register را به عنوان پرسوجوی موردنیاز اضافه کنید. فیلدهای StudentName ، GradeByUnit و CourseUnit را انتخاب کنید. با زدن دکمه  $\Sigma$  نوار ابزار ، ردیف Total در شبکه‌ی نیمه‌ی پایینی ظاهر می‌شود.

- در ستون GradeByUnit و CourseUnit مقدار را از Group by به Sum تغییر دهید.

- از نوار ابزار ، دکمه‌ی View را انتخاب و گزینه‌ی SQL View را برگزینید. SQL را از عبارت زیر به مقدار جدید اصلاح نمایید :

### ادامه‌ی مثال 5-9 :

```
SELECT Qry_Register.StudentName, Sum(Qry_Register.CourseUnit) AS  
SumOfCourseUnit, Sum(Qry_Register.GradeByUnit) AS SumOfGradeByUnit  
FROM Qry_Register  
GROUP BY Qry_Register.StudentName;
```

### مقدار جدید :

```
SELECT Qry_Register. StudentName, Sum(Qry_Register.GradeByUnit) /  
Sum(Qry_Register.CourseUnit) As GradeAverage  
FROM Qry_Register  
GROUP BY Qry_Register. StudentName ;
```

و آن را با نام Qry\_Average ذخیره نمایید.  
- با اجرای پرس‌وجو ، نتیجه‌ی زیر حاصل می‌شود :

	StudentName	GradeAverage
	رضا میرزایی	۱۹,۵
	سجید هدایتی	۱۸,۵
▶	سید کلثم مهاجری	۰

## 5-11- پرس‌وجوهای عملیاتی Action Query

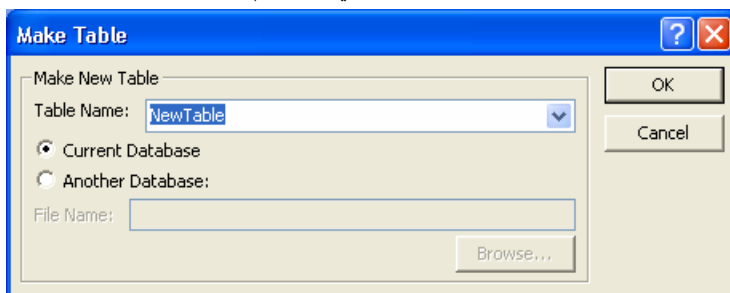
پرس‌وجوهای بازیافت رکوردها (Select) را دیدیم. یکی دیگر از انواع پرس‌وجوها ، پرس‌وجوی عملیاتی است. از این پرس‌وجو بیشتر برای تغییر داده‌ها ( اضافه ، ویرایش ، حذف) استفاده می‌شود. پرس‌وجوی جدول‌ساز ، پرس‌وجوی ویرایش (به‌هنگام‌سازی) داده‌ها ، پرس‌وجوی حذف رکورد.

### 5-11-1- ایجاد پرس‌وجوهای جدول‌ساز

دیدیم که چگونه از پرس‌وجوها برای انتخاب مجموعه‌ای از رکوردها و فیلدهای خاص استفاده می‌شود. رکوردهای نتیجه‌ی پرس‌وجو را می‌توانیم در یک جدول جدید ذخیره نماییم. در این‌صورت فیلدها با همان نوع و مشخصه‌ی جدول اصلی ایجاد می‌شوند و رکوردها نیز به عنوان داده‌های جدول جدید در آن ذخیره می‌شوند.




مشابه روش ایجاد پرسوجوی بازیافت داده (Select) ، پرسوجوی موردنظر را ایجاد میکنیم. پس از ایجاد پرسوجوی جدید (انتخاب جدول/جدولهای موردنیاز و فیلدهای موردنظر) ، از منوی Query گزینهی Make Table Query را انتخاب میکنیم. شکل 5-6 ظاهر میشود.



پنجره ی نام

شکل 5-6 :  
جدول جدید

نام جدول جدید را در قسمت Table Name پنجره تایپ کنید و OK نمایید.

پرسوجو را با گزینهی Save ، ذخیره کنید. علامت  در کنار نام پرسوجو نشان میدهد که این پرسوجو از نوع عملیاتی (Action) است . با اجرای این پرسوجو یک جدول جدید با نام NewTable در لیست جدولهای بانک اضافه میشود.

#### نکته :

زمانی که نمیتوانیم با استفاده از پرسوجو ، اطلاعات موردنیاز را به دست آوریم ، میتوانیم از نتیجهی پرسوجو ، جدول بسازیم و از آن جدول در پرسوجوی بعدی استفاده کنیم تا به جواب مطلوب برسیم.


#### کنجکاو ی :

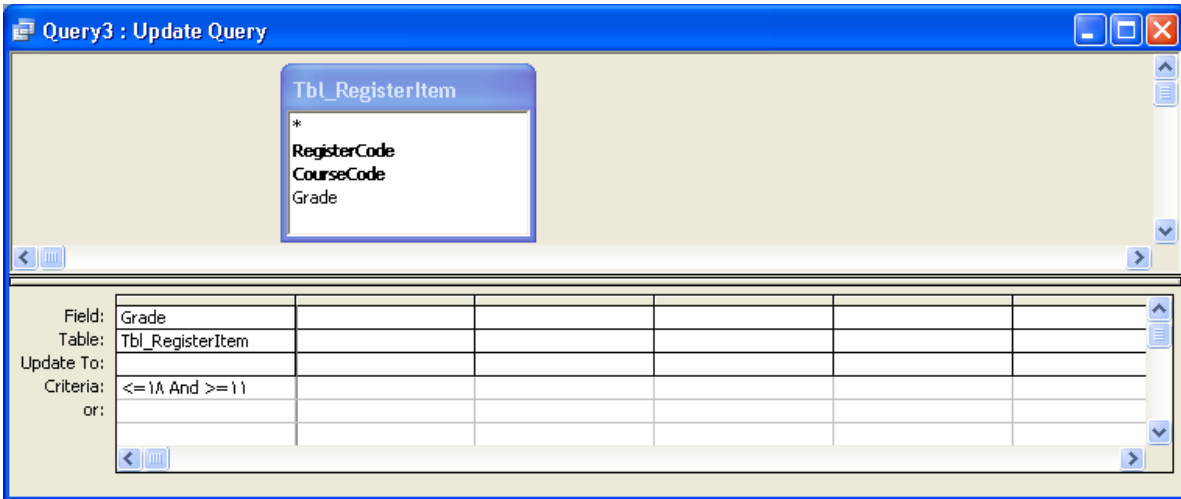
سایر پرسوجوهای عملیاتی که میتوان در Access ایجاد کرد را نام برده و توضیح دهید هر یک برای انجام چه کاری استفاده میشوند؟

### تمرین 5-1 :

پرسوجویی ایجاد کنید که معدل دانش‌آموزان را در جدولی با نام Tbl\_Average ذخیره کند.  
راهنمایی: از پرسوجوی select تمرین معدل هر دانش‌آموز (Qry\_Average) استفاده کنید.

### 2-11-5- ایجاد پرسوجو به‌هنگام‌سازی داده‌ها

یکی از انواع پرسوجوهای عملیاتی است که می‌توانیم فیلد خاصی را در رکوردهای خاص به مقدار جدید یا مقدار محاسبه‌شده‌ی جدیدی تغییر دهیم. پس از ایجاد پرسوجوی Select که رکوردهای مربوطه را برگرداند، از منوی Query، Update Query را انتخاب می‌کنیم. ردیفی با عنوان Update to در شبکه‌ی پنجره‌ی پایینی طراحی پرسوجو (شکل 5-7) ظاهر می‌شود. مقدار جدید را در ستون فیلد موردنظر که می‌خواهیم تغییر کند، تایپ می‌کنیم و پرسوجو را ذخیره می‌نماییم. یک پرسوجوی جدید با علامت  که نشان‌دهنده‌ی عملیاتی بودن آن است خواهیم داشت. با اجرای آن، فیلد موردنظر در جدول اصلی به مقدار جدید به‌هنگام می‌شود. برای ایجاد مقدار محاسباتی، می‌توانیم در ردیف Update to، عبارت محاسباتی را تایپ نماییم. حتی می‌توانیم جدول دیگری را افزوده، ارتباط بین آن دو جدول را ایجاد کنیم و فیلد جدول جدید را به عنوان مقدار جدید برای جدول اصلی در قسمت Update to انتخاب کنیم و یا در عبارت محاسباتی، از فیلد جدول جدید استفاده کنیم. استفاده از جدول دیگر در این نوع پرسوجو مشروط به آن است که با جدول اصلی رابطه داشته باشد، در غیر این صورت به‌هنگام‌سازی معنی‌دار نخواهد بود.



شکل 5-7 : پنجره‌ی به‌هنگام‌سازی

### تمرین 2-5 :

پرس‌وجویی ایجاد کنید که به نمره‌ی درسی دانش‌آموزانی که نمره‌ی آن‌ها از 18 کمتر و از 11 بیشتر است، یک نمره اضافه کند. راهنمایی: پرس‌وجویی ایجاد کنید که از جدول Tbl\_Course آن‌هایی که نمره‌ی آن‌ها از 18 کمتر و از 11 بیشتر باشد را برگرداند، سپس با انتخاب گزینه‌ی Update Query از منوی Query، ستون Grade را انتخاب و مقدار ردیف Update to آن را + 1 Tbl\_RegisterItem.Grade قرار دهید. با اجرای آن، یک نمره به رکوردهای موردنظر اضافه می‌شود.

### 3-11-5- ایجاد پرس‌وجو حذف رکوردها

به کمک این نوع پرس‌وجو می‌توانیم داده‌هایی (رکوردهایی) با شرایط خاص را حذف کنیم. ابتدا پرس‌وجویی می‌سازیم که رکوردهای موردنظر برای حذف را برگرداند. در قسمت شرط، معیار موردنظر را اعمال می‌کنیم، در صورت هیچگونه اعمال شرط، تمام رکوردها حذف خواهند شد. از منوی Query، Delete Query را برمی‌گزینیم. پرس‌وجو را ذخیره می‌کنیم علامت ❗ در کنار آن ظاهر می‌شود و با اجرای آن رکوردهای موردنظر از جدول حذف می‌شوند و اثری از آن‌ها در جدول اصلی نمی‌بینید.

### **دقت :**

رکوردهای تغییر یافته به وسیله ی این پرس و جوها را نمی توان بازگردانید . مثلاً با اجرای حذف ، آن رکوردها برای همیشه حذف شده اند . پس در استفاده از پرس و جوی عملیاتی خصوصاً حذف باید بسیار دقت کنید .  
می توانید با کمک پرس و جوی جدول ساز ، ابتدا از داده های جدول يك جدول جدید حاوی نسخه ی پشتیبانی از داده ها تهیه کنید . تا در صورت هرگونه اشتباه در پرس و جو ، پس از اجرای پرس و جوی حذف ، داده ها را از دست نداده باشید .

### **نکته :**

تمام پرس و جوی های عملیاتی دارای علامت † در کنار نام خود هستند . در هنگام اجرا ، Access با پیغامی از شما می خواهد که اعمال و اجرای عملیات را تایید نمایید . البته ، ظاهر شدن یا نشدن این اخطار و پیغام را می توان با تغییر مشخصات Options در محیط Access تغییر داد .

### **کنجکاو ی :**

چگونه با اجرای پرس و جوی عملیاتی ، پیغام تاییدیه نمایش داده نمی شود؟

### **تمرین 3-5 :**

پرس و جویی ایجاد کنید که در جدول حاوی معدل ، دانش آموزانی که معدل آن ها زیر 10 است را حذف کند .  
راهنمایی : پرس و جویی ایجاد کنید که از جدول Tbl\_Average دانش آموزانی که معدل زیر 10 دارند را برگرداند ، سپس از منوی Query ، گزینه ی Delete را انتخاب کرده و پرس و جو را ذخیره و اجرا نمایید .

## 5-12- ایجاد پرسوجو CrossTab

با این نوع پرسوجو می‌توانیم اطلاعات یک یا چند جدول را در قالب ردیف و ستون خلاصه نماییم. مثلاً محاسبه‌ی **sum** ، **average** ، **count** ، یا دیگر انواع مجموع داده که با دو نوع اطلاع (سطری و ستونی) گروه‌بندی می‌شود. مثلاً یک پرسوجوی ساده ، فقط معدل درسی شاگردان کلاس را برمی‌گرداند. در صورتی که اطلاعات زیاد باشد ، مقایسه سخت می‌شود.

یک پرسوجوی **Crosstab** همان اطلاع را نشان می‌دهد ، با این تفاوت که اطلاعات سطری با شاگردان و ستونی با درس گروه‌بندی می‌شود و این‌کار تحلیل را آسان‌تر می‌کند. بنابراین یک نوع پرسوجوی انتخاب رکوردها (**Select**) است ، با این تفاوت که اطلاعات را خلاصه‌تر ارائه می‌کند. در این پرسوجو ، داده‌های یک فیلد خاص را به عنوان ستون‌های جدول خروجی و فیلد دیگری را به عنوان ردیف‌های جدول خروجی تعیین می‌کنیم و مثلاً جمع کل مقادیر سطر بر حسب ستون را به عنوان مقدار محل برخورد سطروستون (سلول) تعیین می‌کنیم.

برای ایجاد این نوع پرسوجو ، از **New Query** ، گزینه‌ی **CrossTab Query wizard** را انتخاب کنید. سپس جدول را انتخاب نموده و یک فیلد را به عنوان "عنوان سطر" **Row heading** و فیلد دیگری را به عنوان "عنوان ستون" **Column heading** بر می‌گزینیم. یکی از توابع را به عنوان داده‌های محل تقاطع سطرها و ستونها انتخاب می‌کنیم.

### تمرین 4-5 :

معدل هر يك از درس هاي هر دانش آموز را حساب كنيد.  
راهنمائي : از شكل هاي زير كمك بگيريد.

Field:	StudentName: [Stu	CourseName	CourseAverageGrade: Sum((CourseUnit)*Grade))/Sum(Tbl_Course.CourseUnit)
Table:	Tbl_Student	Tbl_Course	Tbl_Course
Total:		Group By	Expression
Sort:			
Show:	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Criteria:			
or:			

### ادامه ي تمرین 4-5 :

Field:	StudentName: [S	CourseName	CourseAverageGrade: Sum((CourseUnit)*Grade))/Sum(Tbl_Course.CourseUnit)
Table:	Tbl_Student	Tbl_Course	Tbl_Course
Total:	Group By	Group By	Expression
Crosstab:	Row Heading	Column Heading	Value
Sort:			
Criteria:			
or:			

### 5-13- زبان پرسوجوي ساختيافته SQL

همان طور که قبلا دیدیم در قسمت طراحی پرسوجو ، در نوار ابزار گزینه ي SQL View وجود دارد. با انتخاب این گزینه ، عبارت SQL مربوط به پرسوجو دیده می شود. می توانید پرسوجوي Qry\_Student را انتخاب کرده ، به Design بروید و نمای Sql view را انتخاب کنید. شکل زیر (5-8) را خواهید دید.

```

SELECT Tbl_Student.StudentCode, Tbl_Student.StudentFName, Tbl_Student.StudentLName, Tbl_Student.StudentTel,
Tbl_Student.StudentAddress, Tbl_Degree.DegreeName, Tbl_Major.MajorName
FROM Tbl_Major INNER JOIN (Tbl_Degree INNER JOIN Tbl_Student ON Tbl_Degree.DegreeCode = Tbl_Student.DegreeCode) ON
Tbl_Major.MajorCode = Tbl_Student.MajorCode;
    
```

## شکل 8-5 : نمای SQL در پنجره ی طراحی پرسوجو

در این پنجره ، عباراتی نوشته شده است که يك دستور Select است که با قواعد زبان استاندارد ساختیافته ی بانک (SQL) نوشته شده است. اگر این عبارت را با عبارت سایر پرسوجوهای موجود مقایسه کنید ، می توانید شباهتی بین آنها دریابید یعنی قالب دستورات زبان ثابت است . با رعایت اصول و قواعد زبانی ، می توان آن را به کار گرفت.

SQL ، زبانی است که تمام دستورات پرسوجو مثل Select ، Update ، Delete و ... به وسیله ی آن تعریف می شوند. ممکن است کلماتی چون Where ، Order by و ... نیز داشته باشد. عموماً این زبان برای ایجاد پرسوجوهای بانک استفاده می شود ولی می توان از آن برای ایجاد یا تغییر ساختار پایگاه داده نیز استفاده کرد که به این نوع SQL زبان تعریف داده (DDL)<sup>1</sup> گویند که در این کتاب به آنها نمی پردازیم. در ادامه روش ایجاد پرسوجوهای معمول را با قواعد این زبان بررسی می کنیم که به آنها زبان دستکاری داده (DML)<sup>2</sup> گفته می شود.

با کمک محیط گرافیکی پرسوجو به سادگی برخی پرسوجوها را ایجاد کردیم. ولی پرسوجوهای پیچیده تر فقط با نوشتن عبارات زبان ساختیافته و با قالب مجاز ایجاد می شوند و نمی توان آنها را به طریق گرافیکی که در موارد بالا دیدیم ، ایجاد کنیم. مثلاً:

Sub-query ، Union ، Data-Definition

### قالب زبان عبارات پرسوجو:

- ابتدا علایم استفاده شده در قواعد را شرح می دهیم :
1. {} باید یکی از موارد داخل آن استفاده شود. موارد انتخابی با "|" و یا [] از هم جدا شده اند.
  2. [] اختیاری بودن .

<sup>1</sup> - Data Definition Language

<sup>2</sup> - Data Manipulating Language

3. **پررنگ بودن قلم** نشانه‌ی کلمه‌ی کلیدی بودن آن  
واژه است. مثل **SELECT** (بزرگ یا کوچک بودن  
حروف مهم نیست ولی ما برای خوانایی از حروف  
بزرگ استفاده کرده‌ایم)

1-9-5- **دستور SELECT** : برای انتخاب رکوردها از این  
دستور و با قالب زیر استفاده می‌شود. این  
دستور می‌تواند بسیار ابتدایی و ساده باشد و یا  
بسیار پیچیده گردد. این دستور زیربنایی‌ترین  
دستورات است و سایر دستورات را می‌توان با  
تغییر این دستور ایجاد کرد. قالب قواعد زبانی  
آن عبارت است از :

```
SELECT [ALL / DISTINCT / DISTINCTROW / [TOP n [PERCENT]]]  
{ * | table.* | [table.]field1 [AS alias1] [, [table.]field2 [AS alias2] [, ...] }  
FROM table [, ...] [IN externaldatabase]  
[WHERE... ]  
[GROUP BY... ]  
[HAVING... ]  
[ORDER BY... ]
```



### مثال 5-10 :

پرسوجوهاي زير را ايجاد كنيد.

1. مثال Select ، ميخواهيم تمام ركوردهاي معلمان مدرسه را مشاهده كنيم :

```
SELECT ALL *           يا           SELECT *
FROM Tbl_Tecaher      FROM Tbl_Tecaher
```

2. مثال Distinct ، ميخواهيم درسهاي ثبتنام شدهي ترم را بهدست آوريم :

```
SELECT DISTINCT CourseCode, CourseName
FROM Tbl_Course , Tbl_RegisterItem
WHERE Tbl_Course.CourseCode =
Tbl_RegisterItem.CourseCode
```

\* اگر از DISTINCT استفاده نكنيم ، چون هر درس بهوسيلهي چندين دانش آموز ثبتنام شده است ، ركوردهاي مشابه بر ميگرداند.

3. مثال As alias ، ميخواهيم نام و نامخانوادگي معلمان را به صورت يك فيلد اطلاعاتي با عنوان TeacherName داشته باشيم :

```
SELECT TeacherCode, TeacherLName + ' ' + TeacherFName
FROM Tbl_Teacher
```

اين پرسوجو را اجرا كنيد . نتيجه به صورت زير خواهد بود :



TeacherCode	Expr1001
۱۰۴۵	سعدی نژاد حامد رضا
۱۰۴۶	مهدی فر محمد
۱۰۴۷	میرزا پور علیرضا
۱۰۴۸	هدایتی فر کامبیز
۱۰۴۹	حسینی امیرعلی
۱۰۵۰	صبور طینت محمد
۱۰۶۰	هدایت نژاد محمد
۱۰۶۱	سالمی پور علیرضا
*	,

حال نام مستعاري به فيلد منتسب ميكنيم ، عبارت AS TeacherName :

### ادامه‌ی مثال 10-5 :

```
SELECT TeacherCode, TeacherLName + ' ' + TeacherFName
AS TeacherName
FROM Tbl_Teacher
```

نتیجه‌ی جدید به صورت زیر خواهد بود :

TeacherCode	TeacherName
۱۰۴۵	سعدي نژاد حامد رضا
۱۰۴۶	مهدي فر محمد
۱۰۴۷	ميرزا پور علي رضا
۱۰۴۸	هدايتي فر کاميبيز
۱۰۴۹	حسيني امير علي
۱۰۵۰	صبور طيننت محمد
۱۰۶۰	هدايت نژاد محمد
۱۰۶۱	سلمانني پور علي رضا
*	

### کنجکاو ي :

1. Top n [percent] چگونه عمل می‌کند؟ کجا می‌توان از این قابلیت استفاده کرد؟ راهنمایی : وقتی داده‌های جدول زیاد باشد. حجم رکوردهای نتیجه بالا است و سرعت بازیابی کند باشد و مشاهده‌ی نتیجه‌ی پرس‌وجو طول بکشد.
2. در قسمت FROM پرس‌وجو می‌توان از گزینه‌ی IN استفاده کرد. این گزینه چه عملکردی دارد؟ راهنمایی : جدول موردنظر در بانک داده‌ی دیگری قرار داشته باشد.

### 2-9-5- دستور SELECT... INTO : ایجاد جدول جدید شامل

رکوردهای نتیجه‌ی پرس‌وجو (پرس‌وجوی جدول ساز).  
قسمت SELECT آن مشابه قبل است. تنها در خط اول پس از انتخاب فیلدهای موردنظر، کلیدواژه‌ی INTO را به همراه نام جدول جدید قرار می‌دهیم. در صورتی که بخواهیم جدول جدید

در پایگاه داده‌ی دیگری قرار گیرد ، آدرس آن را در IN قید می‌کنیم .

```
SELECT field1[,field2[, ...]] INTO newtable [IN externaldatabase]  
FROM source-TableName
```

#### مثال 5-11 :

نسخه‌ی پشتیبانی از جدول معلمان ایجاد کنید :  
پرسوجوی زیر را ایجاد ، ذخیره و اجرا نمایید.

```
SELECT * INTO Tbl_TeacherBackUp  
FROM Tbl_Teacher
```

3-9-5- DELETE : حذف رکوردها .

```
DELETE [table.*]  
FROM table  
WHERE criteria
```

#### مثال 5-12 :

رکوردهای جدول معلمان را حذف کنید .  
پرسوجوی زیر را ایجاد ، ذخیره و اجرا نمایید.

```
DELETE  
FROM Tbl_Teacher
```

4-9-5- INSERT : اضافه کردن رکوردها .  
دو نوع دستور اضافه کردن رکورد وجود دارد .  
اضافه کردن به وسیله ی مقادیر ثابت داده ( Single  
append ) و اضافه کردن رکوردهای از پرسوجوی نوع  
Select (Multiple append) .

- **Multiple-record append query:**

```
INSERT INTO target-TableName [(field1[,field2[, ...]])] [IN externaldatabase]  
SELECT [source-TableName.]field1[,field2[, ...]]  
FROM Source_TableName
```

### مثال 5-13 :

اطلاعات جدول معلمان را از نسخه‌ی پشتیبان برگردانید.  
پرس‌وجوی زیر را ایجاد، ذخیره و اجرا نمایید.

```
INSERT INTO Tbl_Teacher (TeacherCode, TeacherFName,  
TeacherLName, TeacherTel, TeacherAddress)  
SELECT TeacherCode, TeacherFName, TeacherLName,  
TeacherTel, TeacherAddress  
FROM Tbl_TeacherBackUp
```

- **Single-record append query:**

```
INSERT INTO target-TableName [(field1[, field2[, ...]])]  
VALUES (value1[, value2[, ...]])
```

### مثال 5-14 :

اطلاعات معلم جدید را اضافه کنید :  
پرس‌وجوی زیر را ایجاد، ذخیره و اجرا نمایید.

```
INSERT INTO Tbl_Teacher (TeacherCode, TeacherFName,  
TeacherLName, TeacherTel, TeacherAddress)  
VALUES (1070 , ' , 'خیابان وزرا ' , '88971034' , 'یعقوبی' , 'مدفج' ,  
'کوی پنجم ، پلاک 2/12')
```

### 5-9-5 - دستور UPDATE : ویرایش رکوردها .

```
UPDATE table  
SET newvalue  
WHERE criteria;
```

### مثال 5-15 :

نام معلم جدید به اشتباه «صدفج» ثبت شده است. آن را ویرایش نمایید :  
پرس‌وجوی زیر را ایجاد، ذخیره و اجرا نمایید.

```
Update Tbl_Teacher SET TeacherFName='مدف'  
WHERE TecaheCode=1070
```

5-9-6- عملگر UNION : برای ایجاد پرسوجوی ترکیب رکوردهاست.

```
[TABLE] query1 UNION [ALL] [TABLE] query2  
[UNION [ALL] [TABLE] queryn [ ... ]]
```

در این دستور باید تعداد فیلدهای دو Select که با هم Union میشوند ، یکسان باشد. نوع ، ترتیب و نام آنها نیز یکی باشد ، در صورت عدم تساوی نام ، باید از Alias (افزودن عنوان جدید با AS) استفاده کرد.

#### مثال 5-16 :

لیست اسامی معلمان و دانش‌آموزان مدرسه را به دست آورید.  
پرسوجوی زیر را ایجاد ، ذخیره و اجرا نمایید.

```
SELECT StudentCode AS PersonCode, StudentFName AS  
PersonFName, StudentLName AS PersonLName  
FROM Tbl_Student  
UNION  
SELECT TeacherCode AS PersonCode, TeacherFName AS  
PersonFName, TeacherLName AS PersonLName  
FROM Tbl_Teacher
```

#### 5-9-7- دستور JOIN :

دیدیم یکی از روشهای انتخاب اطلاعات از دو یا چند جدول ، آن است که در FROM نام جدولها را با علامت "،" از هم جدا کرده و در WHERE ارتباط آنها را به صورت شرایط دوجه دو بیان کنیم. یکی دیگر از راهها که زمان ایجاد پرسوجو با کمک گرافیک نیز دیده ایم ، اتصال جدولها با دستور JOIN است. شامل دو روش INNER JOIN و OUTER JOIN است. INNER JOIN : اتصال دو جدول ، زمانی که اطلاعات متناظر در هر دو جدول وجود داشته باشند. اطلاعاتی که در یکی باشد و در دیگری وجود نداشته باشد ، در نتیجه پرسوجو نمایش داده نخواهند شد.

```
FROM table1 INNER JOIN table2 ON table1.field1 compopr table2.field2
```

**مثال 5-17 :**

اطلاعات معلمان هر درس را به دست آورید.  
پرسوجوي زیر را ایجاد، ذخیره و اجرا نمایید.

```
SELECT CourseName, TeacherFName, TeacherLName
FROM Tbl_Teacher INNER JOIN Tbl_Course
ON Tbl_Teacher.TeacherCode = Tbl_Course.TeacherCode
```

- **Outer Join (Right/Left):** اطلاعات متناظر دو جدول در نتیجه‌ی پرسوجو مشاهده می‌شود، حتی اگر در دیگری وجود نداشته باشد. **LEFT JOIN** تمام اطلاعات را از جدول اول نشان می‌دهد حتی اگر در جدول دوم رکوردی متناظر آن وجود نداشته باشد. در این صورت مقدار فیلدهای انتخاب شده از جدول دوم را **NULL** می‌آورد. به همین ترتیب، **RIGHT JOIN** تمام اطلاعات جدول دوم را می‌آورد. عبارت تساوی بعد از **ON** نیز، ارتباط دو جدول را نشان می‌دهد.

`FROM table1 { LEFT | RIGHT } JOIN table2 ON table1.field1 compopr table2.field2`

**مثال 5-18 :**

اطلاعات معلمان هر درس را به دست آورید. در ضمن معلمانی که در این ترم درسی ارائه نداده‌اند نیز آورده شود.  
پرسوجوي زیر را ایجاد، ذخیره و اجرا نمایید.

```
SELECT TeacherFName, TeacherLName, CourseName
FROM Tbl_Teacher LEFT JOIN Tbl_Course
ON Tbl_Teacher.TeacherCode = Tbl_Course.TeacherCode
```

### مثال 5-19 :

دروس ثبت‌نام شده‌ی دانش‌آموزان را به‌دست آورید. در ضمن دانش‌آموزانی که در این ترم هنوز درسی ثبت‌نام نکرده‌اند نیز آورده شود.

```
SELECT StudentFName, StudentLName, CourseName
FROM Tbl_Student LEFT JOIN (Tbl_Register LEFT JOIN
(Tbl_Course RIGHT JOIN Tbl_RegisterItem ON
Tbl_Course.CourseCode = Tbl_RegisterItem.CourseCode)
ON Tbl_Register.RegisterCode =
Tbl_RegisterItem.RegisterCode) ON
Tbl_Student.StudentCode = Tbl_Register.StudentCode
```

### کنجکاو ی :

NULL چه مقداری است. اعمال شرط و یا محاسبات روی آن چگونه است؟

### 8-9-5- اعمال سایر شرایط به پرسوجوها :

- عبارت WHERE شرط و معیار محدود ساختن رکوردهاست. یک عبارت Where ، می‌تواند شامل 40 عبارت باشد که با عملگرهای منطقی And یا Or به هم وصل شده‌اند. عملگرهای ریاضی و منطقی عبارت شرط را می‌سازند. معیارهای عبارت Where نیز مشابه فیلتر که در قسمت 4-7 توضیح دادیم است و از همان قواعد ذکر شده پیروی می‌کند.

### مثال 5-20 :

معلمی با نام 'مهدی فر' را حذف کنید.  
پرسوجوی زیر را ایجاد، ذخیره و اجرا نمایید.

```
DELETE
FROM Tbl_Teacher
WHERE TeacherCode=1046
```

می‌توانستیم از عبارت زیر استفاده کنیم ولی دقت نمایید که در صورتی که چند رکورد با نام خانوادگی مذکور وجود داشته باشد، با اجرای این پرسوجو تمامی آن‌ها حذف می‌شوند.

```
DELETE
FROM Tbl_Teacher
WHERE TeacherLName='مهدی فر'
```

- عبارت ORDER BY : مرتب سازی رکوردها.

```
SELECT fieldlist
FROM table
WHERE selectcriteria
[ORDER BY field1 [ASC | DESC ][, field2 [ASC | DESC ]][, ...]]
```

با این عبارت ترتیب نمایش رکوردها را تغییر می‌دهیم. بدون ذکر Order by رکوردها بدون هیچ ترتیب خاصی و بر اساس همان ترتیب ورود اطلاعات نمایش داده می‌شوند. مرتب سازی ابتدا بر اساس فیلد اول و در صورت تساوی مقادیر این فیلد ، روی فیلد دوم ذکر شده در عبارت خواهد بود. دو واژه‌ی ASC و DESC ترتیب صعودی و نزولی را مشخص می‌کنند :

- ASC : مرتب‌سازی به صورت صعودی است. از مقدار کمتر به بیشتر.
- DESC : مرتب‌سازی به صورت نزولی است. از مقدار بیشتر به کمتر.

#### مثال 5-21 :

لیست اسامی معلمان بر اساس نام خانوادگی و در صورت تساوی نام خانوادگی بر اساس نام مرتب شده باشد به دست آورید.  
پرسوجوی زیر را ایجاد ، ذخیره و اجرا نمایید.

```
SELECT Tbl_Teacher.TeacherFName,
Tbl_Teacher.TeacherLName
FROM Tbl_Teacher
ORDER BY Tbl_Teacher.TeacherLName ASC,
Tbl_Teacher.TeacherFName ASC
```

- عبارت GROUP BY : پرسوجوی تجمعی (Aggregate) شامل : جمع کل ، تعداد کل و ....

```
SELECT fieldlist
FROM table
WHERE criteria
[GROUP BY groupfieldlist]
```



## تمرین 5-5 :

SQL پرسوجوی محاسبه‌ی معدل Qry\_Average را مشاهده کنید.

```
SELECT Qry_Register.StudentName,  
Sum(Qry_Register.GradeByUnit) /  
Sum(Qry_Register.CourseUnit) AS GradeAverage  
FROM Qry_Register  
GROUP BY Qry_Register.StudentName
```

این پرسوجو یعنی نام دانش‌آموز را در نظر می‌گیرد و مجموع حاصل‌ضرب نمره در واحد درسی را به‌دست آورده و برای محاسبه‌ی معدل تقسیم انجام می‌دهد.

نکته : تمام فیلدهای عبارت Select غیر از آن‌هایی که در توابع تجمعی Min ، Sum ، Count استفاده شده‌اند ، باید در لیست قسمت Group By آورده شده باشند.

- عبارت HAVING : شرط و معیار در پرسوجوی تجمعی است.

زمانی که از توابع تجمعی Aggregate و Group by استفاده شود ، برای اعمال شرط پس از اجرای پرسوجو ، باید از معیار در HAVING استفاده کنیم. اگر پرسوجو دارای عبارت شرط Where باشد ، رکوردها ابتدا به وسیله‌ی آن شرط محدود می‌شوند ، سپس نتیجه‌ی تابع تجمعی و فیلدها محاسبه می‌شوند و پس از آن شرط HAVING روی نتیجه اعمال می‌گردد. در این صورت می‌توان از فیلدهای Group by و فیلدهای توابع تجمعی نیز در شرط استفاده کنیم. در صورتی‌که در عبارت Where فقط می‌توانستیم فیلدهای Group by را استفاده و بر آن اساس پرسوجو را محدود کنیم.

```
SELECT fieldlist  
FROM table  
WHERE selectcriteria  
GROUP BY groupfieldlist  
[HAVING groupcriteria]
```

### مثال 5-22 :

می‌خواهیم معدل دانش‌آموزانی را به دست آوریم که بیش از 10 واحد درسی انتخاب کرده‌اند. پرس‌وجوی زیر را ایجاد، ذخیره و اجرا نمایید.

```
SELECT Qry_Register.StudentName,  
SUM(Qry_Register.GradeByUnit)/SUM(Qry_Register.CourseUnit)  
AS GradeAverage  
FROM Qry_Register  
GROUP BY Qry_Register.StudentName  
HAVING SUM(Qry_Register.CourseUnit) > 10
```

نکته : دقت کنید که نمی‌توانیم با کمک عبارت شرط Where این پرس‌وجو را ایجاد کنیم.

دیدیم که با محیط گرافیکی ایجاد شده به وسیله Access به آسانی و بدون نوشتن دستورات زبانی SQL، می‌توانیم پرس‌وجوی موردنظرمان را ایجاد کنیم. اما همان‌طور که گفتیم برخی پرس‌وجوها از قبیل: Sub-query، Union، Data-Definition را باید با نوشتن دستورات زبان در قسمت SQL View پرس‌وجو ایجاد کنیم.

## برای مطالعه‌ی بیشتر :

• **UNION** : برای ایجاد UNION ابتدا دو پرس‌وجوی موردنیاز را ایجاد می‌کنیم که تعداد ، نوع ، ترتیب و نام مشابهی از فیلدها را برگردانند. سپس در یک پرس‌وجوی جدید SQL اولی را کپی می‌کنیم ، کلیدواژه‌ی UNION را تایپ می‌کنیم و پس از آن SQL دومی را کپی می‌کنیم. می‌توانیم تعداد Select‌های بیشتری را نیز اضافه کنیم ، یعنی تایپ مجدد UNION در ادامه و کپی SQL دیگر و به همین ترتیب.

• **SUBQUERY** : این نوع پرس‌وجو ، Select‌ای است که داخل پرس‌وجوی دیگر استفاده شود. سه روش استفاده از آن در پرس‌وجو در زیر آورده شده است :

1. *comparison* [ANY | ALL | SOME] (*sqlstatement*)

2. *expression* [NOT] IN (*sqlstatement*)

3. [NOT] EXISTS (*sqlstatement*)

می‌توانیم از SubQuery به‌جای استفاده از عبارت محاسباتی در فیلد Select یا شرط Where و شرط Having استفاده کنیم. به‌طور مثال :

1. 

```
SELECT * FROM Tbl_Student
WHERE StudentCode > ANY
(SELECT StudentCode
FROM Tbl_Register INNER JOIN Tbl_RegisterItem ON
Tbl_Register.RegisterCode = Tbl_RegisterItem.RegisterCode
WHERE Tbl_RegisterItem.Grade >=10);
```
2. 

```
SELECT * FROM Tbl_Student
WHERE StudentCode IN
(SELECT StudentCode
FROM Tbl_Register INNER JOIN Tbl_RegisterItem ON
Tbl_Register.RegisterCode = Tbl_RegisterItem.RegisterCode
WHERE Tbl_RegisterItem.Grade >=10);
```
3. 

```
SELECT * FROM Tbl_Student
WHERE EXISTS
(SELECT StudentCode
FROM Tbl_Register INNER JOIN Tbl_RegisterItem ON
Tbl_Register.RegisterCode = Tbl_RegisterItem.RegisterCode
WHERE Tbl_RegisterItem.Grade >=10);
```

• **Data Definition** : همان‌طور که گفتیم دستورات SQL‌ای وجود دارند که برای ایجاد یا تغییر ساختار پایگاه داده استفاده می‌شوند. در اینجا یک مثال از ایجاد جدول با دستور SQL را می‌بینیم. تمام تغییراتی که در طراحی بانک با کمک محیط گرافیکی Access انجام می‌دادیم ، به وسیله‌ی دستورات زبان نیز قابل انجام است. مثل تغییر نام فیلد ، تغییر نوع داده‌ی یک فیلد ، افزودن فیلد جدید ، افزودن کلیداصلی به جدول و .... به‌طور مثال: با اجرای دستور زیر که در SQL View یک پرس‌وجوی جدید تایپ می‌نمایید ، جدولی با نام Tbl\_Friends و با شش فیلد و کلیداصلی روی فیلد FriendID ایجاد می‌شود.

```
CREATE TABLE Tbl_Friends
([FriendID] integer,
[LastName] text(20),
[FirstName] text(30),
[Birthdate] date,
[Phone] text(25),
[Notes] memo,
CONSTRAINT [Index1] PRIMARY KEY ([FriendID]))
```

## خلاصه ي فصل

براي تصميم گيري نياز به بازيافت اطلاعات ثبت شده در جدول هاي بانك داريم و بايد اين اطلاعات را تجزيه و تحليل كنيم. از پرسوجوها براي دريافت اطلاعات از بانك استفاده مي شود. پرسوجوها انواع مختلفي دارد كه معمول ترين آن پرسوجوي **Select** است. به هر پرسوجو مي توان محدوديت و شرط اضافه كرد. مي توان فيلدهاي غير جدولي از نوع محاسباتي به پرسوجو اضافه كنيم. با اجراي پرسوجو نتيجه ي آن به صورت جدولي نشان داده مي شود كه شبیه نمای **Datasheet** جدول است و همان عملیات را مي توان روي آن اعمال كرد مثل: مرتب سازي، فيلتر كردن و... البته نتيجه ي پرسوجو در فرم يا گزارشها نيز استفاده مي شود. نوع ديگر پرسوجو، پرسوجوهاي عملياتي هستند كه از آنها براي تغيير روي داده ها استفاده مي شود و به آن زبان دستكاري داده **DML** مي گوييم. البته پرسوجوي عملياتي براي تغيير ساختار بانك نيز وجود دارد كه به آن زبان تعريف داده **DDL** گوييم. از انواع **DML** ها، مي توانيم به پرسوجوي جدول ساز، به هنگام سازي، حذف اشاره كنيم كه به وسيله ي محيط گرافيكي **Access** نيز مي توانند ايجاد شوند. در پشت ظاهر گرافيكي هر يك از اين پرسوجوها، زبان ساخت يافته **SQL** قرار دارد. يعني معادل هر يك از اين امكانات يك دستور زباني قرار دارد كه با آنها آشنا مي شويم. پرسوجوهاي پيچيده تر و پيشرفته تر به سختي به وسيله ي محيط گرافيكي قابل ايجاد هستند و بايد به وسيله ي برنامه نويس و با قواعد زبان **SQL** نوشته شوند.

## خود آزمايي

1. پرسوجوها چه كاري انجام مي دهند و چه زماني ثابلي استفاده هستند؟
2. پنجره ي طراحي پرسوجو را به اختصار توضيح دهيد.

3. معيارهاي پرسوجو به چه منظور استفاده مي‌شوند؟
4. فيلدهاي محاسباتي را توضيح دهيد؟ با چه ابزار گرافيكي قابل ايجاد هستند؟
5. پرسوجوهاي عملياتي را شرح دهيد؟
6. زبان پرسوجوي ساختيافته SQL چيست؟
7. برخي دستورات SQL را نام ببريد؟

## 6- فصل ششم : ایجاد و کار با گزارش ها

در این فصل ایجاد گزارش و عملیات روی آن را بررسی می‌کنیم . گزارش‌ها یکی از راه‌های مفید نمایش و آرایه داده‌ها در یک قالب چاپی است. چون اندازه و ظاهر هر شیء روی گزارش قابل کنترل است ، بنابراین می‌توانید اطلاعات را همان‌گونه که می‌خواهید عرضه کنید.

### در پایان این فصل انتظار می‌رود که هنرجو بتواند :

- تفاوت گزارش با سایر اشیاء را تشخیص دهد.
- نماهای مشاهده‌ی گزارش Preview ، ایجاد گزارش و تغییر طراحی را بشناسد.
- گزارش‌ها را به وسیله‌ی AutoReport یا report Wizard ایجاد کند.
- می‌توانید گزارش‌های ایجاد کند که :
  - داده‌ها را در گروه‌هایی دسته‌بندی کرده و نمایش دهد.
  - جمع کل ، جمع برحسب گروه‌ها ، درصد و غیره را محاسبه کرده ، نمایش دهد.
  - شامل زیرگزارش ، زیر فرم ، گراف و نمودار باشد.
  - برچسب‌های پستی ایجاد کند.
  - داده‌های جدول‌ها و پرس‌وجوهای مختلف را ترکیب کرده و نمایش دهد.
  - داده‌ها را با کمک نوع قلم ، خطوط و تصاویر در قالبی جذاب آرایه دهد.

یک گزارش ، رکوردها را شبیه نمای Datasheet فرم‌ها و جدول‌ها نشان می‌دهد. تفاوت نمای View در گزارش با Datasheet آن است که در گزارش خلاصه‌ی آماری ، عناوین ، زیرنویس ، شماره‌ی صفحه ، قالب نمایش داده زیباتری فراهم می‌شود. همچنین می‌توان داده‌های قابل‌نمایش را انتخاب کرد ، داده‌ها را گروه‌بندی کرد و جمع کل روی آن گروه و جمع نهایی را نمایش داد.

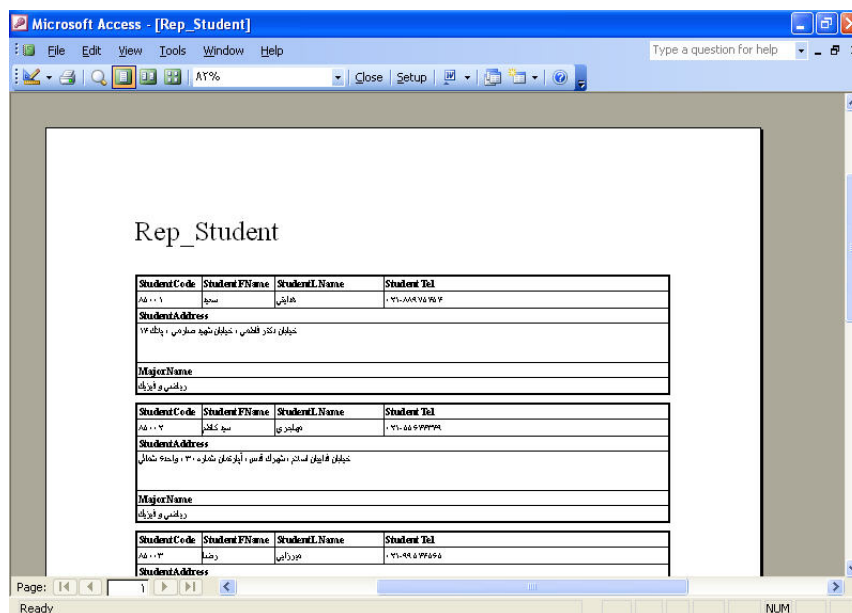
گزارش‌ها در سه نما قابل نمایش است .

- نمای Design ، می‌توانید ظاهر گزارش را ایجاد و یا تغییر دهید.
- نمای Print ، با مرور داده‌های واقعی در هر صفحه‌ی گزارش می‌توانید گزارش را تایید کنید.

- نمای Layout Preview ، با داده های نمونه فقط قالب نمایش گزارش را سریع مشاهده کنید.

**نکته :**  
گزارش فقط لیستی از مقادیر داده نیست. هر زمان که نیاز دارید داده هایی را با قالب خاص چاپ کنید باید گزارش ایجاد نمایید. گزارش می تواند لیست اطلاعات ، برچسب چاپی ، صورت حساب و ... باشد.

قبل از چاپ گزارش به وسیله ی چاپگر ، ابتدا آن را در نمای Preview مشاهده کنید. این حالت نمای گزارش را روی صفحه نشان می دهد و مشابه نمونه ی چاپی است. در این حالت متوجه تغییرات مورد نیاز خواهید شد و در زمان و کاغذ صرفه جویی می شود. با انتخاب دکمه ی Preview در پنجره ی ابزار بانک ، گزارش موجود را در صفحه ی پنجره مشاهده کنید (شکل 1-6) .



شکل 1-6 : پنجره ی نمای Preview یک گزارش نمونه

معمولا نباید تمام داده ها در یک گزارش نمایش داده شوند ، مگر گزارش برای نمایش جزئیات باشد مثل: لیست موجودی انبار ، لیست مشتریان و غیره. به طور کل بخشی از اطلاعات و مقادیر جدول ها که مفید و برای تصمیم گیری مورد نیاز است نمایش داده می شوند.

بهرتر است گزارش‌ها از داده‌های پرسوجوها ایجاد می‌شوند ، هرچند می‌توانیم جدول‌ها را در گزارش اضافه کرده و مانند یک پرسوجو عمل کنیم ولی بهتر است برای نمایش اطلاعات چند جدول و حتی یک جدول ، ابتدا پرسوجویی که داده‌های موردنیاز را برگرداند ساخته شود و از آن پرسوجو در ایجاد گزارش استفاده گردد.

### **1-6- ایجاد گزارش ساده به وسیله ی AutoReports**

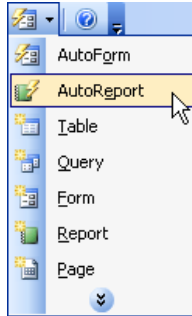
با این امکان در بانک اطلاعاتی Access ، می‌توان در نمای Datasheet ، گزارش‌ها را به سرعت ایجاد می‌کند. این نوع گزارش فقط نام فیلدها و مقادیر متناظر را لیست می‌کند (شکل 2-6).

تفاوت این لیست با چاپ نمای Datasheet آن است که داده‌ها در این نما قالب‌بندی می‌شود و به صورت خوانا شکل می‌گیرد بدون آن‌که اطلاعات زیادی را در یک صفحه فشرده سازد.

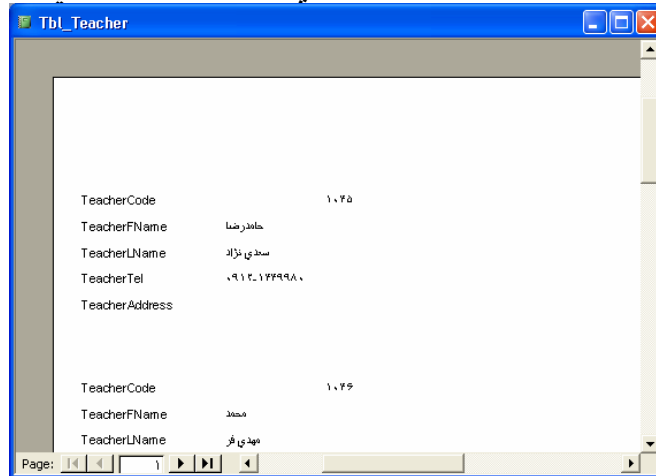


### مثال 6-1 :

می‌خواهیم گزارش سریعی از معلمان مدرسه ایجاد کنیم :  
جدول یا پرس‌وجوی موردنظر را انتخاب نمایید ، مثلا جدول Tbl\_Teacher.  
آن را در نمایی Datasheet باز کنید.  
در صورت نیاز برای محدود کردن رکوردها ، Filter اعمال نمایید.  
روی فلش سمت پایین در نوار ابزار مطابق شکل زیر ، کلیک کنید.

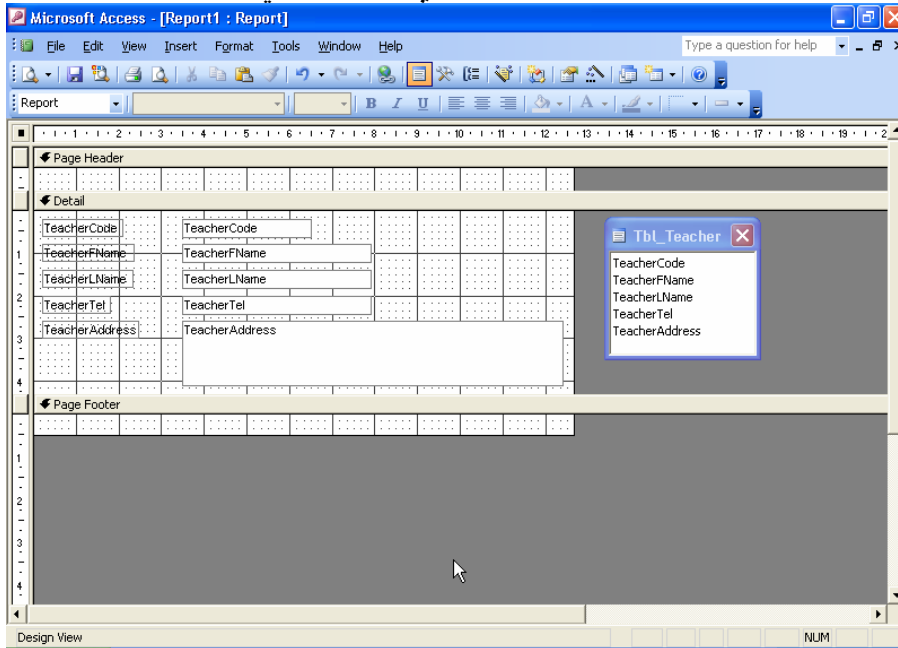


- از لیست ظاهر شده ، گزینه‌ی AutoReport را انتخاب نمایید. بانک گزارشی از نمایی Datasheet ساخته و آن را در نمایی Preview نشان می‌دهد.



## ادامه‌ی مثال 6-1 :

- اگر نمایش گزارش مطابق آنچه می‌خواهید باشد ، آن را با دکمه‌ی Print چاپ کنید.
- دکمه‌ی Design را کلیک کنید تا طرح گزارش را ببینید (شکل 4-6). به خاطر داشته باشید که شاید درک این نما برای بار اول کمی مشکل به نظر رسد.



- نگران ارقام خاص روی صفحه نباشید. AutoReport طراحی گزارش را ایجاد کرده است و نیازی به استفاده از ابزارهای نمای Design و ساخت گزارش نداریم.
- نمای Design را ببندید. در صورتی که می‌خواهید آن را ذخیره نمایید ، نامی به آن اختصاص دهید. زمانی آن را ذخیره کنید که مجدداً به آن نیاز پیدا خواهید کرد.

### نکته :

برای ایجاد گزارش‌های استاندارد و مناسب‌تر و همچنین تا حدودی ساده ، از گزینه‌ی Wizard استفاده می‌شود.

**نکته :**

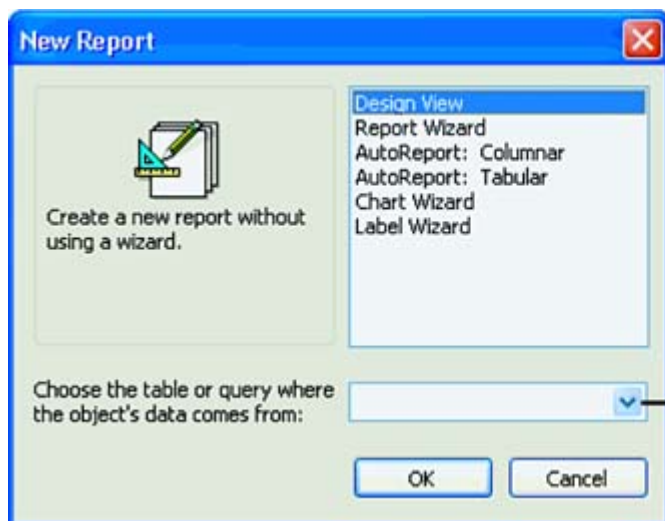
**AutoReport :** ابزارهاي طراحی گزارش را برای ایجاد آن به کار می برد. زمانی که یادگرفتید چگونه از این ابزارها استفاده کنید و گزارش را اصلاح کنید و تغییر دهید ، می توانید از این امکان برای ایجاد گزارش اولیه استفاده کنید و سپس گزارش را مطابق نظر خود تغییر دهید.

**کنجکاو :**

AutoForm چه کاری انجام می دهد ، کجا در دسترس است؟

## 2-6- ایجاد گزارش به وسیله Report Wizard

روی پنجره بانک ، گزینه Reports را انتخاب کنید. از پنجره باز شده Report Wizard را انتخاب نمایید و در قسمت Choose the table or query ، جدول یا پرسوجوی مورد نظر را انتخاب کنید (شکل 5-6).

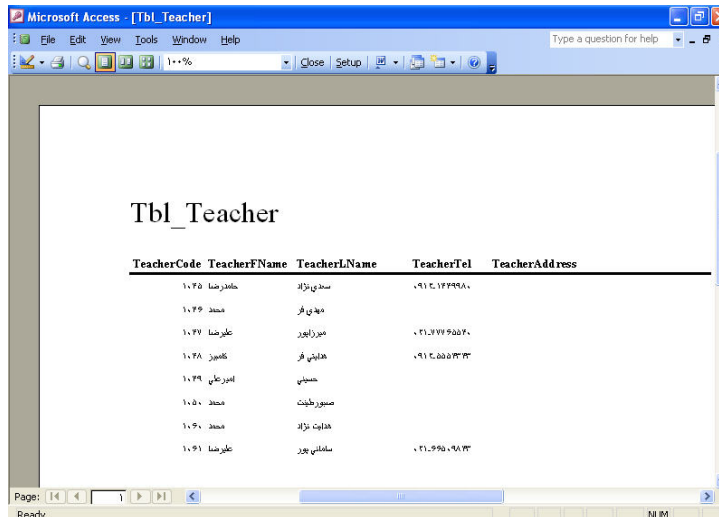


برای انتخاب جدول یا پرسوجو

شکل 5-6 : انتخاب ایجاد پرسوجو

Access ، ابزارهاي متنوعي براي ايجاد گزارش دارد. همانطور كه در شكل مي بينيد اين روشها به اجمال عبارت است از :

- **Design View** : گزارش خالي را در نماي Design باز مي كند. در اين نما مي توانيد عناوين ، پانوشتها ، جزييات و خلاصه ي داده ها را اضافه كنيد. زماني كه بخواهيد گزارش را از ابتدا ايجاد كنيد از اين گزينه استفاده نماييد.
  - **Report Wizard** : با دنبال كردن قدم هاي فرايند توليد گزارش ، يعني انتخاب منابع داده (جدولها/پرسوجوها) ، فيلدهاي موردنظر در گزارش و غيره گزارش توليد مي شود. اين روش ايجاد گزارش معمولترين است.
  - **AutoReport Columnar** : گزارشي را ايجاد مي كند كه تمام فيلدهاي جدول يا پرسوجوي انتخاب شده را دربر دارد. شبيه AutoReport است. با اين تفاوت كه از فضاي صفحه استفاده ي بهتري دارد.
  - **AutoReport Tabular** : گزارشي توليد مي كند كه ركوردهاي جدول/پرسوجو را در يك ردیف نشان مي دهد و اندازه ي قلم را به گونه اي اختيار مي كند كه صفحه ي گزارش تنظيم شود. اين نوع گزارش بهتر از نماي Datasheet ساده به نظر مي آيد و اغلب مفيدتر از AutoReport هاي قبلي است. (شكل 6-6).
- Access معمولاً نمي تواند عناوين فيلدها را در بالاي گزارش به صورت صحيح قرار دهد. براي تنظيم دقيق همانطور كه در ادامه خواهيم ديد بايد در نماي Design آن را اصلاح نماييد.

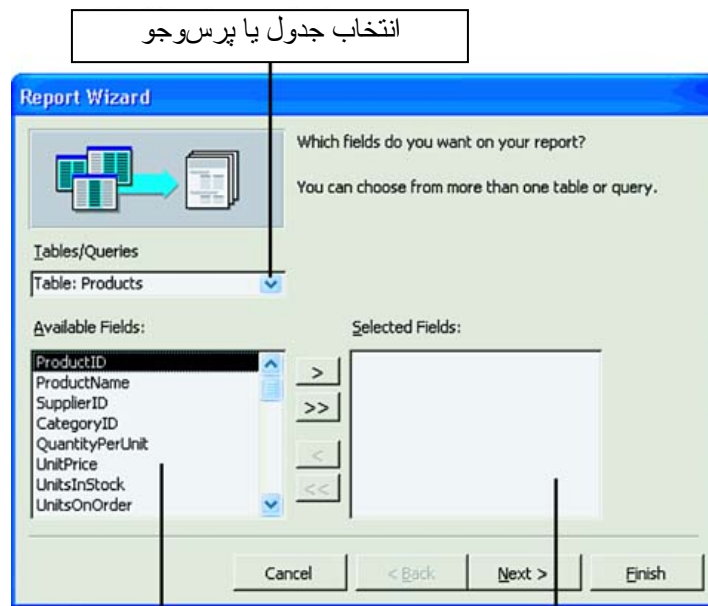


شكل 6-5 : پنجره ي نماي Preview گزارش ايجاد شده به وسيله ي Wizard

- **Chart Wizard** : گراف داده ها را نمايش مي دهد. گراف اين محيط شبيه محيط Excel است و به همان صورت قابل كنترل است.
- **Label Wizard** : ليستي از برچسبها ايجاد مي كند. برچسب پستي ، استاندارد شماره گذاري خاص خود را دارد كه فراهم شده است.

## مثال 2-6 :

- با کمک ویزارد گزارشی از لیست معلم‌های مدرسه و از نوع Tabular ایجاد کنید.
- در قسمت گزارش‌های بانک ، دکمه‌ی New و سپس Report Wizard را انتخاب کنید و OK نمایید.
- پرس‌جویی ایجاد کنید که تمام رکوردهای جدول معلمان را برگرداند و آن را Qry\_Teachers نام‌گذاری کنید.
- در پنجره‌ی باز شده ، در قسمت Tables/Queries جدول Qry\_Teachers را انتخاب کنید.



فیلدهای در دسترس

اسامی فیلدهای انتخاب‌شده

تمام فیلدها

## ادامه‌ی مثال 2-6 :

– قسمت Selected Field ، فیلدهای انتخاب شده برای نمایش در گزارش را نشان می‌دهد. فیلدها به وسیله‌ی دکمه‌های > یا >> به لیست اضافه می‌شوند. همچنین می‌توان روی فیلد در لیست Available Field دوبار کلیک کرد.

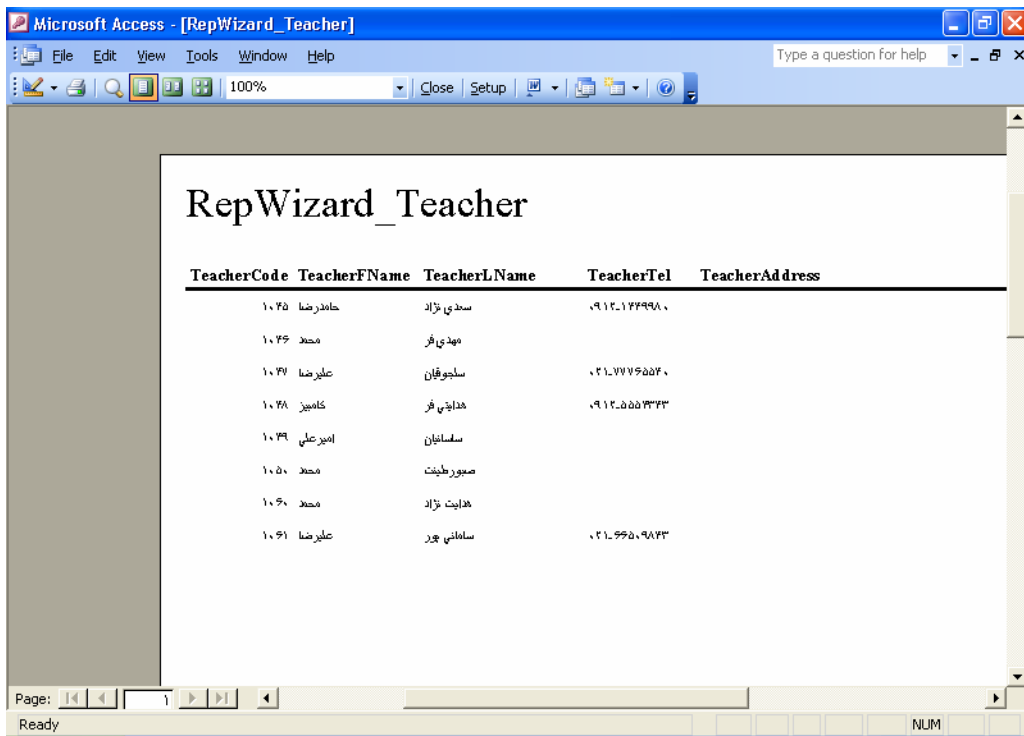
**نکته :**

در قسمت Tables/Queries می‌توان جدول‌های دیگر را نیز انتخاب کرده ، فیلدهای موردنظر را به فیلدهای انتخاب شده‌ی خود اضافه کرد. اگرچه گفتیم بهتر است ابتدا پرس‌وجویی که داده‌های موردنیاز را برمی‌گرداند ، انتخاب کنید و سپس از این پرس‌وجو در گزارش استفاده نمایید.

– بعد از انتخاب فیلدهای موردنیاز ، دکمه‌ی Next را کلیک کنید.

– سایر مراحل را بدون تغییر تایید کنید و در انتها Finish را انتخاب کنید.

– گزارش را در حالت Design یا Preview مشاهده کنید. (شکل زیر)



The screenshot shows the Microsoft Access interface with a report titled "RepWizard\_Teacher". The report displays a table with the following columns: TeacherCode, TeacherFName, TeacherLName, TeacherTel, and TeacherAddress. The data is as follows:

TeacherCode	TeacherFName	TeacherLName	TeacherTel	TeacherAddress
۱۰۴۵	حامدرضا	سعدی نژاد	۰۹۱۲۰۱۴۴۹۹۸۰	
۱۰۴۶	محمد	مهدي فر		
۱۰۴۷	علیرضا	سلجوقیان	۰۲۱۰۷۷۶۵۵۴۰	
۱۰۴۸	کامیلا	فرداوی فر	۰۹۱۲۰۵۵۳۳۴۳	
۱۰۴۹	امیرعلی	سماغان		
۱۰۵۰	محمد	صورتیخت		
۱۰۶۰	محمد	هدایت نژاد		
۱۰۶۱	علیرضا	سماغانی پور	۰۲۱۰۶۶۵۰۹۸۴۳	

### کنجکاو ي :

قسمت Sort ، Grouping و Layout/Style در ویزارد طراحی گزارش را توضیح دهید؟ چه عملی انجام می‌دهند؟

### مثال 3-6 :

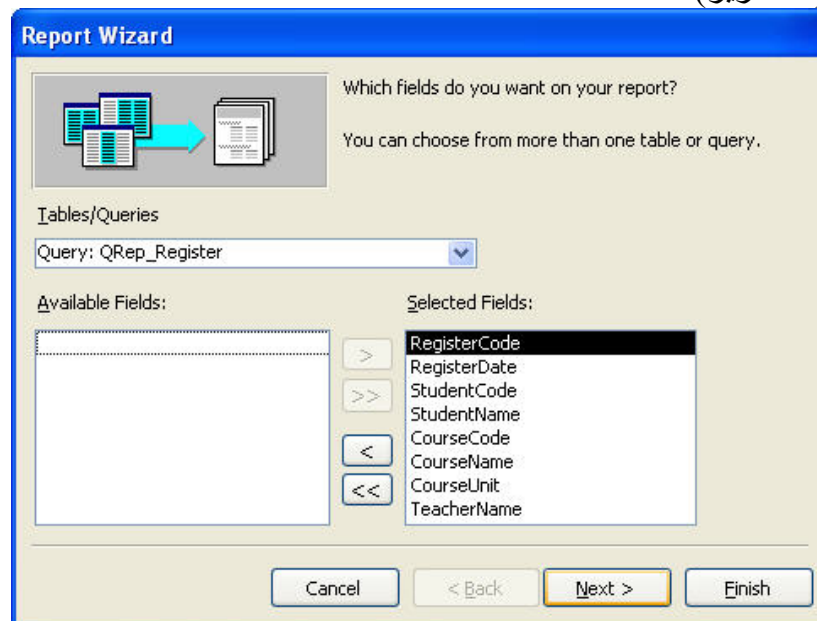
گزارش ثبت‌نام دانش‌آموزان در ترم را همان‌گونه که در طراحی در فصل اول مشخص شده است ، ایجاد کنید .

– ابتدا پرس‌وجویی به نام QRep\_Register ایجاد می‌کنیم که اطلاعات موردنیاز را در بر داشته باشد . SQL موردنظر در زیر آمده است :

```
SELECT Tbl_Register.RegisterCode, Tbl_Register.RegisterDate,
Tbl_Register.StudentCode, [StudentFName]+'+[StudentLName] AS StudentName,
Tbl_RegisterItem.CourseCode, Tbl_Course.CourseName, Tbl_Course.CourseUnit,
Tbl_Teacher.TeacherFName + ' ' +Tbl_Teacher.TeacherLName AS TeacherName
FROM Tbl_Teacher INNER JOIN (Tbl_Course INNER JOIN (Tbl_Student INNER JOIN
(Tbl_Register INNER JOIN Tbl_RegisterItem ON Tbl_Register.RegisterCode =
Tbl_RegisterItem.RegisterCode) ON Tbl_Student.StudentCode =
Tbl_Register.StudentCode) ON Tbl_Course.CourseCode = Tbl_RegisterItem.CourseCode)
ON Tbl_Teacher.TeacherCode = Tbl_Course.TeacherCode
```

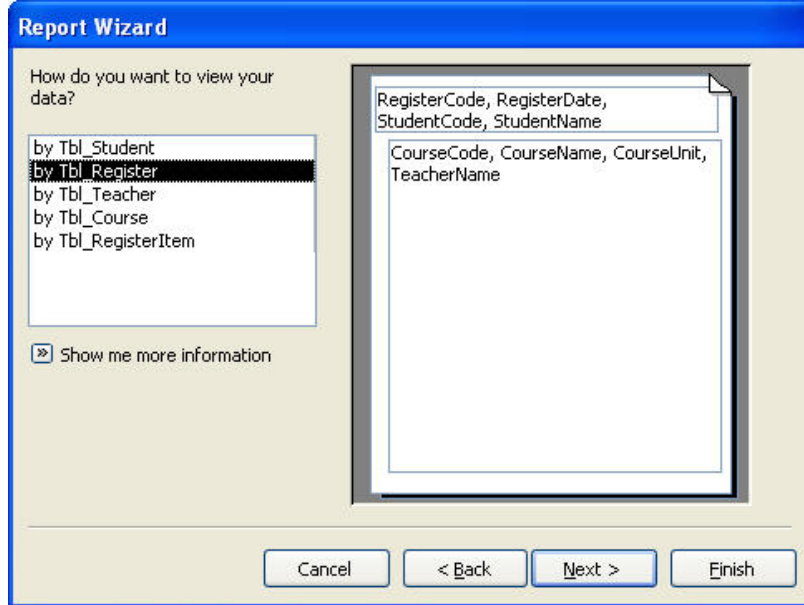
– ویزارد گزارش Report Wizard را اجرا می‌کنیم و پرس‌وجوی QRep\_Register را انتخاب می‌کنیم.

– تمام فیلدها را به وسیله دکمه >> به لیست اضافه می‌کنیم و دکمه Next را انتخاب می‌کنیم. (شکل زیر)



### ادامه‌ی مثال 3-6 :

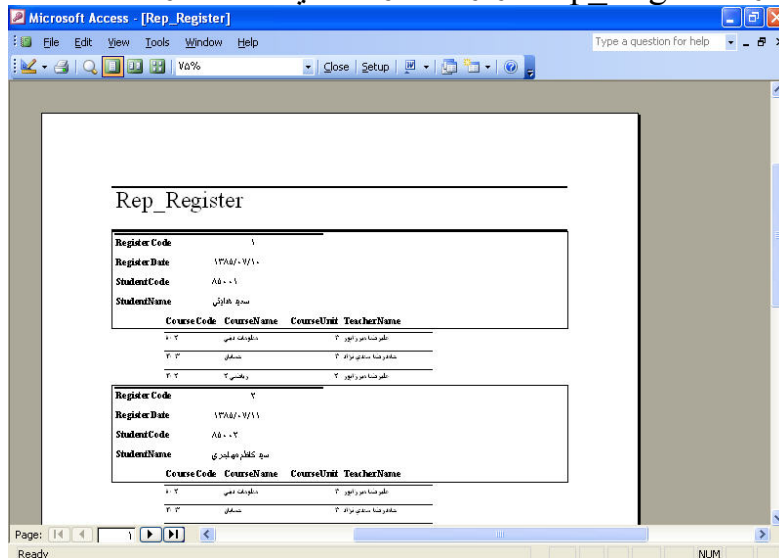
– از پنجره‌ی باز شده گزینه‌ی by Tbl\_Register را انتخاب و دکمه‌ی Next را کلیک کنید.



– پیش‌فرض Group level و Sort Order را بدون تغییر تایید کنید ، در پنجره‌ی Lay out مطابق شکل زیر گزینه‌ی Align Left2 را انتخاب کنید و دکمه‌ی Next را کلیک کنید.

– در پنجره‌ی Style ، پیش‌فرض Formal را انتخاب کرده و دکمه‌ی Next را انتخاب کنید.

– نام گزارش را Rep\_Register قرار دهید و دکمه‌ی Finish را انتخاب کنید.

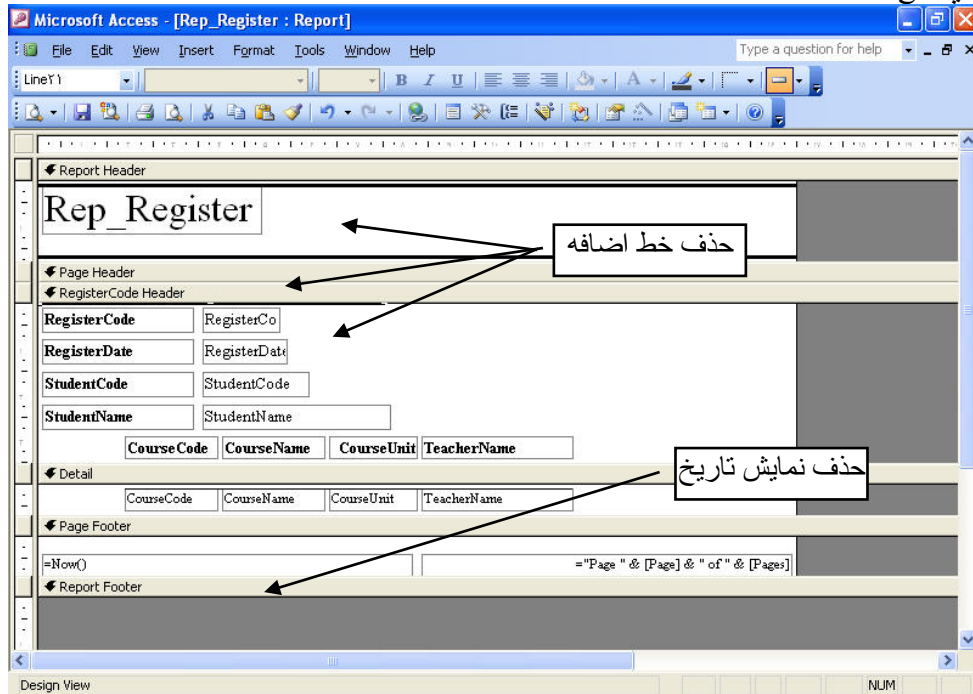


– گزارش از لحاظ شکل ظاهر مشابه گزارش طراحی شده نیست ، بنابراین گزارش را ذخیره می‌کنیم تا در نمای Design آن را تغییر می‌دهیم.




### ادامه‌ی مثال 3-6 :

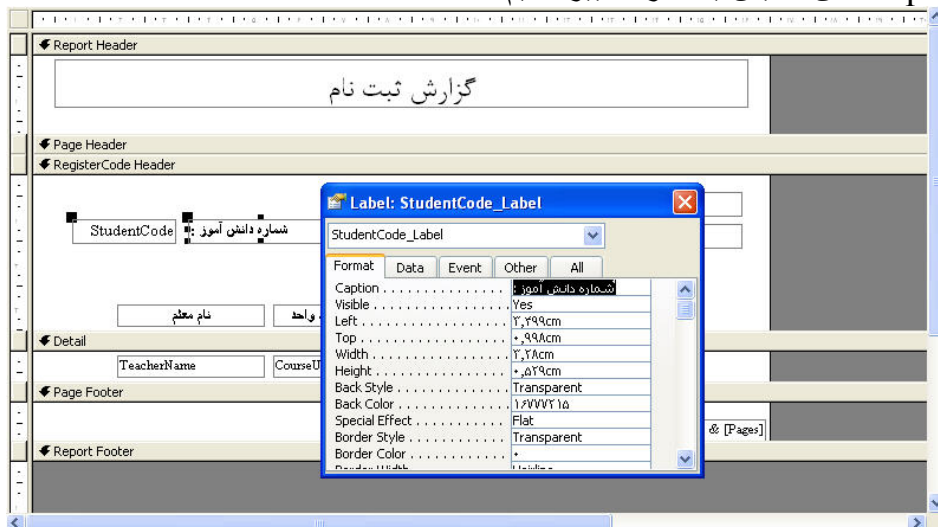
– نمای Design این گزارش به صورت زیر است :



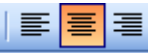
– در این نما خطوط اضافه و موارد غیر ضروری مثل تاریخ جاری سیستم را حذف می‌کنیم.

انتخاب مورد و با زدن دکمه‌ی delete صفحه کلید.

– متن لاتین را به فارسی تغییر می‌دهیم. انتخاب جعبه متن مورد نظر و کلیک مجدد و تایپ فارسی متن مورد نظر انجام می‌دهیم. همچنین می‌توانیم از طریق پنجره‌ی مشخصه‌ها که از طریق دکمه‌ی  روی نوار ابزار قابل مشاهده است. زبانه‌ی Format، گزینه‌ی Caption متن نمایش یافته را تغییر دهیم.



### ادامه‌ی مثال 3-6 :

- جعبه متن‌های مشابه روش‌هایی که در تغییر طراحی فرم دیدید ، جابجا می‌شوند. یادآوری : مثلا درگ کردن از طریق مربع بزرگ سمت راست جعبه متن که با انتخاب آن ظاهر می‌گردد.
- همچنین پس از انتخاب جعبه متن موردنظر ، تنظیم چیدمان متن درون آن (Align) را با کمک دکمه‌های  در نوار ابزار و بر اساس مورد وسط‌چین یا راست‌چین قرار می‌دهیم. همچنین می‌توانیم از گزینه‌ی Text Align در زبانه‌ی Format پنجره‌ی مشخصه‌ها نیز این‌کار را انجام دهیم.
- برای افزودن امکانات گرافیکی ، از منوی View ، گزینه‌ی Toolbox را انتخاب می‌نماییم. پنجره‌ی زیر ظاهر می‌شود. این جعبه دقیقا مشابه ابزار محیط طراحی فرم است.



- با کمک ابزار Rectangle (مستطیل) و Line (خط) طراحی گزارش موردنظر خود را ایجاد می‌کنیم. (شکل زیر)



The screenshot shows a report design interface with the following sections:

- Report Header:** A text box containing "گزارش ثبت نام".
- Page Header:** A section containing a "RegisterCode Header".
- RegisterCode Header:** A section containing four text boxes: "نام دانش آموز:" (StudentName), "تاریخ ثبت نام:" (RegisterDate), "شماره دانش آموز:" (StudentCode), and "نام معلم" (TeacherName).
- Detail:** A section containing four text boxes: "شماره درس" (CourseCode), "نام درس" (CourseName), "تعداد واحد" (CourseUnit), and "نام معلم" (TeacherName).
- Page Footer:** A section containing a text box with the expression: "Page " & [Page] & " of " & [Pages]".
- Report Footer:** A section containing a text box with the expression: "Page " & [Page] & " of " & [Pages]".

### ادامه‌ی مثال 3-6 :

– دقت کنید که برای وضوح بیشتر بین دو بخش عنوان‌های درس و اقلام داده‌ای فاصله ایجاد کرده‌ایم که در عمل برای زیبایی نباید وجود داشته باشند. (شکل زیر)

شماره درس	نام درس	تعداد واحد	نام معلم
CourseCode	CourseName	CourseUnit	TeacherName

– سپس گزارش را ذخیره و در نمای Preview مشاهده می‌کنیم. (شکل زیر)

گزارش ثبت نام

تاریخ ثبت نام : ۱۳۸۵/۰۷/۱۰  
 نام دانش آموز : سعید هدایتی

شماره دانش آموز : ۸۵۰۰۱

شماره درس	نام درس	تعداد واحد	نام معلم
۵۰۲	معلومات دومی	۳	علیرضا میرزاپور
۴۰۳	حصان	۳	حامد رضا سدی نژاد
۴۰۲	ریاضی ۲	۲	علیرضا میرزاپور

– گزارش‌های لیست حضور و غیاب نیز به صورت مشابه ایجاد می‌شود.

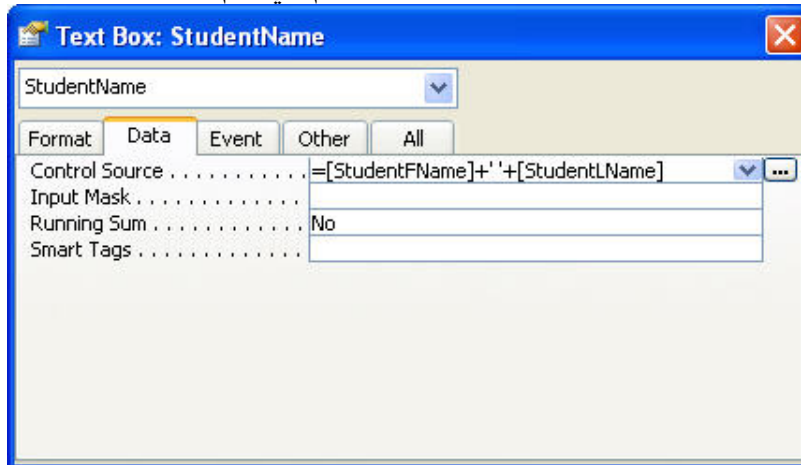
#### مثال 4-6 :

گزارش کارنامه دانش‌آموزان ایجاد می‌کنیم.

- این گزارش مشابه گزارش‌های قبلی ایجاد می‌شود ، فقط کمی پیچیدگی در محاسبه‌ی معدل وجود دارد که به آن می‌پردازیم.
- ابتدا پرس‌وجوی موردنظر را ایجاد می‌کنیم. به نام QRep\_Certificate و با SQL زیر :

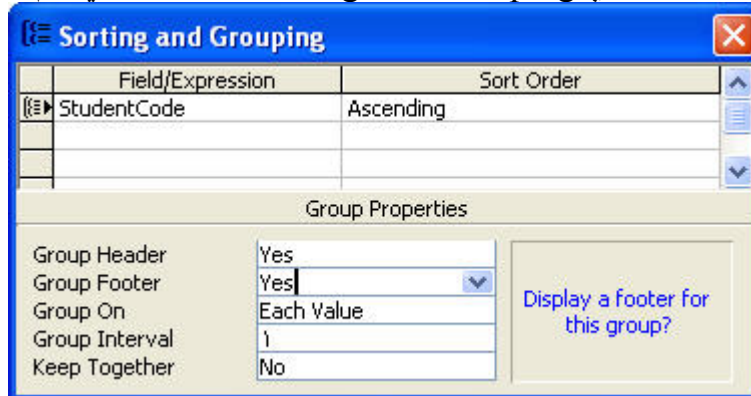
```
SELECT Tbl_Student.StudentCode, StudentFName, StudentLName, RegisterDate,
MajorName, Tbl_RegisterItem.CourseCode, CourseName, CourseUnit, Grade
FROM Tbl_Course INNER JOIN (Tbl_Major INNER JOIN ((Tbl_Student INNER JOIN
Tbl_Register ON Tbl_Student.StudentCode = Tbl_Register.StudentCode) INNER JOIN
Tbl_RegisterItem ON Tbl_Register.RegisterCode = Tbl_RegisterItem.RegisterCode) ON
Tbl_Major.MajorCode = Tbl_Student.MajorCode) ON Tbl_Course.CourseCode =
Tbl_RegisterItem.CourseCode
```

- با کمک ویزارد و این پرس‌وجو گزارش اولیه‌ی مشابه مثال قبلی و به نام Rep\_Certificate ایجاد می‌کنیم.
- دقت نمایی که ما در پرس‌وجو نام و نام‌خانوادگی را ترکیب نکرده‌ایم ، بنابراین باید در گزارش این کار صورت گیرد. برای این منظور ، در پنجره‌ی مشخصه‌ها ، زبانه Data ، گزینه‌ی Control Source به صورت زیر تنظیم می‌کنیم :



#### ادامه ي مثال 4-6 :

– از منوي View ، گزينه ي Sorting and Grouping را انتخاب مي كنيم.



– در اين پنجره گزينه ي Group Footer را مقدار Yes انتخاب مي كنيم ، تا پانوشت گروه بندي نيز نمايش داده شود. جعبه متن هاي مربوط به جمع واحدها و محاسبه ي معدل بايد در اين قسمت قرار گيرد.

– از جعبه ابزار ، دکمه ي Text box را انتخاب مي كنيم و با درگ کردن ، آن را در قسمت پانوشت گروه قرار مي دهيم. مقدار Visible آن را در پنجره ي مشخصه ها No انتخاب مي كنيم و مقدار Control Source آن را  $\text{SUM}([\text{CourseUnit}] * [\text{Grade}])$  قرار مي دهيم. اين فيلد ، حاصل جمع حاصل ضرب نمره در واحد درسي را بر اساس گروه بندي محاسبه مي کند چراکه در پانوشت گروه قرار داده شده است. نام آن را در زبانه ي All يا Other ، گزينه ي Name مقدار SumGradeByUnit قرار مي دهيم.

– Text box ديگري را در قسمت پانوشت گروه ، زير ستون واحدها قرار مي دهيم و مقدار Control Source آن را  $\text{Sum}([\text{CourseUnit}])$  قرار مي دهيم. اين فيلد جمع کل واحدهاي ثبت نام شده ي دانش آموز را بر مي گرداند ، زيرا در پانوشت گروه قرار گرفته است و جمع را بر اساس گروه محاسبه مي کند. نام آن را به وسيله ي گزينه ي Name در پنجره ي مشخصه ها ، مقدار SumUnit قرار مي دهيم.

– Text box ديگري را در قسمت پانوشت گروه قرار مي دهيم و مقدار Control Source آن را  $\text{Format}([\text{SumUnit}], \text{"\#\#\#"})$  قرار مي دهيم. همچنين گزينه ي Border Style را مقدار Solid قرار مي دهيم.

ادامه ي مثال 4-6 :

- طراحی گزارش را در شکل زیر مشاهده می کنید .

The screenshot shows the design view of a report in Microsoft Access. The report is titled 'گزارش کارنامه دانش آموز'. It is divided into several sections:

- Report Header:** Contains the title 'گزارش کارنامه دانش آموز' and a date field 'تاریخ گزارش: -Now()'. The title is centered, and the date is on the right.
- Page Header:** Contains the text 'گزارش کارنامه دانش آموز'.
- StudentCode Header:** Contains fields for 'StudentCode' (شماره دانش آموز), 'FName'+'+[StudentLName]' (نام دانش آموز), 'MajorName' (رشته تحصیلی), and 'gisterDate' (تاریخ ثبت نام).
- Table:** A table with columns: 'شماره درس', 'نام درس', 'تعداد واحد', and 'نمره'.
- Detail:** A table with columns: 'CourseCode', 'CourseName', 'CourseUnit', and 'Grade'.
- StudentCode Footer:** Contains summary fields: '-Sum([Course Unit]) : جمع واحدها' and 'UMGradeByUnit/[SumUnit] : معدل'. Below this is another field: '-Sum([CourseUnit]\*[Grade])'.
- Page Footer:** Contains the text 'گزارش کارنامه دانش آموز'.

- نمایی Preview گزارش به صورت زیر است.

The screenshot shows the preview view of the report in Microsoft Access. The report is titled 'گزارش کارنامه دانش آموز'. It displays the following information:

- Header:** 'گزارش کارنامه دانش آموز'.
- Student Information:**
  - تاریخ ثبت نام: ۱۳۸۵/۰۷/۱۰
  - نام دانش آموز: سعید هدایتی
  - رشته تحصیلی: ریاضی و فیزیک
- Table:** A table with columns: 'شماره درس', 'نام درس', 'تعداد واحد', and 'نمره'.
 

شماره درس	نام درس	تعداد واحد	نمره
۵۰۲	معلومات دبیری	۳	۱۴
۵۰۲	معلومات دبیری	۳	۱۷
۴۰۲	حسابان	۳	۹
۴۰۲	حسابان	۳	۱۹
۴۰۲	ریاضی ۲	۲	۱۳
۴۰۲	ریاضی ۲	۲	۱۷
جمع واحدها:		۱۶	
معدل:		۱۴٫۸۱	
- Footer:** 'گزارش کارنامه دانش آموز'.

#### نکته :

محیط طراحی گزارش‌ها مشابه محیط طراحی فرم است و ابزارهای محیط ، منوها ، کنترل‌ها ، قابلیت‌ها و چگونگی استفاده از آن‌ها مشابه است . بنابراین از ذکر مجدد آن در این قسمت خودداری می‌کنیم.

### خلاصه‌ی فصل

برای چاپ اطلاعات با قالبی زیبا و مناسب از گزارش استفاده می‌کنیم . گزارش نمای قابل چاپ را روی صفحه‌ی نمایش ارائه می‌دهد . می‌توانیم اطلاعات را در گزارش‌ها گروه بندی کنیم و گزارش‌های با جزئیات و یا اطلاعات خلاصه ایجاد کنیم . محیط ایجاد گزارش در Access ، شبیه برنامه‌های گزارش‌ساز در محیط ویندوز است . مثلاً همان طور که در محیط Word ، اطلاعات چاپی را در نمای Preview مشاهده می‌کردید و یا بخشی از آن را چاپ می‌کردید ، در محیط Access نیز همان امکانات برای گزارش ایجاد شده وجود دارد . اشیای بانک مشابه کنترل‌های فرم هستند و همان عملیات مشخصات ظاهری کنترل‌ها ، جابه جایی آن‌ها و ... روی آن قابل انجام است .

### خودآزمایی

1. گزارش چیست؟
2. روش‌های ایجاد گزارش را توضیح دهید؟
3. به نظر شما کدامیک از روش‌های ساخت گزارش مناسبتر است و چرا؟
4. کنترل‌های گزارش را بررسی کنید ، آیا اختلافی بین آن‌ها و کنترل‌های فرم وجود دارد؟

## 7- فصل هفتم : نرمال سازي

پس از شناسايي موجوديتها و رسم ER ، صفات موردنياز و ارتباط بين موجوديتها ، مي‌بايد جدول‌ها (روابط) موردنياز طراحي شوند. يك تصور نادرست اين است كه هرچه صفات جدول بيشتر باشد ، از اكمال و جامعيت بالاتري برخوردار است ، حال آن‌كه گزينش درست صفات ، از بروز برخي مشكلات در بانك‌هاي اطلاعاتي اجتناب مي‌كند. نارسايي‌ها در طراحي و يك طراحي نامناسب ، باعث ايجاد پديدهي "آنومالي"<sup>1</sup> مي‌گردد.

آنومالي در سه محور قابل طرح و بررسي است :

**الف)** "انجام ناپذيري" يكي از عمليات در بانك.  
**ب)** "بروز تبعات نامطلوب" در پي انجام يك عمليات مبنايي.

**ج)** "فزون كاري" در پي انجام يك عمليات مبنايي.

قبل از شرح موارد آنومالي ، ويژگي‌هاي رابطيه و چند مفهوم مقدماتي ديگر ذكر مي‌شود ، سپس در قالب يك مثال آنومالي و راه‌هاي رفع آن گفته خواهد شد.

### 7-5-1- "ويژگي‌هاي رابطه"

رابطه كه در بانك‌هاي اطلاعاتي رابطه‌اي<sup>2</sup> ، به صورت جدول پياده‌سازي مي‌شود داراي ويژگي‌هاي زير است :

**الف)** منظم نيست ، به اين معني كه ترتيب صفات ، در هنگام تعريف بي‌اهميت است و سطرهاي جدول<sup>3</sup> نظم ندارد.

**ب)** داراي صفات تك مقداري است يعني يه اجزاي معني‌دار تقسيم نمي‌شود. به عنوان مثال اگر مقدار شماره‌ي دانش‌آموزي برابر 851743 باشد و 85 نشان‌دهنده‌ي سال تحصيلي باشد و در پرسوجوها بخوايم دانش‌آموزاني كه در سال 1385 ثبت‌نام کرده‌اند را مشخص كنيم ، فيلد "شماره‌ي دانش‌آموزي" يك فيلد تك‌مقداري نيست.

**ج)** سطرهاي جدول ، تكراري ندارد.

### 7-5-2- "وابستگي تابعي"<sup>4</sup> :

<sup>1</sup> - Anomaly

<sup>2</sup> - Relational Data Base

<sup>3</sup> - به سطرهاي جدول در اين مدل داده‌اي اصطلاحاً تاپل گفته مي‌شود.

<sup>4</sup> - Functional Dependency



شبيه تعريف تابع در رياضيات ، در رابطه ي  $R$  ، صفت  $y$  با صفت  $x^1$  وابستگي-تابعي دارد اگر به ازاي هر مقدار  $x$  ، تنها يك مقدار  $y$  وجود داشته باشد. وابستگي-تابعي  $A \rightarrow B$  و  $A$  را به صورت  $A \rightarrow B$  نشان مي-دهيم<sup>2</sup>.

### مثال 7-1 :

در رابطه ي  $R1$  ، وابستگي-تابعي  $A \rightarrow B$  و  $B \rightarrow C$  را بررسي كنيد :

R1		
A	B	C
$a_1$	$b_1$	$c_1$
$a_2$	$b_2$	$c_3$
$a_1$	$b_1$	$c_2$
$a_3$	$b_4$	$c_2$
$a_5$	$b_1$	$c_1$

پاسخ:

وابستگي  $A \rightarrow B$  برقرار است. زيرا به ازاي هيچ مقدار مساوي از  $A$  ، دو مقدار متفاوت از  $B$  وجود ندارد. اما وابستگي  $B \rightarrow C$  برقرار نيست. زيرا كه به ازاي  $b_1$  ، دو مقدار  $c_1$  و  $c_2$  در  $R$  وجود دارد.

وابستگي-تابعي بيان-كننده ي قواعد محيط عملياتي است. اين قواعد را مي-توان در نمودار "وابستگي-تابعي" يا نمودار  $FD$  نمايش داد.

### مثال 7-2 :

فرض كنيد قواعد زير در يك محيط عملياتي برقرار است. نمودار  $FD$  متناظر را رسم كنيد. يك جدول با مقادير دلخواه متناظر با آن ايجاد كنيد.

قاعده ي 1 : هر ناشر ، تعدادي كتاب منتشر مي-كند.

قاعده ي 2 : هر ناشر از يك كتاب ، شمارگان مشخصي منتشر مي-كند.

قاعده ي 3 : هر ناشر در يك شهر دفتر دارد.

قاعده ي 4 : هر ناشر داراي يك رتبه ي صنف ي است.

<sup>1</sup> -  $x$  و  $y$  مي-توانند صفات خاصه ي ساده يا مركب باشند.  
<sup>2</sup> -  $A$  را اصطلاحاً دترمينا ن گويند.

ادامه ي مثال 7-2 :

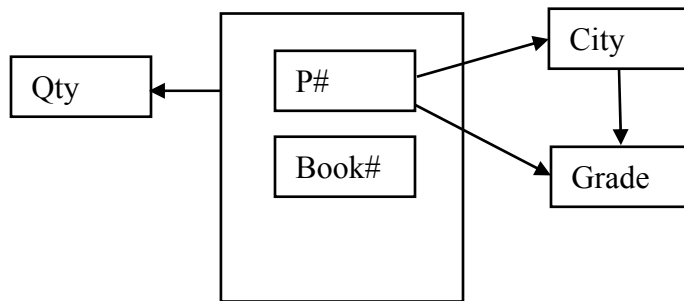
قاعده ي 5 : ناشران يك شهر داراي يك رتبه ي صنف ي هستند .

پاسخ :

با توجه به قواعد بالا داريم :

(P#=شماره ناشر ، Book#=شماره كتاب ، Qty=شمارگان ، City=شهر و Grade=رتبه صنف ي ناشر)

- 1) P#  $\rightarrow$  Book#
- 2) (P# , Book#)  $\rightarrow$  Qty
- 3) P#  $\rightarrow$  City
- 4) P#  $\rightarrow$  Grade
- 5) City  $\rightarrow$  Grade



FIRST

P#	Book#	Qty	City	Grade
P <sub>1</sub>	b <sub>1</sub>	1000	c <sub>1</sub>	40
P <sub>2</sub>	b <sub>4</sub>	3000	c <sub>2</sub>	60
P <sub>3</sub>	b <sub>3</sub>	2000	c <sub>1</sub>	40
P <sub>1</sub>	b <sub>2</sub>	3000	c <sub>1</sub>	40
P <sub>4</sub>	b <sub>3</sub>	4000	c <sub>3</sub>	50
P <sub>4</sub>	b <sub>5</sub>	3500	c <sub>3</sub>	50

### تمرین 7-1 :

در يك محیط عملیاتی چهار قاعده بیان کنید و نمودار FD متناظر را رسم کنید.

### نکته :

تمام صفات يك موجودیت با صفت کلید آن، وابستگی تابعی دارند.

### مثال 7-3 :

اگر  $A \rightarrow B$  و  $B \rightarrow C$  برقرار باشد، آیا  $A \rightarrow C$  برقرار است؟  
پاسخ :

فرض می‌کنیم  $A \rightarrow C$  در این صورت داریم :

$a_1 \dots c_1$

$\rightarrow$

$a_1 \dots c_2$

دو حالت زیر قابل تصور است :

$a_1 \ b_1 \ c_1$

$a_1 \ b_1 \ c_1$

$a_1 \ b_1 \ c_2$

$a_1 \ b_2 \ c_2$

در هر دو حالت با تناقض در فرض مواجه می‌شویم. در قسمت (الف)  $A \rightarrow C$  و  
در قسمت (ب)  $A \rightarrow B$  در نتیجه طبق برهان خلف داریم :  $A \rightarrow C$ .

$\rightarrow$

$\rightarrow$

$\rightarrow$

### 3-5-7- "وابستگی تابعی کامل" <sup>1</sup> :

صفت  $y$  ، با صفت  $x$  وابستگی تابعی کامل دارد  $(x \Rightarrow y)$  اگر

4.  $y$  با  $x$  وابستگی تابعی داشته باشد.

5.  $y$  با هیچ زیرمجموعه‌ی  $x$  وابستگی تابعی نداشته باشد.

<sup>1</sup> - Fully Functional Dependency

**مثال 4-7 :**

با توجه به R2 تحقیق کنید ، آیا وابستگی تابعی کامل بین C و (A و B) برقرار است؟

R2

A	B	C
a <sub>1</sub>	B <sub>1</sub>	c <sub>1</sub>
a <sub>2</sub>	B <sub>1</sub>	c <sub>3</sub>
a <sub>1</sub>	b <sub>2</sub>	c <sub>2</sub>

**پاسخ :**

شرط اول برقرار است. یعنی :

(A , B) C

شرط دوم نیز برقرار است. زیرا :

A C ((a<sub>1</sub>, c<sub>1</sub>) , (a<sub>1</sub>, c<sub>2</sub>))

B C ((b<sub>1</sub>, c<sub>1</sub>) , (b<sub>1</sub>, c<sub>3</sub>))

→

↗

در نتیجه : (A , B) ⇒ C

**نکته :**

اگر x صفت ساده باشد و وابستگی تابعی بین x و y برقرار باشد (x → y) ،  
وابستگی تابعی کامل (x ⇒ y) همواره برقرار است.

**4-5-7- شرح آنومالی‌ها :**

همان‌طور که گفتیم آنومالی سه وجه دارد. اینک با در نظر گرفتن رابطه‌ی مثال 3-1 آنومالی را توضیح می‌دهیم.

**(1) درج :**

در آنومالی ناشی از درج ، تمام یا بخشی از کلید اصلی تعریف نشده و نامعین است. طبق یکی از قواعد عام بانک اطلاعاتی ، مقدار کلید اصلی باید کاملاً مشخص باشد.

فرض کنید می‌خواهیم رکورد <P<sub>10</sub>, 1000, c<sub>4</sub>, 10> را درج کنیم. این درج امکان‌پذیر نیست. زیرا مشخص نیست ناشر چه کتابی را منتشر کرده است.

**(2) حذف :**

در آنومالی نوع دوم پس از انجام يك عمل ، عوارض نامطلوب داریم. فرض کنید می‌خواهیم رکورد  $\langle P_2, b_4, 3000 \rangle$  را حذف کنیم. این حذف اگرچه شدنی است اما رتبه‌ی صنفی ناشران شهر  $c_2$  (مقدار 60) ناخواسته حذف می‌شود. (دقت کنید فعلا تنها يك ناشر ساکن شهر  $c_2$  است).

3) به‌هنگام‌سازی :

این نوع آنومالی ، موجب فزون‌کاری می‌شود. در فزون‌کاری به ازای يك عمل مبنایی چندین عملیات صورت می‌گیرد. فرض کنید قرار است رتبه‌ی صنفی ناشران ساکن شهر  $c_3$  از 50 به 70 تغییر کند. این عمل موجب به‌هنگام‌سازی منتشرشونده<sup>1</sup> (فزون‌کاری) در سیستم می‌شود ، یعنی در دو سطر جدول باید تغییر صورت گیرد.

همان‌طور که مشاهده می‌شود ، رابطه‌ی FIRST (مثال 1-7) دارای آنومالی است. در يك بیان غیردقیق ، علت آنومالی‌ها ، "اختلاط اطلاعاتی" است. به این معنی که اطلاعات نشر و اطلاعات پایه‌ای ناشر در يك رابطه (جدول) با هم جمع شده است. برای ایجاد يك طراحی مناسب از رابطه‌ها باید آن‌ها را از لحاظ سطوح نرمال بررسی کنیم و در صورت نرمال نبودن در آن سطح ، آن رابطه را با "تجزیه" اصلاح نماییم. در این صورت از بروز آنومالی جلوگیری می‌شود. اینک مناسب است سطوح نرمال رابطه را با بیان دقیق‌تر مورد بررسی قرار دهیم.

### 5-5-7- سطوح نرمال :

روابط از سطح غیرنرمال تا سطح نرمال 5 قابل تبیین و بررسی است. سطوح نرمال عبارتند از :

- 1) سطح نرمال اول 1NF
- 2) سطح نرمال دوم 2NF
- 3) سطح نرمال سوم 3NF
- 4) سطح نرمال  $BCNF^2$
- 5) سطح نرمال چهارم 4NF
- 6) سطح نرمال پنجم 5NF

<sup>1</sup> - Propagated Update

<sup>2</sup> - به احترام نام Boyce و Codd نامیده شده است. Boyce-Codd Normal Form (BCNF)

در این کتاب تا سطح نرمال 3NF را بررسی می‌کنیم. چرا که رابطه در این سطح، از بسیاری آنومالی‌ها جلوگیری می‌کند.

### 1-5-7- سطح نرمال اول :

رابطه‌ی First در مثال 1-7 را در نظر بگیرید. تمام فیلدها به نحوی هستند که مقادیر آنها به صورت منطقی قابل تقسیم نیست. به هر یک از این فیلدها، یک فیلد تک مقداری گوئیم. اگر تمام فیلدها تک مقداری باشد، رابطه در سطح نرمال اول قرار دارد.

رابطه‌ی R در سطح نرمال اول (1NF) است، اگر تمام فیلدهای آن در هر سطر جدول، تک مقداری باشد.

### مثال 5-7 :

رابطه‌ی کارمند را در نظر بگیرید، آیا این رابطه در سطح INF است؟

نام	شماره شناسنامه	میزان تحصیلات	کد پرسنلی
علی علوی	243	دیپلم	110
ستایش یمقانی	2719	کارشناسی	140
رضا قاسمیان	593	کاردانی	130
اسماعیل میرزایی	909	کارشناسی	170
علیرضا مسچی	714	کارشناسی	190

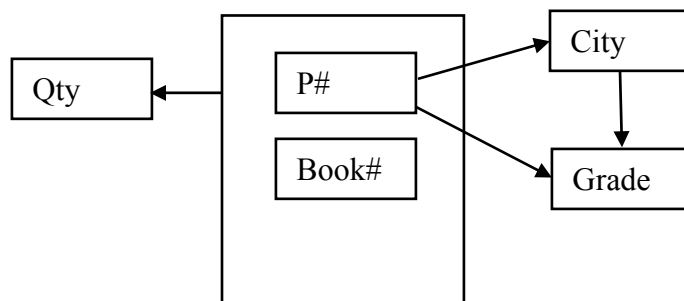
پاسخ : فیلد نام ، قابل تجزیه به دو فیلد نام و نام خانوادگی است . اگر بخواهیم نام خانوادگی کارمندان را از رابطه به دست آوریم ، فیلد نام را تجزیه کرده ایم ؛ بنابراین رابطه غیر نرمال است و در سطح INF قرار ندارد . شکل نرمال INF آن به صورت زیر است :

نام	نام خانوادگی	شماره شناسنامه	میزان تحصیلات	کد پرسنلی
علی	علوی	243	دیپلم	110
ستایش	یمقانی	2719	کارشناسی	140
رضا	قاسمیان	593	کاردانی	130
اسماعیل	میرزایی	909	کارشناسی	170
علیرضا	مسچی	714	کارشناسی	190

### 2-5-7- سطح نرمال دوم :

در قسمت‌های قبل با تعریف وابستگی تابعی کامل آشنا شدیم .

رابطه وقتی در سطح INF باقی می‌ماند که وابستگی تابعی کامل بین حداقل یک فیلد با کلید اصلی نقض شود . به عنوان مثال نمودار FD برای رابطه‌ی FIRST را در نظر بگیرید :



کلید اصلی در این رابطه ، صفت مرکب (P# , Book#) است. از آن جایی که تمام فیلدها با کلید FD دارند ، داریم :

(P# , Book#) → City

برای برقراری وابستگی تابعی کامل باید داشته باشیم :

P# → City

Book# → City

اما همانطور که در نمودار FD قابل مشاهده است ، داریم :

P# → City

در نتیجه FD کامل بین City و (P# , Book#) نقض شده است.

اینک می‌توانیم دلیل آنومالی‌های رابطه‌ی FIRST را بهتر دریابیم. نقض FD کامل.

رابطه‌ی R در سطح نرمال دوم (2NF) است ، اگر  
**اولا :** 1NF باشد.  
**ثانیا :** تمام صفات غیرکلید با صفت کلید ویژگی وابستگی تابعی کامل را نقض نکنند.

برای رفع آنومالی و افزایش سطح نرمال رابطه ، می‌باید رابطه‌ی FIRST را تجزیه نماییم . برای اینکار FIRST را به صورت زیر تجزیه می‌کنیم :

SECOND (P# , City , Grade)

PB (P# , Book# , Qty)

SECOND		
P#	City	Grade
P <sub>1</sub>	c <sub>1</sub>	40
P <sub>2</sub>	c <sub>2</sub>	60
P <sub>3</sub>	c <sub>1</sub>	40
P <sub>4</sub>	c <sub>3</sub>	50

PB		
P#	Book#	Qty
P <sub>1</sub>	b <sub>1</sub>	1000
P <sub>2</sub>	b <sub>4</sub>	3000
P <sub>3</sub>	b <sub>3</sub>	2000
P <sub>1</sub>	b <sub>2</sub>	3000
P <sub>4</sub>	b <sub>3</sub>	4000
P <sub>4</sub>	b <sub>5</sub>	3500



### ملاك تجزيه‌ي رابطه :

تجزیه‌ي رابطه R به رابطه‌هاي  $R_1$  و  $R_2$  می‌باید به‌نحوي باشد که پیوند<sup>1</sup> دو رابطه‌ي  $R_1$  و  $R_2$  رابطه‌ي R را ایجاد کند و تاپلي (رکورد یا سطري) کم و زیاد نشود. از طرف دیگر تجزیه‌ي R می‌باید ، وابستگی‌هاي تابعي را حفظ کند.

#### تمرین 2-7 :

آیا معیار‌هاي تجزیه‌ي مناسب در تجزیه‌ي FIRST لحاظ شده است؟

رابطه‌ي SECOND ، در سطح دوم نرمال است. زیرا اولاً: INF است و ثانياً : وابستگی تابعي کامل نقض نشده است. اما این رابطه نیز دارای آنومالي می‌باشد.

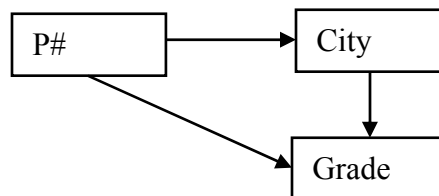
#### تمرین 3-7 :

آنومالي‌هاي SECOND را ذکر کنید.

### 3-5-7- سطح نرمال سوم :

رابطه‌ي SECOND را در نظر می‌گیریم. نمودار FD آن به صورت زیر است :

1. P# → City
2. City → Grade
3. P# → Grade



همان‌طور که در مثال .... دیدیم ، از دو وابستگی تابعي اول و دوم ، می‌توان وابستگی تابعي سوم را نتیجه گرفت ، یعنی Grade وابستگی تابعي با واسطه با P# از طریق City دارد.

<sup>1</sup> - Join

علت بروز آنومالی‌های SECOND نیز همین عامل یعنی وابستگی تابعی با واسطه است.

رابطه‌ی R در سطح نرمال سوم (3NF) است. اگر :  
**اولا : 2NF باشد.**  
**ثانیا : هر صفت غیر کلید با کلید اصلی ، وابستگی تابعی با واسطه نداشته باشد.**

برای رفع آنومالی‌های آن ، می‌باید این رابطه تجزیه شود. بنابراین SECOND را به رابطه‌های PC و CS تجزیه می‌کنیم :

PC (P# , City)

CS (City , Grade)

PS	
P#	City
P <sub>1</sub>	c <sub>1</sub>
P <sub>2</sub>	c <sub>2</sub>
P <sub>3</sub>	c <sub>1</sub>
P <sub>4</sub>	c <sub>3</sub>

CS	
City	Grade
c <sub>1</sub>	40
c <sub>2</sub>	60
c <sub>3</sub>	50

با حذف آنومالی‌های SECOND ، روابط جدید نه تنها در سطح 3NF قرار می‌گیرند بلکه در سطح بالاتر یعنی BCNF نیز قرار می‌گیرند که ما به بررسی آن نمی‌پردازیم.

**تمرین 4-7 :**  
 1. آیا تجزیه‌ی انجام‌شده ، ضوابط یک تجزیه‌ی مناسب را دارا می‌باشد؟  
 2. حالت‌های دیگر تجزیه‌ی SECOND را در نظر بگیرید. آیا این تجزیه‌ها مناسب هستند؟

اگرچه نرمال‌سازی ، آنومالی‌ها را از بین می‌برد اما در پرسوجوها ، به علت نیاز به پیوند رابطه‌ها (جدول‌ها) سرعت انجام پرسوجو کاهش می‌یابد.

### **کنجکاو ي :**

پس از اعمال سطوح نرمال و تجزیه‌ی جدول‌ها ، سرعت بازیابی اطلاعات موردنیاز کاربران بررسی می‌شود . در صورت عدم کارایی پرس‌وجوهای موردنظر ، و بنا بر ملاحظاتی ممکن است جدول‌ها مجدداً ترکیب گردند. چه ملاحظاتی در این مورد مطرح است؟

## خلاصه ي فصل

پس از طراحی مفهومی بانک اطلاعاتی و رسم نمودار ER ، جداول ، فیلدها و کلید اصلی آنها تعیین میشوند . در برخی موارد جدولهای ایجاد شده دارای آنومالی هستند . یعنی برخی عملیات حذف ، اضافه ، ویرایش روی آنها غیرممکن میشود ، یا در صورت انجام اطلاعات مهم از دست میرود و یا با دشواری همراه است .

دلایل ایجاد آنومالی عدم نرمال بودن جدول است . ما پنج سطح نرمال داریم که سه سطح آن را بررسی کرده ایم . بیشتر جدولهای نرمال در سطح سوم ، در سطح بالاتری نیز قرار میگیرند که BCNF نامیده میشود . پس از تعیین جدولها ، هر جدول از نظر سطح نرمال بررسی شده و در صورت لزوم تجزیه میشود . هر سطح نرمال دارای شرایطی است .

هر جدول غیرنرمال برای نرمال شدن باید تجزیه شود . عمل تجزیه ملاکهایی دارد که به برخی از آنها در فصل اشاره کرده ایم . تجزیه شدن جدول به جدولهای جدید که حافظ اطلاعات جدول قبلی باشد ، آنومالیهای موجود را از بین میبرد ولی در مواردی سرعت اجرا و بازیافت اطلاعات در زمان اجرای پرسوجوها را افزایش میدهد . ترکیب جدول به جدول قبلی بنا به ملاحظات صورت قابل انجام است .

## خود آزمایي

1. آنومالی به چه معنی است؟
2. وابستگی تابعی و وابستگی تابعی کامل را با ذکر مثال توضیح دهید .
3. سطح نرمال اول ، دوم و سوم چه ویژگیهایی دارند؟
4. محاسن و معایب نرمال سازی چیست؟

## 8- فصل هشتم : مطالعه موردی

در فصل‌های قبل با امکانات محیط بانک اطلاعاتی آشنا شدیم. حال می‌خواهیم با یک مثال از این امکانات برای ایجاد یک سیستم بانک اطلاعاتی استفاده کنیم. مهدکودکی علاقه‌مند به ایجاد یک پایگاه داده برای مدیریت اطلاعات کودکان تحت حمایت خود است. والدین، نگهداری روزانه‌ی کودک خود را به این موسسه محول کرده‌اند. روزانه غذاهای متنوعی توزیع می‌گردد، با این حال باید بدانیم آیا کودکی وجود دارد که نتواند از این تنوع غذایی استفاده کند و در این صورت، باید تدارک این‌گونه استثنایات نیز دیده شود. بنابراین اطلاعات نیاز روزانه‌ی غذایی و رژیم‌ی کودکان باید ذخیره گردد، مشخصات فردی کودکان و والدین آن‌ها باید نگهداری شود و همچنین برای بهبود فعالیت مهدکودک، می‌توانیم زمان واکسیناسیون، پایش رشد (کنترل قد و وزن) را نیز نگهداری و برآورد کنیم و به والدین اطلاع دهیم و یادآوری کنیم.

### 8-1- اطلاعات کودکان Children :

زمانی که یک کودک برای اولین بار به مهدکودک آورده می‌شود، یک مشخصه منحصر به فرد می‌گیرد. هر کودک در یک تخت مخصوص به خود و در اتاقی خاص نگهداری می‌شود. کودکان نوپا می‌توانند از انواع غذاها استفاده کنند ولی نوزادان به برنامه‌ی غذایی خاص نیاز دارند. همچنین برخی کودکان ممکن است به غذاهای خاصی حساسیت داشته باشند و نتوانند برخی انواع مواد غذایی مثل توت‌فرنگی، تخم‌مرغ، گوجه‌فرنگی و... را مصرف کنند. همچنین اطلاعات واکسیناسیون و آخرین تاریخ پایش رشد کودکان نیز باید نگهداری شود.

### 8-2- اطلاعات والدین Parents :

هر یک از والدین می‌تواند مراقبت یک یا چند کودک خود را به مهدکودک بسپارد. برخی والدین علاقه‌مند هستند کودک خود را در اتاق غیرخصوصی و مشترک نگهداری کنند تا با کودکان دیگر مرادوده داشته باشند و همچنین هزینه‌ی آن ارزان‌تر باشد.

### 8-3- تحلیل

مهارت و دانشی که در زمان تحلیل موردنیاز است دانش کامپیوتری فرد طراح نیست، بلکه دانش درک مساله

است. توجه ما در این کتاب ، تحلیل و شناخت سیستم نیست ، بنابراین فرض می‌کنیم تحلیل قبلا انجام شده است تا بتوانیم کاربرد Access را با هم بیاموزیم.

1. **موجودیت‌ها Entity** : این اصطلاح برای هر چیزی به کار می‌رود که باید اطلاعاتی درمورد آن ذخیره کنیم. موجودیت‌های دنیای واقع به جدول‌های بانک تبدیل می‌شوند. در مثال ، دو جدول Parents و Children قابل شناسایی است.

2. **مشخصه‌ها Attribute** : جزییات بیشتر یک موجودیت را در بر دارد.

- مشخصه‌ی Parents عبارت است از : اطلاعات تماس والدین ، شناسه‌ی منحصر به فرد که به هر یک از اولیاء تخصیص می‌دهیم ، نام و نام‌خانوادگی وی ، همچنین اطلاعات آدرس، شهر، کشور، کدپستی و شماره تلفن وی که برای ما اهمیت دارد. به خاطر بسپارید که مشخصات کودک جزو مشخصه‌ی والدین نیست و به صورت مستقل به عنوان یک موجودیت مطرح است.

- مشخصه‌ی Children عبارت است از : شناسه کودک، نام، سن، جنسیت، تاریخ واکسیناسیون و نیازهای خاص رژیمی و ملاحظات تغذیه‌ای وی.

3. **ارتباط بین موجودیت‌ها Relationship** :

برای تعیین ارتباط Relation بین این دو موجودیت ، شناسه‌ی جدول Parents را در جدول Children اضافه می‌کنیم

#### نکته :

1) در زمان طراحی بانک ، بهتر است شناسه‌های منحصر به فرد را تعیین و یا ایجاد کنیم و از شناسه‌های ترکیبی استفاده نکنیم. مثلا می‌توانیم از نام و آدرس به صورت مشترک برای کلید استفاده کنیم ولی این کار ایجاد ارتباط بین دو جدول را مشکل می‌سازد.

2) شاید بتوان از شماره‌ی کارت ملی و یا شماره حساب سیبا که منحصر به فرد است استفاده کرد.

آن‌ها نیز به وسیله‌ی ParentsId تعریف شده است. ( Id نمایانگر عبارت Identifier به معنی شناسه است).

### یادآوری :

Table ، روش ساختیافته‌ی نمایش داده‌ها است. فیلدها در Access به صورت عنوان ستون‌ها نمایش داده می‌شوند. جدول ، مقادیر داده‌ها را برای نام ، نام خانوادگی و ... نگهداری می‌کند. این مقادیر به صورت ردیف نمایش می‌یابند که در اصطلاح بانک‌داده ، رکورد Record نامیده می‌شوند.

### جدول 1 : فیلدهای جدول Parents

ParentsId	FirstName	LastName	Address	City	Country	PostCode	PhoneNumber
شناسه	نام	نام خانوادگی	آدرس	نام شهر	نام کشور	کد پستی	شماره تلفن تماس

شرح فیلد:

### جدول 2 : فیلدهای جدول Children

ChildId	ParentsId	FName	Age	Sex	VacDate	DietNotes
شناسه	شناسه والدین	نام	سن	جنسیت	تاریخ واکسیناسیون	ملاحظات تغذیه‌ای

شرح فیلد:

## 8-4- استاندارد نام‌گذاری اشیاء بانک

اعمال یک استاندارد برای نام‌گذاری مناسب فایل‌ها و اشیاء بانک ، کار را قابل اعتماد می‌سازد ، خصوصاً زمانی که فایل به اشتراک گذارده می‌شود. مثلاً تمام اشیاء ، توصیف‌کننده‌ای داشته باشند که در شروع نام قرار داده شود. استاندارد Access را می‌توانید در آدرس زیر و راهنمای Leszynski/Reddick پیدا کنید. مثلاً این‌که از کاراکتر فاصله در اسامی استفاده نکنید ، ارتباط بین جدول‌ها با استفاده از اسامی مشابه فیلدهای متناظر ایجاد گردد و ....

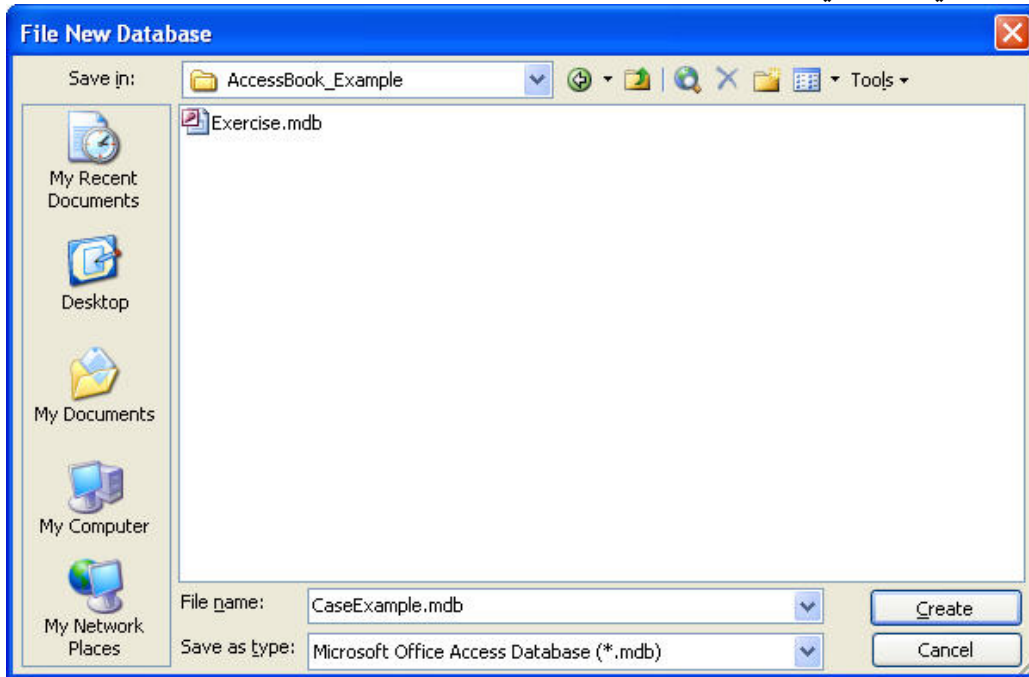
[http://www.microsoft-accesssolutions.co.uk/naming\\_conv.htm](http://www.microsoft-accesssolutions.co.uk/naming_conv.htm)

## 8-5- ایجاد پایگاه داده

فایل پایگاه داده‌ای که در این قسمت ایجاد می‌کنید ، در واقع ظرف نگهدارنده‌ی (Container) کلیدی اشیاء بانک خواهد بود مثل: جدول‌ها، فرم‌ها و ....

- برنامه‌ی Microsoft Access را اجرا کنید.
- Create a new file را از پانل سمت راست انتخاب کنید. روی Blank Database کلیک کنید.

- در پنجره‌ی محاوره‌ای File New Database محل موردنظر و نام بانک را CaseExample.mdb قرار داده و Create را کلیک کنید.



حال که فایل نگهدارنده ایجاد شد، می‌توان اشیاء (جدول‌ها، فرم‌ها، پرس‌وجوها و گزارش‌ها) را به آن اضافه کرد.

## 8-6- ایجاد جدول‌ها

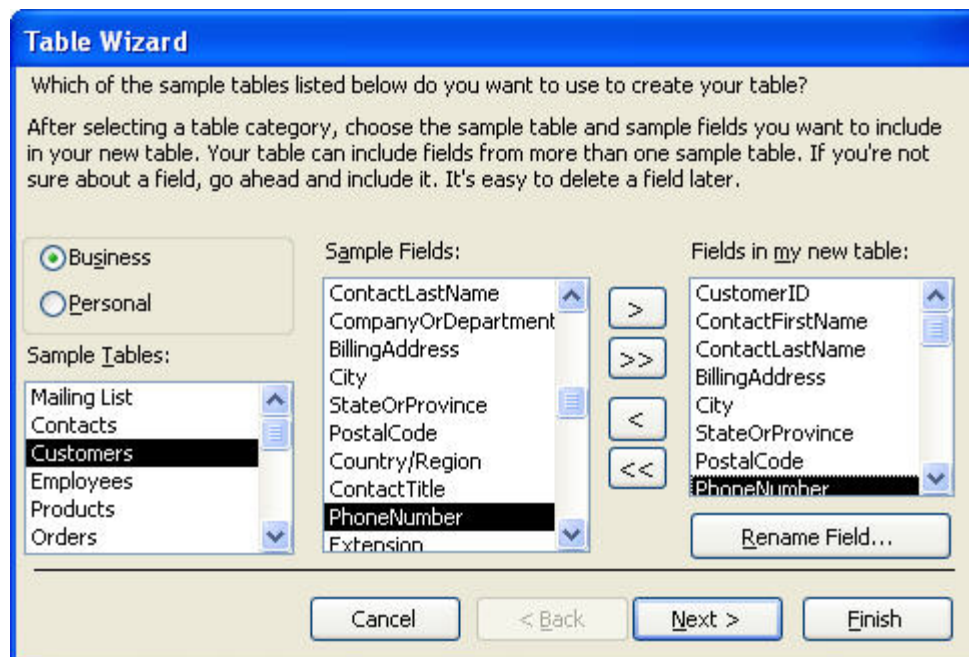
- ایجاد جدول به وسیله‌ی Table Wizard چون اشیاء فرم‌ها، گزارش‌ها و پرس‌وجوها بر مبنای جدول‌ها ایجاد می‌شوند. ابتدا باید دو جدول را ایجاد کنیم.

### یادآوری:

نکات موردنیاز در زمان استفاده از Wizard:  
 از لیست جدول‌ها موجود، جدولی را انتخاب کنید که بیشترین شباهت را به جدول طراحی‌شده داشته باشد. مثلاً چون Parents شبیه Customer است، مشخصه‌های مشابه را دربرخواهد داشت.  
 اسامی نباید فاصله داشته باشند. چون برخی کامپیوترها آن را پشتیبانی نخواهند کرد.  
 تمام مشخصه‌ها (فیلد) در لیست نمونه Sample Field ظاهر می‌شوند، اگرچه برخی نیاز به اصلاح دارند. از دکمه > برای افزودن استفاده کنید و در صورت اشتباه از < برای حذف مورد استفاده کنید.



- Table Wizard را از لیست New Table انتخاب کنید و OK کنید.
- روی Customers از لیست Sample Table کلیک کنید.
- از Sample fields ، فیلد CustomerId را انتخاب و دکمه > را بزنید تا به لیست Fields in My New Table اضافه گردد. برای باقی فیلدها مثل Firstname ، Lastname ، Address ، City ، Country ، PostCode ، PhoneNumber نیز همین‌گونه عمل کنید و فیلدهای 'CustomerId' را به جای 'ParentsId' ، 'CustomerFirstName' را به جای 'FirstName' ، 'CustomerLastName' را به جای 'LastName' ، 'BillingAddress' را به جای 'Address' و 'StateOrProvince' را به جای 'County' اضافه کنید.
- روی CustomerId در قسمت Fields in my new table کلیک کرده و دکمه‌ی Rename Field را کلیک کنید و آن را به ParentsId تغییر دهید. به همین صورت برای سایر فیلدها عمل کنید.
- روی دکمه Next کلیک کنید و نام جدول موردتقاضا را TableParents بگذارید.
- گزینه‌ی No, I'll set the primary key را انتخاب کنید و دکمه Next را کلیک کنید .
- گزینه‌ی Modify Table Design را انتخاب کنید چراکه می‌خواهیم تغییرات مختصری در طراحی جدول اعمال کنیم.
- روی دکمه Finish کلیک کنید.



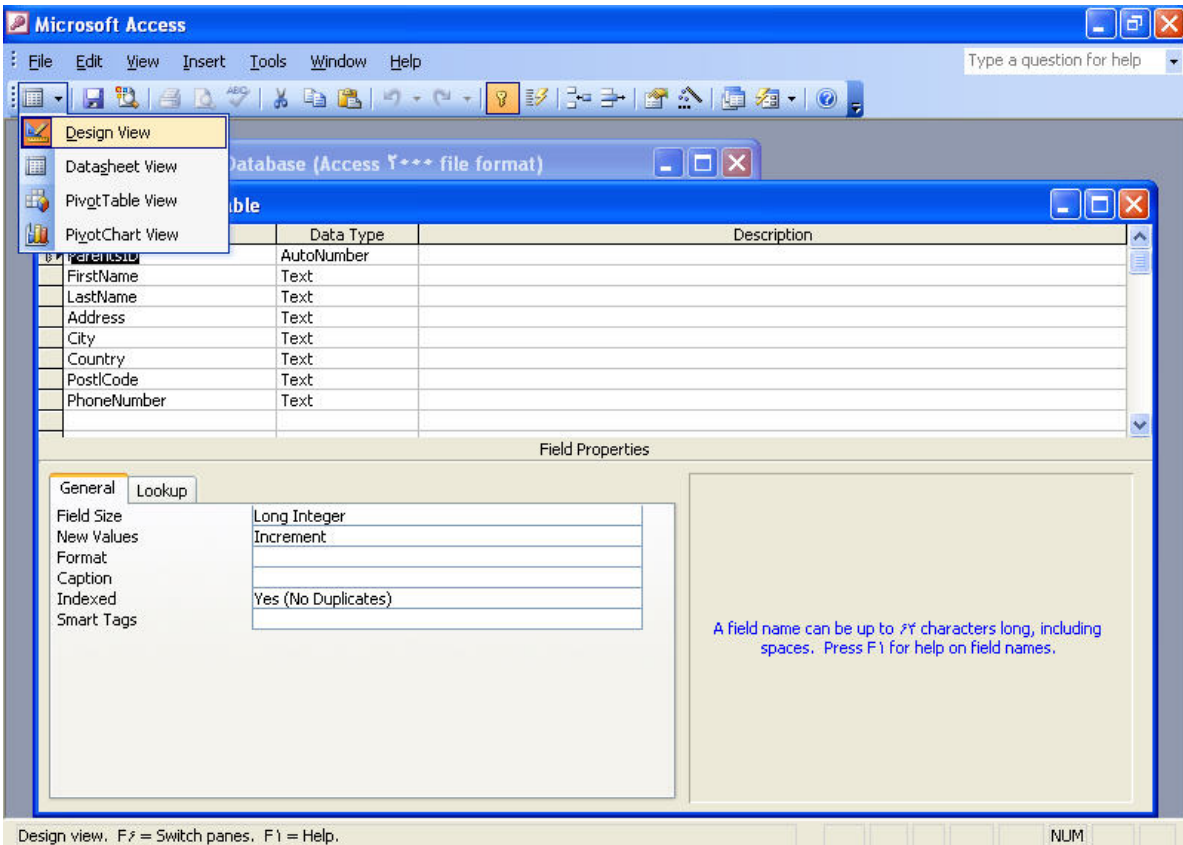
### 8-7- اصلاح طراحي جدولها

با انتخاب Modify Table Design در ايجاد جدول به كمك Wizard ، جدول در نماي Design باز مي‌شود.

#### نکته :

اگر در مراحل بعد ، بخواهيد طراحي جدول را اصلاح كنيد ، مي‌توانيد از نماي Design جدول استفاده كنيد. براي اين منظور پنجره‌ي بانك را انتخاب ، جدول موردنظر را انتخاب کرده و روي دكمه‌ي Design View كليك نماييد. اگر جدول باز است و داده‌هاي آن را مشاهده مي‌كنيد ، مي‌توانيد از دكمه‌هاي نوار ابزار براي اين منظور استفاده كنيد. با كمك دكمه‌ي table design view مي‌توانيد بين نماها جابه‌جا شويد. در صورتي كه دكمه‌ي مربوطه را در نوار ابزار مشاهده نمي‌كنيد ، از منوي View ، انتخاب Toolbars ، مطمئن شويد كه نوار ابزار موردنظر نشانه‌گذاري شده است.






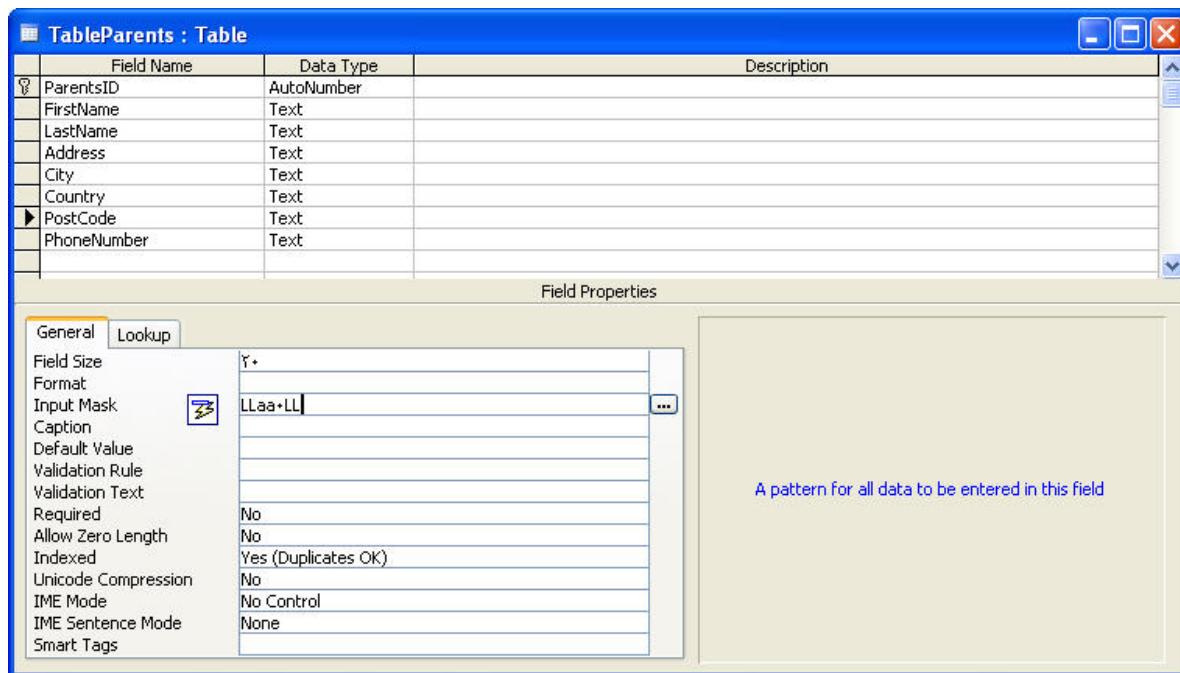
در نمای Design ، می‌توانید نام ، نوع فیلد ، توضیحات و دیگر خصوصیات فیلد را تغییر دهید.

توجه شود که ParentsId از نوع داده AutoNumber است ، یعنی هر زمان داده‌ای (رکورد هر یک از والدین جدید) اضافه شود، Access به‌طور خودکار ، عددی منحصر به فرد به عنوان ParentsId ایجاد می‌کند و بدین ترتیب هر رکورد ، کلید منحصر به فردی خواهد داشت.

همچنین در Access می‌توان مشخصه Input Mask فیلد را تعریف کرد. این مشخصه کمک می‌کند کاربر داده غلط وارد نکند. مثلاً جا افتادن رقمی از شماره تلفن و ...

- کلیک روی PostCode و انتخاب آن.
- کلیک در بخش Field Properties و انتخاب ردیف Input Mask. (باید زبانه general انتخاب شده باشد)
- روی دکمه  سمت راست آن کلیک کنید. پیغام ذخیره‌ی جدول را تایید کنید.

- زمانی که Mask Wizard شروع شد ، PostCode را انتخاب نمایید.
- در قسمت Try It ، عبارت LLaa0LL را تایپ کنید تا عبارت باید شامل دو حرف سپس دو رقم/حرف و ارقام صفر تا 9 و سپس دو حرف دیگر باشد.



- روی Finish کلیک کنید.
- از منوی File ، Save را انتخاب کنید.
- در صورتی که کار با طراحی جدول تمام شده باشد ، آن را Close کنید. ولی ما به حالت Datasheet می‌رویم تا داده‌ها و رکوردهایی را در جدول اضافه کنیم.

### عبارتهای مجاز در Input Mask :

کاراکتر	توضیحات
0	رقم ( صفر تا 9 و علامت [+] یا [-] مجاز نیست).
9	رقم یا خالی (ورود اطلاع ضروری نیست و علامت مثبت یا منفی مجاز نیست).
#	رقم یا فضای خالی (ورود اطلاع ضروری نیست . در زمان ویرایش علامت Space به صورت فضای خالی نمایش داده می‌شود ولی در زمان ذخیره‌سازی حذف می‌گردد. علامت مثبت و منفی مجاز است).
L	حروف ( فارسی از الف تا ی و انگلیسی از A تا Z . ورود اطلاع ضروری است).
?	حروف ( فارسی از الف تا ی و انگلیسی از A تا Z . ورود اطلاع ضروری نیست).

<b>A</b>	حروف یا رقم (ورود اطلاع ضروري است) .
<b>a</b>	حروف یا رقم (ورود اطلاع ضروري نیست) .
<b>&amp;</b>	هر کاراکتری و یا فضای خالی (ورود اطلاع ضروري است) .
<b>C</b>	هر کاراکتری و یا فضای خالی (ورود اطلاع ضروري نیست) .
./-:;>	جداساز ( برای محل رقم هزارگان ، تاریخ ، زمان و استفاده از کاراکترهای جداساز استاندارد در محیط ویندوز) .
<	باعث می‌شود تمام کاراکترها به حروف کوچک تبدیل شوند .
>	باعث می‌شود تمام کاراکترها به حروف بزرگ تبدیل شوند .
!	باعث می‌شود ، عبارت از راست به چپ نمایش یابد . کاراکترهایی که در این فیلد تایپ می‌شوند ، فیلد را از چپ به راست پر می‌کنند . این علامت در هر جایی از عبارت می‌تواند قرار گیرد .
\	باعث می‌شود کاراکترها به صورت کاراکتر ثابت نمایش داده شوند ( مثلا \A به صورت A نمایش داده می‌شود)

## 8-8- نمای Datasheet جدول

پس از تغییر طراحی جدول ، نمای Design را ترک کرده ، به نمای Datasheet می‌رویم تا بتوانیم در این نما ، داده‌هایی را به جدول اضافه کنیم .

ParentsID	FirstName	LastName	Address	City	Country	PostCode	Phone Number
(AutoNumber)							

### ورود داده به جدول :

مطابق شکل زیر ، داده‌ها را به جدول اضافه کنید. با کمک کلیدهای TAB روی فیلدها و از رکوردی به دیگری جابه‌جا شوید و یا بالعکس ، از Shift+TAB استفاده کنید. پس از اتمام ورود داده‌ها ، از منوی File ، Close را انتخاب کنید.

ParentsID	FirstName	LastName	Address	City	Country	PostCode	Phone Number
۱	هما	سحرخیز	خیابان ولیعصر ، خیابان ظفر	تهران	ایران	KA۶۵۸GH	(۰۹۸)۲۱-۲۲۰۵۶۷۸
۲	سعید	میرزایی	خیابان استاد نجات‌اللّهی	تهران	ایران	MB۹۷۸SF	(۰۹۸)۲۱-۲۹۵۲۶۷۸
۳	فرح	نرحمی	خیابان دکتر فلسفی	تهران	ایران	MB۹۷۸SF	(۰۹۸)۲۱-۸۹۶۵۴۳۴
۴	منیره	میرزایی	بزرگراه کردستان	تهران	ایران		
۵	جمیله	پوررضایی		تهران	ایران		
۶	رضا	حکمی	انواران شهید ستاری ، خیابان مطهری	تهران	ایران		(۰۹۸)۲۱-۲۲۶۵۵۳۴

## 8-9- استفاده از فرم‌ها

### ایجاد فرم جدید ورود اطلاعات با کمک Form Wizard

با کمک فرم می‌توان اطلاعات وارد شده را کنترل کرد و از ثبت اطلاعات غلط جلوگیری کرد. مثلاً زمان ورود اطلاعات کودکان کودک ، می‌توانیم از جدول والدین استفاده و لیستی از آن‌ها ارائه کنیم. بدین ترتیب شناسه‌ی والدین را به‌طور صحیح به‌دست آورده و در جدول Children به‌همراه

سایر اطلاعات کودک ذخیره نماییم. ابتدا فرمی بدون هیچ گونه کنترلی ایجاد می‌نماییم.

### ایجاد فرم :

- روی دکمه Form از پنجره پایگاه داده کلیک کنید.
- روی دکمه New در بخش Form پنجره پایگاه داده کلیک کنید.
- در پنجره محاوره‌ای باز شده ، Form Wizard را انتخاب نمایید و در قسمت Choose the table or query ، TableParents را انتخاب کنید و OK نمایید.
- این برنامه‌ی کمکی ، درخصوص فیلدهای موردنظر سوال می‌کند. با دکمه >> می‌توانید تمام فیلدها را انتخاب نمایید. فیلدهای موجود به قسمت Selected Fields منتقل می‌شوند.
- Next را کلیک نمایید. برنامه در مورد الگوی ( Lay out) موردنظر سوال می‌کند. الگوی Columnar را انتخاب نمایید.
- Next را کلیک کنید. برنامه در مورد قالب (Style) فرم سوال می‌کند. همان الگوی Standard را برگزینید.
- Next را کلیک نمایید. برنامه در مورد عنوان (Title) فرم سوال می‌کند. عبارت "Form Parents" را تایپ نمایید.
- روی Finish کلیک نمایید. فرم در نمای View باز می‌شود. در این نما می‌توانید داده‌ها را وارد و در جدول ذخیره نمایید ، اطلاعات را ویرایش و یا حذف کنید.

### ورود اطلاعات به وسیله‌ی فرم :

- دکمه‌ی (+) Ctrl+Shift را کلیک کنید تا در یک رکورد جدید قرار گیرید.
- اطلاعات نام و آدرس بیشتری را مطابق شکل زیر وارد نمایید.

۷	محمدرضا	بمقانی	خیابان سنارخان ، تهران وچلا	تهران	ایران	۴۳۵۰۸۷۶۵
۸	آبت	کلشانی		کرج	ایران	(۰۲۶۱)-۷۶۵۶۴۲۱۱
۹	مینا	راهنورد		تهران	ایران	
*	oNumber)					

**FormParents**

ParentsID: ۷

FirstName: محمدرضا

LastName: يمقاني

Address: خيابان ستارخان ، تهران ويدا

City: تهران

Country: ايران

PostCode:

Phone Number: ۴۳۵۰۸۷۶۵

Record: ۷ of ۷

- با دکمه‌ی Close فرم را ببندید.
- جدول TableParents را باز کنید و چک کنید آیا اطلاعات اضافه شده است.

**TableParents : Table**

ParentsID	FirstName	LastName	Address	City	Country	PostCode	Phone Number
۱	هما	سحرخیز	خیابان ولیعصر ، خیابان ظفر	تهران	ایران	KA۶۵۸GH	(۰۹۸)۲۱-۲۲۰۵۶۷۸
۲	سعید	میرزایی	خیابان اسلا نجات الاهی	تهران	ایران	MB۹۷۸SF	(۰۹۸)۲۱-۲۹۵۲۶۷۸
۳	فرح	ترجمی	خیابان دکتر فلسفی	تهران	ایران	MB۹۷۸SF	(۰۹۸)۲۱-۸۹۶۵۴۴۸
۴	منیژه	میرزایی	بزرگراه کردستان	تهران	ایران		
۵	جمیله	پوررضایی		تهران	ایران		
۶	رضا	حکمی	انوبان شهید سناری ، خیابان مظفری	تهران	ایران		(۰۹۸)۲۱-۲۲۶۵۵۳۴
۷	محمدرضا	یمقانی	خیابان ستارخان ، تهران ویدا	تهران	ایران		۴۳۵۰۸۷۶۵
۸	آبت	کاشانی		کرج	ایران		(۰۲۶۱)-۷۶۵۶۴۲۱۱
۹	مینا	راهنورد		تهران	ایران		
* aNumber)							

Record: ۱ of ۹

### یادآوری :

به خاطر داشته باشید که فرم ، یکی از راه‌های ورود داده به جدول است و تمام اطلاعات در جدول ذخیره می‌شود. زمانی که داده‌ای در فرم تایپ می‌شود ، اطلاعات آن در جدول ذخیره می‌شود و نه در فرم مربوطه.



## 8-10 - روابط Relationship

انواع رابطه‌ها : سه نوع اصلی ارتباط وجود دارد :

- **يك-به-يك (One-to-One) :**  
يك رکورد در يك جدول با فقط يك رکورد از جدول ديگر مرتبط است.
- **يك-به-چند (One-to-Many) :**  
يك رکورد در يك جدول مي‌تواند به رکوردهاي زيادي از جدول ديگر مرتبط باشد.
- **چند-به-چند (Many-to-Many) :**  
يك رکورد در يك جدول مي‌تواند به يك يا چند رکورد جدول ديگر مرتبط باشد و يك رکورد آن جدول مي‌تواند به يك يا چند رکورد جدول اولي مرتبط گردد.

ارتباط يك-به-يك با استفاده از يك جدول به‌جاي دو جدول حل مي‌شود. ارتباط چند-به-چند معمولا به جدول سومي نياز دارد كه به عنوان رابط بين دو موجوديت به‌كار رود. براي توضيحات بيشتر در اين خصوص به كتابهاي پايگاه داده و بخش مفاهيم اوليه مراجعه كنيد.

### افزودن جدول ديگر :

اکنون به جدول ديگري نياز داريم كه موجوديت Children را ايجاد كند. چون جدولهاي نمونه Table Wizard مناسب به‌نظر نمي‌رسند ، بايد خودمان بدون كمك Wizard جدول را ايجاد و طراحي كنيم. بايد نام‌فيلد ، نوع‌فيلد و ساير خصوصيات را تعريف كنيم .  
نام فيلدهاي جدول عبارتند از :

ChildId, ParentsId, Name, Age, Sex, VacDate, DietNotes

از توصيف فارسي نام فيلدها به عنوان (Caption) فيلد استفاده مي‌كنيم. در اينجا انواع داده و خصوصيات فيلدها را بررسي مي‌كنيم :

- ChildId از نوع AutoNumber است. چراكه Access خود به‌طور اتوماتيك به هر رکورد جديد مقدار منحصربه‌فردي را تخصيص دهد. اين فيلد به عنوان كليد اصلي (Primary key) شناخته مي‌شود.

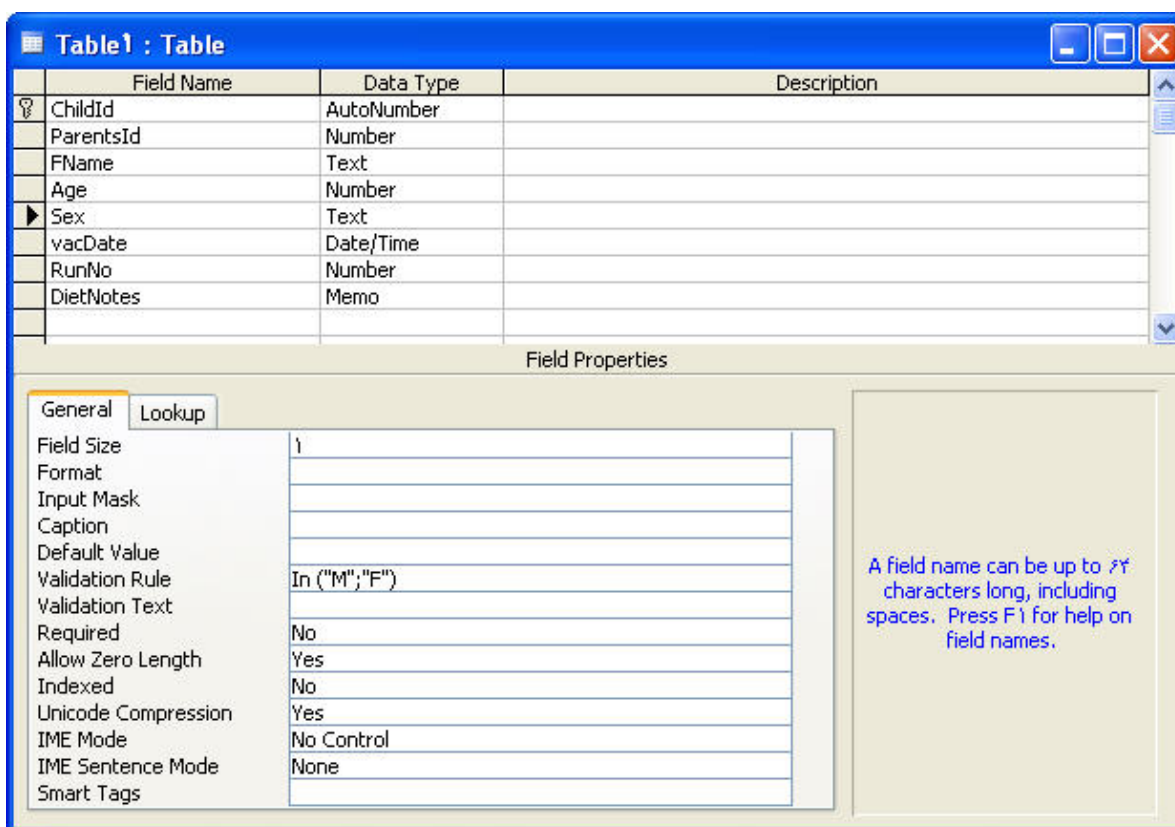
- ParentsId از نوع داده‌ی Number است و اندازه‌ی فیلد (Field size) آن Long Integer است. این موضوع بسیار مهم است چراکه باید بتوانیم Children و Parents را به هم مرتبط سازیم. ParentsId در جدول Parents ، والدین را مشخص می‌کند و از نوع AutoNumber است. اگر بخواهیم این دو جدول را به هم مرتبط کنیم باید فیلد متناظر در جدول دیگر از نوع Long Integer باشد. در این حالت ParentsId در جدول Children ، به عنوان کلید خارجی (Foreign key) شناخته می‌شود و کلید اصلی جدول Parents است.
- Age از نوع Number است. می‌توانیم اندازه فضایی که برای این فیلد ذخیره می‌شود را کاهش دهیم ، بدین صورت که اندازه‌ی فیلد را Byte قرار دهیم. این نوع ، امکان ذخیره‌ی تا مقدار 255 را فراهم می‌کند و بیشتر کودکان بیش از 100 سال زندگی نمی‌کنند و حد مقدار حتی بیشتر از کفایت است.
- Sex از نوع Text است. در قواعد کنترلی (validation rule) مقدار In ("M" ; "F") قرار می‌دهیم. این قاعده به معنی آن است که مقدار وارد شده در این فیلد باید M یا F باشد. چون این مقدار یک حرف است ، باید اندازه‌ی فیلد 1 قرار داده شود. (مقدار پیش‌فرض اندازه‌ی فیلد text ، 50 بوده است).
- VacDate از نوع Date/Time در نظر می‌گیریم. (به خاطر داشته باشید که این نوع فقط مقادیر تاریخ میلادی را ذخیره می‌کند و می‌توان محاسبات ریاضی روی آن اعمال کرد. در صورتی بخواهید تاریخ شمسی را ذخیره نمایید باید از نوع text و اندازه‌ی 10 استفاده کنید ولی دیگر قادر به استفاده از توابع تاریخ و عملگرهای ریاضی روی آن نخواهید بود).
- RunNo عددی مثبت با اندازه‌ی فیلد Byte است. Access به‌طور اتوماتیک مقدار پیش‌فرض (Default Value) صفر را برای آن لحاظ می‌کند. ما از این خصوصیت در زمان ایجاد Query استفاده خواهیم کرد.
- DietNotes از نوع فیلد Memo است. چون نوع text می‌تواند حداکثر تا 255 کاراکتر را دربر گیرد. این مقدار برای توضیحات ما کافی نیست بنابراین از نوع Memo استفاده می‌کنیم و کاربرد آن را نشان دهیم.

#### کلیدهای میانبر :

- کلید TAB ، جابه‌جایی بین ستون Field Name و Field Type.
- کلید F6 ، برای جابه‌جایی بین بخش Field Name/Field Type و بخش Field Properties.
- ALT + down arrow باز کردن لیست. برای جابه‌جایی در لیست از کلیدهای

## ایجاد جدول جدید با استفاده از نمای Design

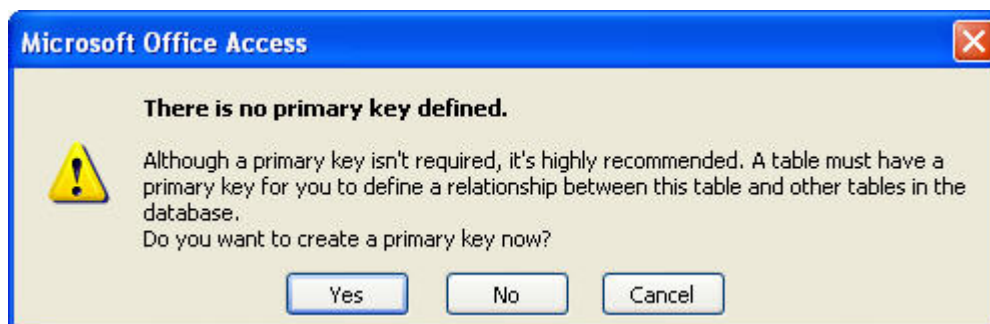
- روی دکمه New در بخش جدول‌های پنجره پایگاه داده کلیک کنید.
- در پنجره New Table گزینه‌ی Design View را انتخاب و OK نمایید.
- نام فیلد و نوع فیلد را مطابق خصوصیات تعریف‌شده‌ی بالا وارد نمایید.



- یادآوری می‌شود که در قسمت Field Properties ، خصوصیات را به صورت زیر تغییر دهید :
- برای فیلد ParentsId ، اندازه‌ی فیلد را Long Integer قرار دهید.

براي فيلد Age ، اندازهي فيلد را Byte قرار دهيد.  
براي فيلد Sex ، اندازهي فيلد را 1 و Validation Rule  
را يه In ("M"; "F") قرار دهيد.  
براي فيلد RunNo ، اندازهي فيلد را Byte قرار دهيد.

با ذخیره‌ی جدول به نام TableChildren ، Access پیغام زیر را نمایش خواهد داد با انتخاب Yes ، فیلد ChildrenId کلید اصلی می‌شود.

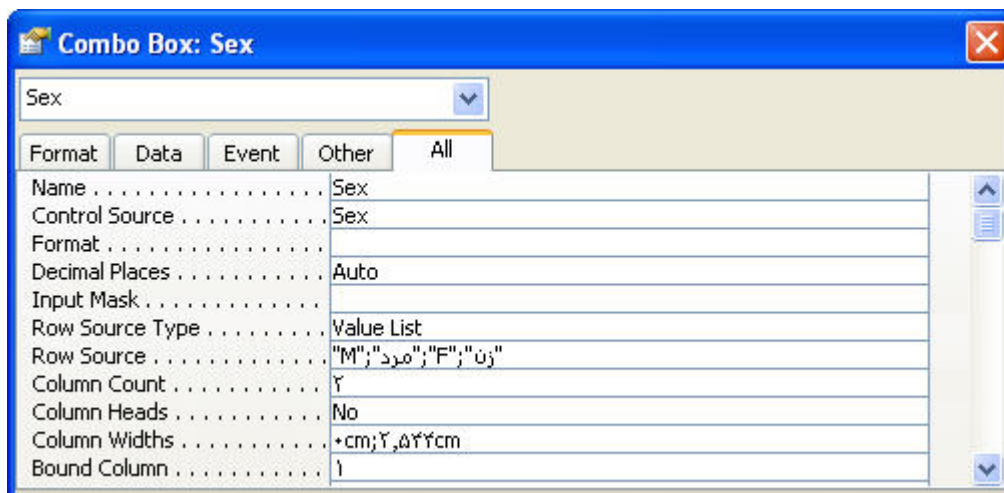
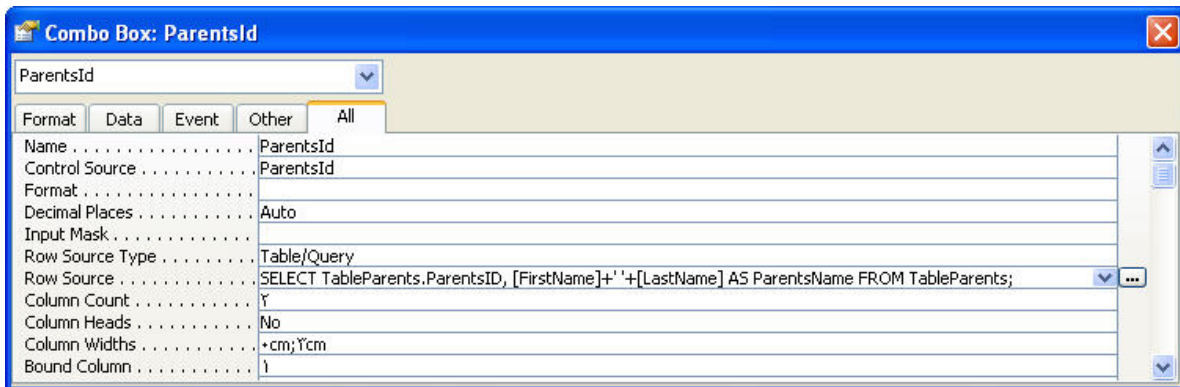
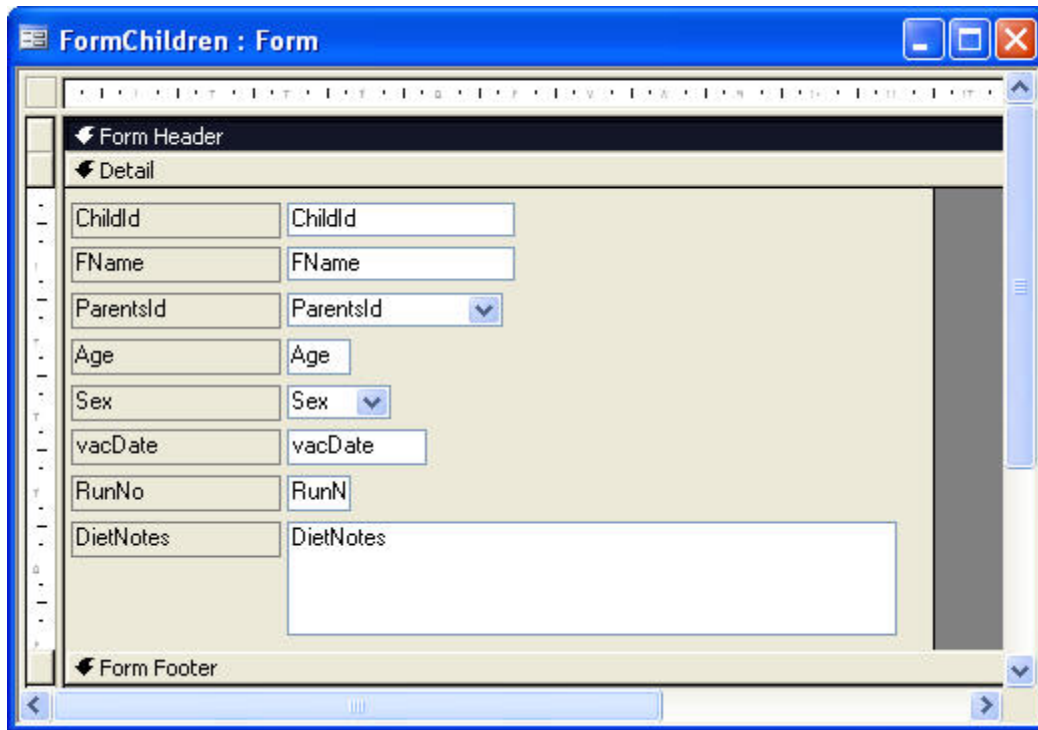


### نکته :

1) دقت کنید که در این حالت فیلد ChildrenId چون از نوع AutoNumber است ، کلید اصلی می‌گردد. در صورتی که فیلدی از نوع AutoNumber در جدول موجود نباشد ، یک فیلد جدید با نام ID و از نوع AutoNumber ایجاد می‌گردد و کلید اصلی می‌شود.  
 2) تعریف کلید اصلی به صورت زیر است: "یک یا چند فیلد که مقدار یا مقادیر آن‌ها هر رکورد را به صورت منحصر به فرد تعریف می‌کند. در یک ارتباط کلید اصلی به رکوردهای مشخصی در جدول دیگر اشاره می‌کند. کلید اصلی که در جدول دیگر اشاره می‌شود ، کلید خارجی نامیده می‌شود."

بر اساس جدول TableChildren و با استفاده از Form wizard یک فرم Columnar ایجاد نمایید . نوع ParentsId و Sex را به Combo Box تبدیل کنید . مشخصات تغییر Combo Box ها در زیر آورده شده است. سپس از این فرم برای ورود اطلاعات زیر به جدول استفاده کنید.

ChildId	ParentsName	FName	Age	Sex	vacDate	RunNo	DietNotes
۱	سجید میرزایی	علی	۲	M	۲۰۰۵/۰۶/۱۶	۱	حساسیت به گوجه فرنگی
۵	منیره میرزایی	محمد	۴	M	۲۰۰۵/۰۵/۲۱	۰	حساسیت به نینن زنبور
۳	جمیله پوررضایی	فریما	۵	F	۲۰۰۳/۱۱/۰۶	۱	
۶	جمیله پوررضایی	فاطمه	۲	F	۲۰۰۶/۰۹/۱۵	۱	
۲	رضا حکمتی	پوریا	۱	M	۲۰۰۵/۱۰/۰۵	۱	داروی آنژی بیوتیک هر ۶ ساعت
۳	محمد رضا بهمنی	سناپتن	۳	F	۲۰۰۶/۱۱/۱۵	۱	
▶ AutoNumber							



### نکته :

به خاطر داشته باشید که ممکن است AutoNumber های تولید شده در جدول بانك شما متفاوت باشد. اگر اشتباه کرده باشید و رکوردي را حذف کنید ، مقدار Autonumber آن رکورد نیز از دست می‌رود و عدد قبلي هرگز براي رکوردهای جدید ایجاد نمی‌گردد.

## 8-11- پرسوجوها Queries

يك کلاس دیگر از اشیاء محیط Access ، پرسوجو است. از پرسوجو برای مشاهده ، تغییر و تحلیل داده‌ها به طرق مختلف استفاده می‌شود . همچنین از آن به عنوان يك منبع رکورد برای فرم‌ها و گزارش‌ها استفاده می‌شود.

### قدم‌های ایجاد پرسوجو عبارت است از :

- مشخص کنید چه نوع پرسوجویی می‌خواهید ایجاد کنید.
- جدول‌هایی که داده‌های موردنظر را دربر دارند اضافه کنید.
- فیلدهایی که موردنظر است انتخاب کنید.
- معیاری که بر آن اساس Access مشخص می‌کند چه رکوردهایی و با چه محدودیتی نمایش داده شوند را مشخص کنید.

### ساخت پرسوجو برای استفاده در گزارش

گزارشی نیاز داریم که نیازهای تغذیه‌ای کودکانی که در مهدکودک نگهداری می‌شوند را نشان دهد. می‌توانیم به سادگی از روی جدول TableChildren گزارش را بسازیم . در این حالت مربیان مهدکودک باید کودکانی که  $RunNo > 0$  است یعنی امسال در مهدکودک ثبت‌نام شده‌اند را از لیست گزارش جدا کنند. معمولاً تعداد کل کودکانی که به مهد آمده‌اند به مراتب بیشتر از کودکان فعلی است و کار کنترل چشمی مشکل‌تر می‌شود. همچنین چنین گزارشی اطلاعات زاید زیادی دربر خواهد داشت که نه تنها گمراه کننده خواهد بود ، بلکه ارتکاب اشتباه را نیز بیشتر می‌سازد.

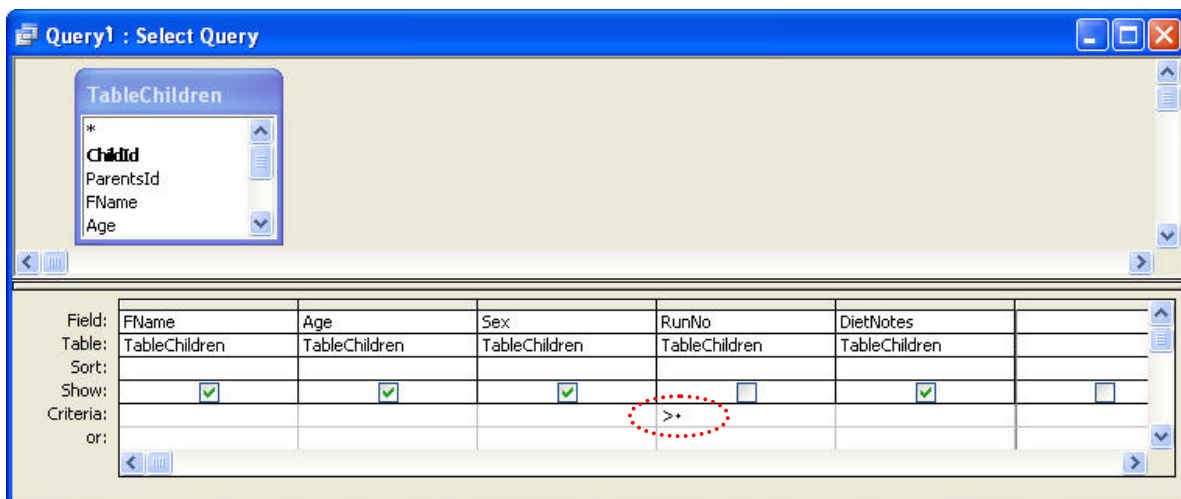
در این حالت شرط مقدار  $RunNo=0$  نشان می‌دهد آن کودک در مهدکودک نیست. فیلد RunNo زمانی که کودک در مهدکودک ثبت‌نام می‌شود به‌هنگام می‌شود و مقدار يك می‌گیرد و در

پایان سال یا زمانی که کودک ، مهد را ترک کند به هنگام شده و صفر می شود.


### ایجاد یک پرسوجو با معیار ثابت :

- در پنجره پایگاه داده ، زبانه Query را انتخاب نمایید و روی New کلیک کنید.
- در پنجره New Query روی Simple Query Wizard کلیک و OK نمایید.
- در لیست Simple Query Wizard Box ، در قسمت Tables/Queries جدول TableChildren را انتخاب نمایید.
- از لیست Available Fields ، Name را انتخاب و کلید > را کلیک کنید تا این فیلد به قسمت لیست Selected Fields منتقل شود. در مورد Age ، Sex و DietNotes نیز همین گونه عمل کنید. سپس روی Next کلیک کنید.
- اطمینان یابید که Detail انتخاب شده باشد و کلیک Next را بزنید.
- عنوان Query Children in Runs را برای پرسوجو برگزینید. اطمینان یابید که Modify Query Design انتخاب شده باشد و سپس روی Finish کلیک نمایید.
- پرسوجو باید در نمای Design مشاهده شود. در غیر این صورت با کمک دکمه View روی نوار ابزار ، نما را به Design تغییر دهید.
- در پنجره طراحی پرسوجو (شکل زیر) ، روی ردیف Criteria در ستون RunNo کلیک کنید و عبارت 0 > را تایپ نمایید. توجه کنید که نباید این عبارت در [] قرار گیرد چراکه در این صورت ، Access در مورد مقدار آن سوال می کند.





- از منوی File ، پرسوجو را با نام QueryChildrenInRuns ذخیره نمایید.

- در نمای Design ، دکمه‌ی میانبر  روی نوار ابزار را برای اجرای پرسوجو کلیک نمایید. نتیجه‌ی پرسوجو شبیه شکل زیر خواهد بود.

	FName	Age	Sex	DietNotes
▶	علی	۲	M	حساسیت به گوجه فرنگی
	پوریا	۱	M	داروی آنژی بیوتیک هر ۶ ساعت
	سکینش	۳	F	
	فریمه	۵	F	
	قلطمه	۲	F	
*		۰		

Record: 1 of 5

- پرسوجو را Close نمایید.

## 8-12- گزارش‌ها Reports

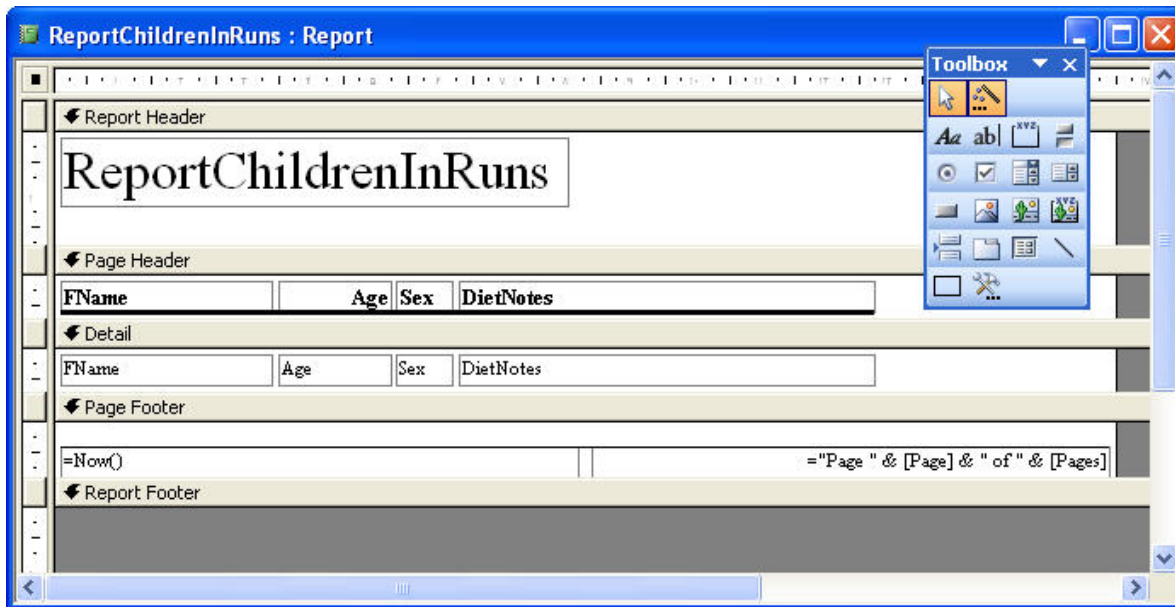
کلاس بعدی اشیاء که ایجاد می‌کنیم ، گزارش است. گزارش روشی برای نمایش محتوای داده‌های بانک است که مشابه صفحه‌های قابل چاپ باشد. به کمک آن مشخص می‌کنید کدام فیلدها چاپ شوند و چه داده‌هایی نمایش داده شوند. همچنین می‌توان خلاصه‌ی داده‌ها را نیز نمایش داد.

**Access** امکان طراحی گزارش‌ها با الگوی خاص را فراهم می‌آورد مثلا : محل سرتیتر ، اسامی فیلدها ، خود داده‌ها به همراه گرافیک و تصویر خاص **Logo**. گزارش داده‌ها را از جدول‌ها ، پرس‌وجوهای ایجادشده و یا پرس‌وجوهای که منحصرأ برای گزارش و در زمان طراحی گزارش ایجاد می‌شوند ، دریافت می‌کند. بهتر است ابتدا پرس‌وجوی موردنظر برای ایجاد گزارش ایجاد شود که تمام فیلدهای موردنیاز را دربر داشته باشد ، سپس از آن برای ایجاد گزارش استفاده شود.

### ایجاد يك گزارش و اصلاح آن:

می‌توان از پرس‌وجوی **QueryChildrenInRuns** به عنوان مبنای ایجاد گزارش استفاده کرد. اگرچه گزارشی که به وسیله **Report Wizard** ایجاد می‌شود دقیقا آن چیزی نیست که موردانتظار ماست ، ولی ما از آن برای ایجاد گزارش استفاده می‌کنیم و سپس طراحی آن را تغییر داده و اصلاح می‌کنیم .

- در پنجره‌ی پایگاه داده روی دکمه **Reports** کلیک می‌کنیم و سپس **New** را کلیک می‌کنیم .
- در پنجره‌ی **New Report** ، **Auto Report Tabular** را در نوع گزارش انتخاب می‌کنیم . از لیست **Table/Query** ، **QueryChildrenInRuns** را انتخاب می‌کنیم و **OK** می‌کنیم .
- گزارش در نمای **Print Preview** باز می‌شود. روی دکمه‌ی **Design View** در نوارابزار کلیک کنید تا شکل ظاهری گزارش به صورت زیر ظاهر شود و گزارش را با نام **ReportChildrenInRuns** ذخیره کنید.



### انتخاب اشیاء در گزارش

قبل از آن که کاری روی یک شیء انجام شود ، باید آن شیء انتخاب گردد. زمانی که در حالت نمایی Design قرار دارید با کلیک روی یک شیء می‌توانید آن را انتخاب کنید. در فرم‌ها و گزارش‌ها می‌توان همزمان چندین شیء را انتخاب کرد.

#### نکته :

- برای انتخاب اشیاء نزدیک به هم ، کلید Shift را نگهداشته و به ترتیب روی اشیاء کلیک کنید.
- برای انتخاب اشیایی که نزدیک هم نیستند ، کلید Ctrl را نگهداشته و به نوبت روی اشیاء کلیک کنید.
- یک شیء خصوصیات متنوعی دارد مثل Data ، Events و ....
- خصوصیات اشیاء مختلف ، متفاوت هستند و تمام اشیاء ، تمام خصوصیات را ندارند.

### مرتب‌سازی داده‌ها در گزارش

پس از آنکه اشیاء را انتخاب کردید ، می‌توانید عملیات موردنظر را روی آن‌ها اعمال کنید ، مثلاً مرتب‌سازی داده‌ها یا تغییر خصوصیات مثل نوع قلم (Fonts) ، رنگ‌ها (Colours) ، چیدمان (Alignment) .

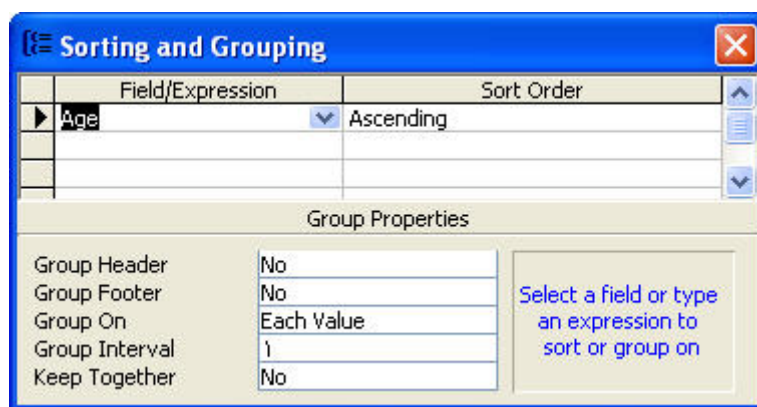
داده‌های جدول TableChildren که پرسوجوی QueryChildrenInRuns ما مبتنی بر آن است ، به ترتیب داده‌های وارد شده مرتب

شده است. ولي شايد اطلاعات مرتب شده بر اساس سن كودك مناسبتر باشد.

- اطمینان داشته باشید كه ReportChildrenInRuns در نماي Design باز باشد.

- روي دكمه ي Sorting and Grouping در نوارابزار كليك نماييد. پنجره اي ظاهر ميشود.

- در اين پنجره ، از ستون Field/Expression فيلد Age را انتخاب كنيد. (در صورت انتخاب چند فيلد براي مرتبسازي ميتوانيد با استفاده از ▼ در ليست حركت كنيد).



- پنجره را با دكمه ي (□) ببنديد. (دقت كنيد كه فقط اين پنجره را ببنديد ، نه كل گزارش را)

- روي دكمه ي Print Preview در نوارابزار كليك كنيد تا ليستي ملاحظات غذايي كودكان فعلي مهدكودك را مشاهده كنيد (شكل زير). اين ليست بر اساس Age مرتب شده است.



FName	Age	Sex	DietNotes
چوربا	1	M	داروي آنتي بيوتيک هر 6 ساعت
فاطمه	2	F	
علي	2	M	حساسيت به گوجه فرنگي
سكاشن	3	F	
فر يماه	5	F	

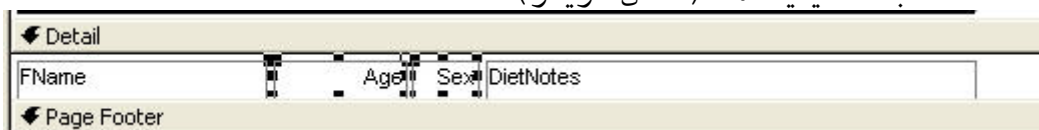
### تغيير خصوصيات شيء


يکي از تغييرات مفيد و تاثيرگذار در ظاهر گزارش ، خصوصيت Alignment فيلدهاي name ، Age و Sex است. اين خصوصيت قالب نمايش اطلاعات در اين فيلدها را تغيير مي دهد و از نوع Format است.

#### نکته :

- اگر پنجره ي خصوصيات مشاهده نمي شود. Properties را از کليک راست ماوس انتخاب نماييد.
- براي تغيير يک خصوصيت ، روي فيلد کليک کنيد. در پنجره ي Properties ، براي برخي خصوصيات علامت نمايش داده مي شود. بدین معني که مي توانيد از يک منو ، مقدار خصوصيت را انتخاب کنيد. براي ساير خصوصيات ، نياز به تايپ مقدار داريد.

- اطمینان يابيد که گزارش در نماي Design باز شده باشد. در قسمت detail گزارش ، فيلدهاي Age ، Sex را انتخاب نماييد. (شکل زير)



- در پنجره‌ی Properties ، خصوصیت Text align (در زبانه‌ی Format) را انتخاب و مقدار آن را Left قرار دهید. یا از دکمه‌های  روی نوار ابزار ، چیدمان نمایش متن را تغییر دهید و فونت مناسب فارسی را انتخاب کنید.
- چیدمان قسمت عنوان را وسطچین Center قرار دهید و عنوان آن را به "گزارش ملاحظات غذایی کودکان" تغییر دهید. طول اندازه‌ی آن را بزرگتر کنید و فونت آن را مناسب فونت فارسی تنظیم کنید.

#### نکته :

برای نمایش بهتر متن‌های فارسی باید فونت فارسی مناسب نصب شود. پس از نصب می‌توانید از هر کدام که در نمای Preview گزارش بهتر باشد و همچنین در چاپ نیز کیفیت مناسبی داشته باشد ، استفاده کنید. ما از Nazanin استفاده کرده‌ایم.

- گزارش را ذخیره نمایید. سپس با دکمه‌ی Preview آن را مشاهده کنید.



نام	سن	جنسیت	ملاحظات و رژیم غذایی
پوریا	۱	M	داروی آنتی بیوتیک هر ۶ ساعت
فاطمه	۲	F	
علی	۲	M	حساسیت به گوجه فرنگی
ستایش	۳	F	
فریماه	۵	F	



### 8-13- اضافه شدن نیازهای جدید

تحلیل سیستم در دو بخش طراحی منطقی (logical) و طراحی فیزیکی (physical) صورت می‌گیرد. همان‌طور که در بیشتر مواقع نیز رخ می‌دهد، مشتری نیازهای خود را در ابتدا و شروع پروژه به طور کامل بیان نمی‌کند و حتی شاید در ابتدا آن را نداند.

کارفرمای مهدکودک ما پس از مشاهده‌ی پایگاه داده‌ی ایجاد شده خوشحال شده است و نیازهای جدیدی را مطرح می‌سازد. مثلاً می‌خواهد اطلاعات رزرو ثبت‌نام را داشته باشد. برای این منظور ابتدا ساختار داده‌ای (جدول‌ها) تحلیل می‌شود تا کسری اطلاعات اضافه شود و سپس مراحل بعدی طراحی و از آن جمله شناسایی سایر اشیاء پایگاه داده صورت می‌گیرد.

#### نیازهای جدید مطرح شده

والدین در تابستان و از قبل، ثبت‌نام کودک خود را به‌صورت رزرو انجام می‌دهند. مهدکودک موردنظر دوست دارد بداند:

- چه تعدادی درخواست رزرو داشته‌اند؟
- چه زمانی کودک به مهدکودک آورده می‌شود؟
- برای سال جدید چه تعداد تخت و اتاق نگهداری لازم است؟
- هزینه‌ی نگهداری مثل استفاده از تخت و غذا چقدر است؟
- هزینه‌ی ایجاد گرما چقدر است؟ (در زمستان برای همه و در سایر فصل‌ها برای برخی خردسالان موردنیاز است)
- هزینه‌ی بیمه چقدر می‌شود؟

نرخ‌های روزانه به صورت زیر است: (نرخ‌ها در زمان‌های مختلف تغییر می‌کنند)

- هزینه‌ی نگهداری: به ازای هر اتاق 3000 + به ازای هر کودک 2000
- هزینه‌ی تولیدحرارت: به ازای هر اتاق 50000
- هزینه‌ی بیمه: به ازای هر کودک 10000

#### تحلیل نیازهای جدید

قبل از هر چیز و قبل از ایجاد اشیاء جدید، باید به سوالات زیر پاسخ دهیم:

- آیا می‌توانیم نیازهای جدید را با کمک اطلاعات موجود در پایگاه داده ارزیابی کنیم؟



- **پاسخ** خیر، اطلاعات موردنظر رزرو در هیچ جدولی (نه Parents و نه Children) وجود ندارد.
- اگر نه، چه اطلاعات بیشتری لازم است تا در پایگاه داده ذخیره شود؟
- **پاسخ** باید جزییات رزرو را ذخیره کرد مثل: تاریخ ثبتنام، زمان ورود کودک، تعداد کودکان، تعداد اتاقها و نرخهای روزانه.
- آیا میتوان با افزودن مشخصه‌ها (فیلدها) جدید به موجودیتهای موجود (جدولها) آنها را ذخیره کرد؟
- **پاسخ** چون والدین رزرو را انجام میدهند، ممکن است وسوسه شوید و بخواهید فیلدهای جدید را به جدول والدین اضافه کنید. با توجه به مفاهیم اولیه و بحث نرمالسازی میدانید که این کار موجب آنومالی‌هایی میشود.
- اگر نیاز به افزودن جدول جدید است، چه موجودیتهایی باید در نظر گرفته شود؟
- **پاسخ** موجودیت رزرو (Booking) مطرح میشود.
- اگر تصمیم به افزودن جدولهای جدید گرفته ایم، ارتباطات آنها با جدولها دیگر چیست؟
- **پاسخ** ارتباط، بین والدین و رزرو وجود دارد ولی ارتباطی بین کودک و رزرو وجود ندارد. هر از والدین ممکن است یک یا بیشتر رزرو داشته باشد. هر رزرو فقط به وسیله ی یک فرد صورت می‌گیرد. در اصطلاح پایگاه داده این نوع ارتباط از نوع یک-به-چند (One-to-Many) است.
- کلید اصلی هر یک از جدولها جدید چیست؟
- **پاسخ** شماره رزرو را به عنوان کلید اصلی ایجاد میکنیم. و ثبت آن را در جدول به Access محول میکنیم. یعنی شماره رزرو از نوع AutoNumber باشد.
- چه کلیدهای خارجی باید به خصوصیات جدول جدید یا جدولها قبلی اضافه شود؟
- **پاسخ** ارتباط جدول جدید با والدین از نوع یک-به-چند است بنابراین کلیدخارجی ParentsId را به جدول جدیدمان اضافه میکنیم.
- تحلیل بالا نشان میدهد تا اینجا، دو شیء جدید مورد نیاز است. جدول TableBookings که داده‌های رزرو را نگه میدارد. همچنین شیء فرم FormBookings که برای ورود اطلاعات رزرو موردنیاز است.

### نکته :

همیشه برای تعیین نوع داده‌ی فیلدهای جدول دقت کنید ، چراکه فضای حافظه‌ی موردنیاز را تعیین می‌کند.

### طراحی منطقی جدول رزرو Booking

اکنون که نیاز به جدول جدید را بررسی کردیم ، باید فیلدهای آن را تعیین کنیم .

- این جدول نیز مانند سایر جدول‌ها ، نیاز به شناسه‌ی منحصر به فردی دارد که آن را BookingId می‌نامیم و نوع آن را AutoNumber قرار می‌دهیم .

- همچنین باید مشخص کنیم چه کسی رزرو را درخواست کرده است . می‌دانیم ParentsId ابزار ایجاد این ارتباط است . همان‌طور که از آن به عنوان کلید خارجی در جدول کودکان استفاده کردیم ، آن را به جدول رزرو نیز اضافه می‌کنیم و همان‌طور که قبلاً گفتیم نوع آن باید Long Integer باشد .

- همچنین باید بدانیم ، کودک چه زمانی به مهدکودک آورده می‌شود . این فیلد را ChildInDate می‌نامیم و نوع آن را Date فرض می‌کنیم .

- برای ذخیره تعداد کودکان در درخواست . فیلد NoOfChildren را از نوع Byte در نظر می‌گیریم .

- برای ذخیره‌ی تعداد اتاق‌های مورد درخواست ، فیلد NoOfRooms را از نوع Byte در نظر می‌گیریم .

- برای دانستن هزینه‌ی نگهداری به اطلاعات تعداد کودکان ، تعداد اتاق‌های موردنیاز ، تعداد روزها و نرخ‌های روزانه نیاز داریم تا هزینه‌ی نگهداری محاسبه شود . بنابراین این فیلد محاسباتی است و نیازی به ذخیره آن در یک فیلد نداریم . با این وجود می‌دانیم که نرخ‌ها متغیر هستند بنابراین به جای ذخیره‌ی فیلد هزینه ، دو فیلد دیگر را اضافه می‌کنیم : RoomRate و ChildRate به ترتیب نرخ هزینه‌ی اتاق و نرخ کودک . هر دو آن‌ها از نوع Currency باشد (چون نشان‌دهنده‌ی قیمت هستند) . با عبارت زیر می‌توان هزینه‌ی نگهداری را حساب کرد .

( [RoomRate]\*[NoOfRooms] + [ChildRate]\*[NoOfChildren] )

- همچنین باید هزینه‌ی تولید حرارت را در نظر بگیریم. می‌توانیم HeatRate را از نوع داده‌ی Currency ذخیره کنیم و از عملگر IIF() برای محاسبه‌ی هزینه‌ی تولید حرارت در غیر زمستان که فقط برای خردسالان لازم است استفاده کنیم. در این عملگر پیش‌ساخته ، بسته به درستی یا نادرستی عبارت قرار گرفته در قسمت اول ، قسمت دوم یا قسمت سوم را اجرا می‌کند.

**Syntax** Iif(expr ; truepart ; falsepart)

### آرگومان‌های تابع :

- Expr** : عبارتی است که باید محاسبه شود.
- Truepart** : مقدار یا عبارتی است که در صورت درست بودن عبارت expr برمی‌گردد.
- Falsepart** : مقدار یا عبارتی است که در صورت نادرست بودن عبارت expr برمی‌گردد.

عبارت محاسبه‌ی هزینه‌ی تولید حرارت عبارت است از :

$Iif( [Heating] , [HeatRate] * [NoOfRooms] , 0 )$

- هزینه‌ی بیمه ، اگر نرخ آن را بدانیم به آسانی قابل محاسبه است. با داشتن فیلد InsRate (نرخ بیمه) از نوع Currency می‌توان این هزینه را با فرمول زیر محاسبه کرد :

$[InsRate]*[NoOfChildren]$

قبل از ایجاد جدول ، به طراحی منطقی FormBookings نگاهی می‌اندازیم تا ببینیم چه اشیایی در فرم مورد نیاز است. با طراحی منطقی این فرم ( که ورودی و خروجی جدول TableBookings را فراهم می‌آورد) می‌توانیم طراحی جدول را کنترل کنیم و از هر گونه خطای منطقی در ساختار جدول جلوگیری شود.

### طراحی منطقی فرم رزرو Booking

فرم تعریف رزرو

شناسه رزرو :	نام والدین :
تعداد کودکان :	زمان ورود :
اتاق‌های مورد نیاز :	
نرخ شارژ روزانه‌ی اتاق :	نرخ شارژ روزانه‌ی کودک :
نرخ شارژ روزانه‌ی تولید حرارت :	نرخ بیمه :
هزینه نگهداری :	
هزینه تولید حرارت :	
هزینه بیمه :	

اشياء زير در فرم موردنياز است:

**BookingNo** : شناسه ي منحصر به فرد جدول را نشان مي دهد. تا در صورت نياز به اطلاع ده ي به وسيله ي اين شماره به والدين اطلاع داده شود. از شيء **Text Box** براي آن استفاده مي شود. منبع داده اي آن فيلد **BookingNo** از جدول **Bookings** است.

**ChildInDate** : زمان ورود كودك به مهدكودك را نشان مي دهد. شيء **Text Box** براي آن مناسب است و با فيلد **ChildInDate** جدول **Bookings** در ارتباط است و بنا بر اين خصوصيت **Control Field** آن ، **ChildInDate** است.

**NoOfRooms** : تعداد اتاقهاي موردنياز براي نگهداري كودك است. شيء **Text Box** كه خصوصيت فيلد متناظر آن **NoOfRooms** است ، براي آن مناسب است.

**Heating** : براي ذخيره آنكه حرارت موردنياز است يا خير استفاده مي شود. از **Text Box** براي آن استفاده مي كنيم كه خصوصيت فيلد متناظر آن **Heating** است.

**RoomRate** : نرخ شارژ روزانه ي اتاق است. **Text Box** اي قرار مي دهيم كه به **RoomRate** متصل مي شود.

**ChildRate** : نرخ شارژ روزانه ي كودكان است. **Text Box** اي است كه به **ChildRate** متصل است.

**HeatRate** : نرخ شارژ روزانه ي توليد حرارت است. **Text Box** اي قرار مي دهيم كه به **HeatRate** متصل است.

**InsRate** : نرخ شارژ روزانه بيمه است. **Text Box** اي است كه به **InsRate** متصل است.

**Text Box** اي براي نمايش **KennelCost** كه هزينه ي نگهداري را محاسبه و نشان دهد.

**Text Box** اي براي نمايش **HeatingCost** كه هزينه ي توليد حرارت را محاسبه و نشان دهد.

**Text Box** اي براي نمايش **InsuranceCost** كه هزينه ي بيمه را محاسبه و نشان دهد.

حال بايد طراحي پيشنهاده شده را با نيازهاي جديد تطابق دهيم. تا مشخص شود تمام اطلاعات موردنياز براي پاسخگويي به نيازهاي جديد فراهم آمده است.

### ايجاد جدول رزرو **Booking**

نام فيلد	نوع داده اي	اندازه فيلد
----------	-------------	-------------

BookingNo	AutoNumber	
ParentsId	Number	Long Integer
ChildInDate	Date/Time	
NoOfChildren	Number	Byte
NoOfRooms	Number	Byte
Heating	Yes/No	
RoomRate	Currency	
ChildRate	Currency	
HeatRate	Currency	
InsRate	Currency	

روش ایجاد جدول را به طور کامل آموخته اید. پس جدول TableBooking را با توجه به اطلاعات بالا ایجاد کنید.

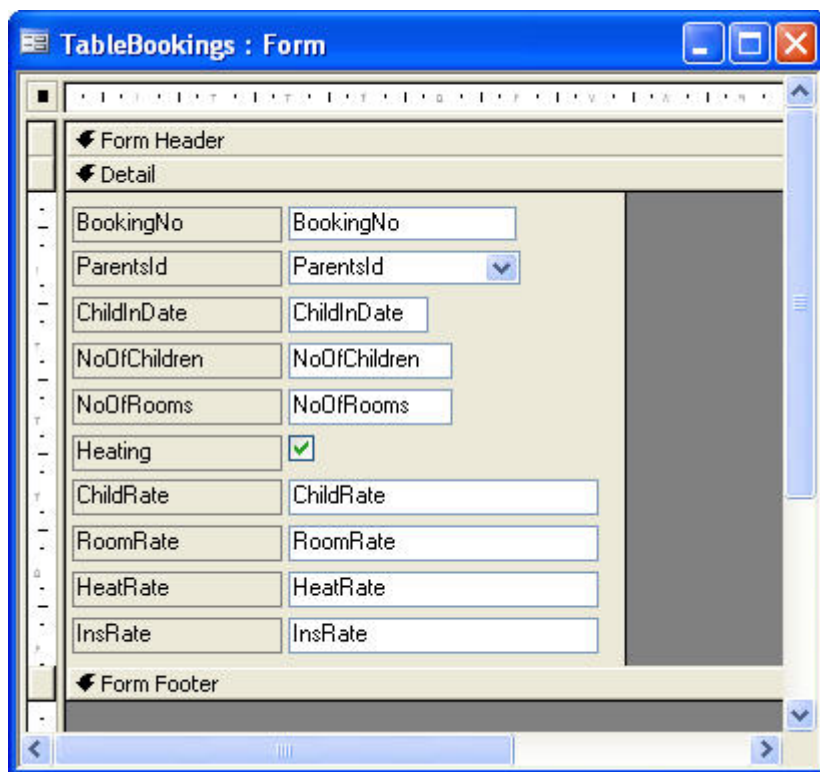
در نمای Datasheet داده های زیر را در جدول ذخیره کنید.

BookingNo	ParentsId	ChildInDate	NoOfChildrer	NoOfRooms	Heating	ChildRate	RoomRate	HeatRate	InsRate
۱	۲	۲۰۰۷/۰۱/۱۵	۱	۱	<input checked="" type="checkbox"/>	۲,۰۰۰,۰۰۰ ریال	۳,۰۰۰,۰۰۰ ریال	۱۰,۰۰۰,۰۰۰ ریال	۱۰,۰۰۰,۰۰۰ ریال
۲	۳	۲۰۰۷/۰۲/۲۵	۱	۱	<input checked="" type="checkbox"/>	۲,۰۰۰,۰۰۰ ریال	۴,۰۰۰,۰۰۰ ریال	۲۰,۰۰۰,۰۰۰ ریال	۱۰,۰۰۰,۰۰۰ ریال
۳	۷	۲۰۰۷/۰۹/۰۵	۲	۲	<input type="checkbox"/>	۲,۰۰۰,۰۰۰ ریال	۲,۵۰۰,۰۰۰ ریال		۱۰,۰۰۰,۰۰۰ ریال
۴	۹	۲۰۰۶/۱۱/۲۲	۱	۱	<input checked="" type="checkbox"/>	۲,۰۰۰,۰۰۰ ریال	۳,۰۰۰,۰۰۰ ریال	۱۰,۰۰۰,۰۰۰ ریال	۱۰,۰۰۰,۰۰۰ ریال
۵	۱	۲۰۰۶/۱۲/۱۵	۲	۱	<input type="checkbox"/>	۲,۰۰۰,۰۰۰ ریال	۳,۰۰۰,۰۰۰ ریال		۱۰,۰۰۰,۰۰۰ ریال
۶	۶	۲۰۰۷/۰۵/۱۴	۲	۱	<input checked="" type="checkbox"/>	۲,۰۰۰,۰۰۰ ریال	۶,۰۰۰,۰۰۰ ریال	۱۰,۰۰۰,۰۰۰ ریال	۲۰,۰۰۰,۰۰۰ ریال
۷	۷	۲۰۰۷/۰۱/۲۳	۱	۱	<input checked="" type="checkbox"/>	۲,۰۰۰,۰۰۰ ریال	۳,۰۰۰,۰۰۰ ریال	۱۰,۰۰۰,۰۰۰ ریال	۱۰,۰۰۰,۰۰۰ ریال
▶ AutoNur	۰		۰	۰	<input type="checkbox"/>	۰,۰۰۰ ریال	۰,۰۰۰ ریال	۰,۰۰۰ ریال	۰,۰۰۰ ریال

فیلدهای از نوع Yes/No به صورت Check box دیده می شوند. اگر انتخاب شده باشد ، Yes و در غیر این صورت No است. با کلید Space صفحه کلید می توانید با هر بار فشردن کلید ، بین انتخاب شدن و یا عدم انتخاب جابه جا شوید. این حالت را به صورت زیر می توانید تغییر دهید. در Field Properties فیلد مورد نظر کلیک کنید. در زبانه ی Lookup خصوصیت Display Control را مورد نظر خود انتخاب کنید.

### ایجاد فرم رزرو FormBooking

برای ایجاد فرم از روش ویزارد استفاده می کنیم و پس از آن اشیاء دیگر را به فرم افزوده و با تغییر می دهیم.



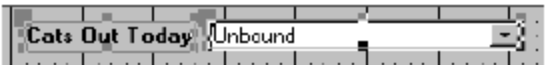
**نکته :**  
 اگر جعبه ابزار (ToolBox) را مشاهده نمی‌کنید ، از منوی View آن را کلیک کنید.

**جابجایی کنترل :**

می‌توانید آن را انتخاب نموده و به محل مناسب جابه‌جا نمایید و یا اندازه‌ی آن را تغییر دهید. برای انتخاب روی آن کلیک کنید. پس از انتخاب ، مربع‌های تغییراندازه و جابه‌جا کردن ظاهر می‌شوند (شکل زیر).

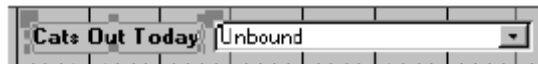


برای انتخاب برچسب متناظر آن باید با Shift ، روی برچسب مجدداً کلیک کرد. در این‌صورت برای آن نیز مربع‌های تغییراندازه ظاهر می‌شوند (شکل زیر).



برای انتخاب برچسب به تنهایی باید با Shift ، روی برچسب کلیک کرد و مجدداً روی Combo box کلیک کرد. در این‌صورت

مربع هاي تغييراندازه فقط براي برچسب ظاهر مي شود (شکل زیر).



با كليدهاي Arrow صفحه كليد و يا درگ خود شيء ، هميشه اين دو با هم جابه جا مي شوند. درصورتی که بخواهيد يکي را فقط جابه جا کنيد روي مربع جابه جايي (مربع بزرگ چپ-بالا) کليک و درگ مي کنيد.

براي حذف برچسب متصل به Combo Box ، آن را به تنهائي انتخاب و دکمه ي حذف را بزنييد.

### نکته :

اين تکنیک براي تغييراندازه و جابه جايي تمام کنترل ها و اشياء فرم به کار مي رود. همچنين نحوه ي حذف برچسب متصل نيز به همين روش صورت مي گيرد.


### افزودن Text Box براي نمايش فيلدهاي محاسباتي

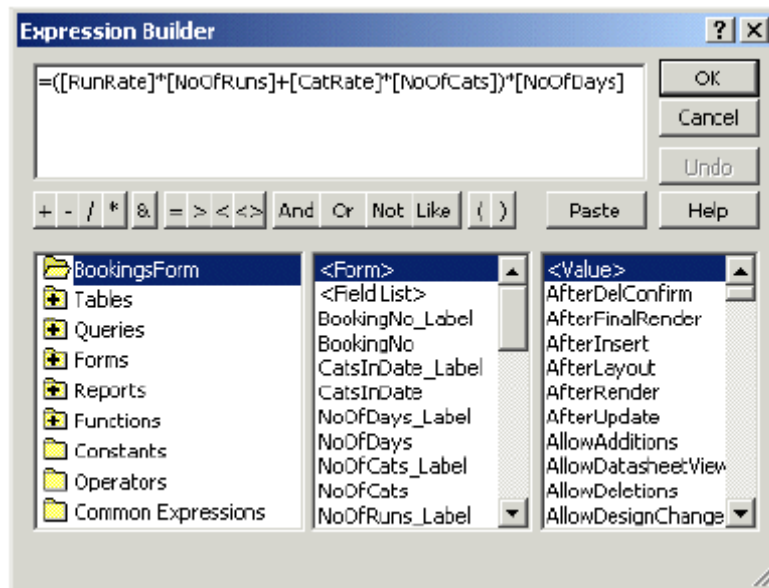
- اطمینان يابيد که فرم در نماي Design باز باشد. در جعبه ابزار (ToolBox) روي دکمه ي Text Box کليک کنيد.
- در فضاي خالي فرم مربعي رسم کنيد. کنترل به همراه برچسب ايجاد مي شود (شکل زیر).



- روي برچسب کليک کنيد. در پنجره ي Properties ، خصوصيات برچسب ظاهر مي شود.
- خصوصيت Caption را LabelKennelCost بگذاريد.
- در خارج پنجره کليک کنيد تا تغيير را روي فرم مشاهده کنيد.
- روي برچسب کليک کنيد تا انتخاب شود. ماوس را در مرز آنقدر جابه جا کنيد که شبیه فلش دوجته شده شود. سپس مرز را درگ کنيد تا جايي که کل متن برچسب مشاهده شود.
- Text Box را انتخاب کنيد و در پنجره ي خصوصيات ، خصوصيت Name را به FieldKennelCost تغيير دهيد.

در ادامه براي توليد عبارت محاسباتي از Expression Builder استفاده مي کنيم تا هزينه ي نگهداري را محاسبه کند. مي توان عبارت را تايپ کرد ولي با کمک Builder احتمال خطا و اشتباه تايپي کاهش مي يابد.

- در پنجره ي Properties روي زبانه ي Data كليك كنيد. سپس در خصوصيت Control Source ، روي  سمت راست كليك كنيد. پنجره ي Expression Builder نشان داده مي‌شود.
- روي "=" و سپس "(" در قسمت مياني كليك كنيد. اين دو در قسمت متني بالا نمايش داده مي‌شوند.
- از پنجره مياني قسمت سه بخشي پايين ، RoomRate را دوباركليك نماييد. (اگر آن را مشاهده نمي‌كنيد در ليست به پايين جابه‌جا شويد). در قسمت متني فيلد در بين [ ] قرار دارد.



- به اين ترتيب عبارت زير را با كليك روي عملگرهاي مياني و دوباركليك روي فيلد موردنظر ايجاد كنيد.  

$$=([RoomRate]*[NoOfRooms]+[ChildRate]*[NoOfChildren])$$

- OK را كليك كنيد تا Expression Builder بسته شود.
- خصوصيت Format را در پنجره ي خصوصيات ، Currency قرار دهید.

كنترل‌هاي محاسباتي Heating Cost ، Insurance Cost و Total Cost را نيز با افزودن Text Box و با كمك Expression Builder و عبارات تعريف شده در طراحي منطقي به فرم اضافه كنيد.



TableBookings : Form

Detail

BookingNo	BookingNo
ParentsId	ParentsId
ChildInDate	ChildInDate
NoOfChildren	NoOfChildren
NoOfRooms	NoOfRooms
Heating	<input checked="" type="checkbox"/>
ChildRate	ChildRate
RoomRate	RoomRate
HeatRate	HeatRate
InsRate	InsRate
LabelKennelCost	=([RoomRate]*[NoOfRooms]+[ChildRate]*[NoOfChildren])
HeatingCost	=([HeatRate]*[NoOfRooms])
InsuranceCost	=([InsRate]*[NoOfChildren])
TotalCost	=[FieldKennelCost]+[FieldHeatingCost]+[FieldInsuranceCost]

به کمک این فرم اطلاعات رزرو را مشابه زیر ایجاد کنید.

TableBookings

BookingNo	
ParentsId	سمید میرزایی
ChildInDate	۲۰۰۷/۰۱/۱۵
NoOfChildren	۱
NoOfRooms	۱
Heating	<input checked="" type="checkbox"/>
ChildRate	۲۰۰۰۰۰۰ ریال
RoomRate	۳۰۰۰۰۰۰ ریال
HeatRate	۱۰۰۰۰۰۰ ریال
InsRate	۱۰۰۰۰۰۰ ریال
LabelKennelCost	۵۰۰۰۰۰۰ ریال
HeatingCost	۱۰۰۰۰۰۰ ریال
InsuranceCost	۱۰۰۰۰۰۰ ریال
TotalCost	۲۵۰۰۰۰۰ ریال

Record: 1 of 7

منابع

1. Date C.J., An Introduction to Database Systems. 7<sup>th</sup> ed. Addison-Wesley. 2000.
  2. Codd. E.F., The Relational Model for Database Management. Version 2. Addison-Wesley. 1990.
  3. Viescas John L., Microsoft Office Access 2003 Inside Out. Microsoft Press. 2004.
  4. Office.microsoft.com, “Access 2003 home page”.
  5. Microsoft Office Access Help.
  6. Sheldon Robert, Microsoft Office Access 2003 A Beginner’s Guide. Edition 1. McGraw-Hill, 2003.
7. روحاني رانكوهي ، سيد محمدتقي ، مفاهيم بنيادي پايگاه داده ها ، انتشارات جلوه ، چاپ اول ، پاييز 1380.

## ضمیمه 1 :

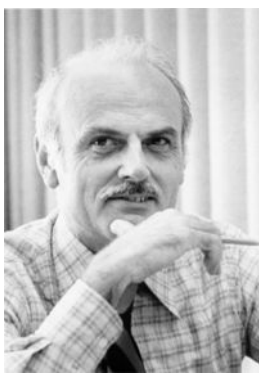
### میانبرهای صفحه کلید در Access

Ctrl + N	باز کردن بانک اطلاعاتی جدید
Ctrl + O	باز کردن بانک اطلاعاتی موجود
CTRL+N	ایجاد بانک اطلاعاتی جدید
Alt + F4	خروج
Ctrl + P	چاپ شيء انتخاب شده
Ctrl + S	دخیره کردن
Alt + V + P	باز کردن صفحه ی property برای شيء انتخاب شده
F5	مشاهده ی فرم در نمای طراحی
Ctrl + C	کپی شيء انتخاب شده به Clipboard
Ctrl + V	کپی شيء از Clipboard به محیط
Alt + SPACEBAR	مشاهده ی منوی کنترل
Ctrl + F4	بستن پنجره ی فعال
Alt + TAB	رفتن به پنجره ی بعدی در ویزارد
F7	بررسی spelling
F2	تغییر نام شيء انتخاب شده
Shift + F10	نمایش منو به صورت میانبر
F10	فعال کردن نوار منو
F1	راهنمای سیستم Help
Ctrl + Z	لغو عمل
Ctrl + P	بازکردن صفحه ی تنظیمات چاپ و چاپ
S	بازکردن صفحه ی Page setup
Z	بزرگنمایی Zoom
Alt + ENTER	نمایش صفحه ی مشخصه ها
CTRL+ENTER	ایجاد خط جدید در فیلد از نوع memo یا text
CTRL+'	اضافه کردن فیلد مشابه در رکورد قبلی به رکورد جاری
ESC	برگرداندن تغییرات اعمال شده روی فیلد جاری
ESC ESC (press ESC twice)	برگرداندن تغییرات اعمال شده روی رکورد جاری
F11	نمایش پنجره ی بانک داده
CTRL+F	جستجو کردن و جایگزین کردن

## ضمیمه 2 :

### معرفی Edgar F. Codd و Christopher J. Date

ادگار فرانک تد کاد (23 آگوست 1923 - 18 آوریل 2003)



کاد یک دانشمند علوم کامپیوتر و انگلیسی بود. وی نقش به‌سزایی در بنیانگذاری تئوری بانک رابطه‌ای داشت و می‌دانیم که این مدل رایج‌ترین مدل داده‌ای بانک اطلاعاتی است. زمانی که برای IBM فعالیت می‌کرد، مدل رابطه‌ای را برای مدیریت بانک اطلاعاتی بنا کرد. اگرچه نقش ارزشمند بسیاری در سایر پیشرفت‌های علوم کامپیوتر داشت ولی مدل رابطه‌ای وی از تئوری‌های مهم و به‌یادماندنی او است.

کاد در پورتلند انگلستان متولد شد. او ریاضی و شیمی را در دانشکده‌ی اکستر و آکسفورد گذراند. پس از آن مدتی را در نیروی هوایی سلطنتی به عنوان خلبان خدمت کرد. در سال 1948، به نیویورک رفت و به عنوان برنامه‌نویس با شرکت IBM کار کرد. کاد دکتری علوم کامپیوتر خود را از دانشگاه میشیگان دریافت کرد و دو سال بعد مجدداً در دفتر پژوهشی IBM فعالیت خود را آغاز کرد.

دو دهه (1960 و 1970) را بر روی تئوری‌های مدیریت داده کار کرد و در سال 1970 مقاله‌ی خود را در مورد مدل رابطه‌ای برای بانک‌های داده‌ی مشترک و بزرگ<sup>1</sup> منتشر کرد. متأسفانه شرکت IBM نظریه‌های او را به سرعت به کار نگرفت، تا آن جا که رقباتی تجاری شروع به استفاده از دانش وی کردند و به این ترتیب IBM پروژه‌ی System R را در تحقیقات بعدی خود در نظر گرفت. وی چون مسوولیت این کار را بر عهده‌ی برنامه‌نویسانی سپرد که با ایده‌ی کاد آشنا نبودند و تیم را نیز از ارتباط با کاد جدا نگه داشت، سیستمی ایجاد شد که رابطه‌ای نبود و آن را SEQUEL نامیدند. اگرچه همین سیستم ایجاد شده‌ی غیر رابطه‌ای نیز بر سیستم‌های رابطه‌ای مشابه که بر اساس مقالات منتشر شده‌ی دیگران<sup>2</sup> ایجاد شده بودند امتیاز بسیار داشت و نام آن را از SEQUEL به SQL تغییر دادند.

کاد بر توسعه و بسط مدل رابطه‌ای خود ادامه داد و با کریس دیت<sup>3</sup> همکاری خود را شروع کرد. بر این اساس پس از کاد، یکی از مدل‌های نرمال را به نام Boyce-Codd نامیدند که به BCNF معروف است. پس از دهه 1980 که مدل رابطه‌ای به اوج شهرت رسید، بحث‌های کاد با تولیدکنندگان بانک‌اطلاعاتی منجر به پیشنهاد اصول 12گانه‌ی وی شد که یک پایگاه داده‌ی رابطه‌ای باید از آن‌ها پیروی کند. با ایجاد زبان SQL، به نظر وی تئوری رابطه‌ای به نادرستی

<sup>1</sup> - "A Relational Model of Data for Large Shared Data Banks"

<sup>2</sup> - مقالات لری الیسون در مورد DBMS اراکل

<sup>3</sup> - Chris Date

پياده‌سازي شد و مخالفت‌هايش ، پست و مقام شغلي وي را در IBM با مشكل مواجه كرد كه شركت را ترك گفت. پس از آن وي با كمك كريس ديت و ديگر همراهان ، شركتي مشاوره‌اي راه‌اندازي كرد.

كاد جايزه‌ي تورينگ را در سال 1981 دريافت كرد و در سال 1994 وي به عنوان عضو افتخاري انجمن ماشين‌هاي محاسب منصوب شد.

كاد بر اثر سكته قلبي در منزل خود در جزيره‌ي ويليامز فلوريدا در 18 آوريل 2003 در سن 79 سالگي درگذشت.



### **كريستوفر جي ديت (23 آگوست 1923 - 18 آوريل 2003)**

ديت يكي از نويسندگان ، محققان ، مشاوران و متخصصان مستقل در زمينه‌ي فن‌آوري بانك داده‌ي رابطه‌اي است. زماني كه براي شركت IBM كار مي‌كرد ، در زمينه‌ي برنامه‌ريزي و طراحي فني محصولات شركت چون SQL/DS و DB2 همكاري داشت.

او زماني با كاد در مديريت پايگاه داده‌ها و مدل رابطه‌اي همكاري مي‌كرد. در سال 1983 شركت IBM را ترك كرد و يكي از توسعه‌دهندگان و پايه‌گذاران مدل رابطه‌اي شناخته مي‌شود.

كتاب وي به عنوان مقدمه‌اي بر سيستم‌هاي بانك داده ، تاكنون به هشتمين ويرايش خود رسيده است و كتابي مرجع در زمينه‌ي بانك‌هاي داده‌اي است و در صدها دانشكده و دانشگاه در سراسر دنيا تدريس مي‌شود. وي همچنين نويسنده‌ي كتاب‌هاي بسياري در زمينه‌ي مديريت داده است.

## واژه نامه ي تشریحي

Anomaly	<b>آنومالي</b> انجام ناپذیري يکي از عملیات در بانک ، بروز تبعات نامطلوب و یا فزون کاری در پی انجام يکي از عملیات در بانک را گویند.
Cardinality	<b>ماهیت ارتباط</b> تناظر بین موجودیت ها است که به سه نوع 1:1 ، 1:n و n:m وجود دارد.
Data Model	<b>مدل داده اي</b> مدلي است مفهومي که برای نمایش داده ها و ارتباط بین آن ها استفاده می شود. انواع رایج آن عبارت است از : رابطه اي ، سلسله مراتبي و شبکه اي.
Database	<b>پایگاه داده / بانک اطلاعاتي / بانک داده</b> مجموعه اي از داده ها که به صورت یکپارچه با حداقل افزونگی تحت یک سیستم متمرکز و در چارچوب یک مدل داده اي مدیریت می شوند.
Database Management system ( DBMS )	<b>سیستم مدیریت پایگاه داده</b> سیستمي است که مدیریت بانک اطلاعاتي را بر عهده دارد و هرگونه دستیابی به داده ها از طریق آن انجام می شود.
Entity	<b>موجودیت</b> هر مفهوم یا شیء در محیط عملیاتی است.
Entity-Relationship Diagram ( ER )	<b>نمودار ارتباط-موجودیت</b> نمودار نشان دهنده ي موجودیتها ، ویژگی های هر موجودیت ، ارتباط بین موجودیتها و بیان عملکرد و ماهیت ارتباط است. در این نمودار از شکل های استاندارد استفاده می شود.
Field	<b>فیلد</b> یک قطعه داده است ، همان واحد معنایی داده . نام فیلد به همراه مقدار آن در حکم یک اطلاع است.
Filing system	<b>سیستم فایلینگ</b> نرم افزار کاربردی است که خود ذخیره و بازیابی داده ها را نیز بر عهده دارد.
Foreign key	<b>کلید خارجی</b> کلید اصلي یک جدول است که بنا بر ماهیت ارتباط ، به جدول دیگر اضافه می شود و در آن جدول با نام کلید خارجی شناخته می شود و باعث ارتباط دو جدول می شود.
Form	<b>فرم</b> نمای گرافیکی برای نمایش داده است که امکانات اضافه ، ویرایش ، حذف و هر گونه عملیات روی داده ها را فراهم می سازد.

key	<b>کلید</b> فیلد یا مجموعه‌ای از فیلدها که باعث یکتایی هر یک از رکوردها شود.
Normal form	<b>فرم نرمال</b> تجزیه‌ای از جدول است که دارای آنومالی نباشد. دارای سطوح نرمال 1NF ، 2NF ، 3NF ، BCNF ، 4NF و 5NF است.
Operational Environment	<b>محیط عملیاتی</b> محیطی است که می‌خواهیم یک سیستم ذخیره و بازیابی برای آن ایجاد کنیم.
Primary key	<b>کلید اصلی</b> یکی از کلیدهاست که با توجه به ملاحظات به عنوان کلید اصلی جدول در نظر گرفته می‌شود. ممکن است از ویژگی‌های آن موجودیت نباشد و به صورت مجازی برای یکتا شدن رکوردهای جدول ایجاد شود.
Query	<b>پرس و جو</b> ابزار مناسب استخراج اطلاعات از جدول‌ها است. برای ایجاد آن از قواعد یک دستور زبان استاندارد به نام SQL استفاده می‌شود. اگرچه برخی سیستم‌های مدیریت پایگاه داده ، محیطی گرافیکی برای آن ایجاد کرده‌اند ولی پشت محیط سیستم دستور زبان استاندارد برای آن را ایجاد می‌کنند.
Record	<b>رکورد</b> مجموعه‌ی فیلدهای مرتبط را گویند.
Relation	<b>رابطه</b> زیر مجموعه‌ی حاصلضرب دکارتی مجموعه‌ها است. مفاهیم مرتبط با نظریه مجموعه‌ها به ترتیب عبارت است از : عضو ، مجموعه ، حاصلضرب دکارتی ، رابطه و تابع.
Relational Database	<b>بانک داده‌ی رابطه‌ای</b> سیستم مدیریت بانک داده است که از مدل رابطه‌ای که بر مفهوم ریاضی مجموعه‌ها بنا شده است ، پشتیبانی کند.
Relationship	<b>ارتباط</b> ارتباط بین موجودیت‌ها است. هر ارتباط دارای مفهوم و عملکرد است.
Report	<b>گزارش</b> یک فرم بیان اطلاعات است در قالبی مشخص و گویا که بر روی صفحه نمایش یا چاپگر نمایش داده می‌شود و چاپ می‌شود.
Table	<b>جدول</b> جدول محلی است که داده‌ها ذخیره می‌شوند و نشان‌دهنده‌ی یک موجودیت یا ارتباط بین موجودیت‌ها است.

## مرجع سریع Access