هدف کلی

آشنایی با مفاهیم بانک اطلاعاتی و کار با DBMS متداول

فصل اول

مفاهیم بانک اطلاعاتی[1]

✓ در این فصل به معرفی تعاریف و مفاهیم اولیه‌ی بانک‌های اطلاعاتی خواهیم پرداخت. اگرچه نرم‌افزارهای کاربردی به‌سهولت 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.

هدف‌های رفتاری: در پایان این فصل انتظار می‌رود که هنرجو بتواند :

- انواع سیستم‌های ذخیره و بازیابی را توضیح دهد.
- مفهوم فیلد، رکورد، فایل را توضیح دهد.
- بانک اطلاعاتی را تعریف کند.
- سیستم مدیریت بانک اطلاعاتی و معماری آن را شرح دهد.
- مفاهیم مدل رابطه‌ای را شرح دهد.
- سیستم بانک اطلاعاتی را طراحی کند.

۱ــ۱ــ کلیات

مدیریت بانک‌های اطلاعاتی یکی از انواع سیستم‌های «ذخیره و بازیابی اطلاعات»[2] است. سیستم ذخیره و بازیابی سیستمی است که به کاربر امکان می‌دهد تا داده‌ها و اطلاعات خود را ذخیره، بازیابی و پردازش نماید.

۱ــ در این کتاب پایگاه داده‌ها، بانک اطلاعاتی و بانک داده به یک معنی استفاده شده است.

۲ــ Information Storage and Retrieval

نظر به تعریف عام بالا، موارد زیر قابل ذکر است :

۱ــ کاربر می‌تواند دارای قابلیت «برنامه‌سازی» یا فاقد این مهارت باشد.

۲ــ داده می‌تواند متن، تصویر، صوت و نظایر آن باشد.

سیستم ذخیره و بازیابی اطلاعات با گذشت زمان، از مفهوم ساده‌ی فایلینگ[۱] به‌صورت زیر توسعه پیدا کرده است :

۱ــ سیستم فایلینگ Filing System (FS)

۲ــ سیستم مدیریت داده‌ها Data Management System (DMS)

۳ــ سیستم مدیریت پایگاه داده‌ها Database Management System (DBMS)

۴ــ سیستم مدیریت پایگاه دانش Knowledge Base Management System (KBMS)

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.

۲ــ۱ــ مفاهیم مقدماتی در ذخیره و بازیابی

۱ــ۲ــ۱ــ فیلد[۲]: یک قطعه داده است[۳].

هر فیلد دارای دو جزء می‌باشد :

۱ــ نام

۲ــ مقدار

مقدار حکم داده[۴] را دارد و نام فیلد به همراه مقدار در حکم اطلاع[۵] خواهد بود. به‌عنوان **مثال:**

$$\underbrace{\text{نام‌خانوادگی}}_{\text{نام فیلد}} = \underbrace{\text{علوی}}_{\text{مقدار فیلد}}$$

درواقع هر فیلد یک ویژگی یا صفت[۶] به‌حساب می‌آید. توجه کنید که از نام فیلد و مقدار فیلد به همراه هم اطلاع پیدا می‌کنیم که نام‌خانوادگی فرد موردنظر علوی است. مثال جالب‌تر این‌که : مقدار عددی «۸۸۹۶۵۴۲۳» هیچ معنی خاصی ندارد. حال به این عبارت توجه کنید :

۱ــ Filing ۲ــ Field

۳ــ فیلد واحد معنایی داده است. (Semantic Data Unit)

۴ــ Data ۵ــ Information ۶ــ Attribute

«تلفن = ۸۸۹۶۵۴۲۳»، با درج نام فیلد در کنار مقدار عددی، شما اطلاع پیدا می‌کنید که آن عدد شماره تلفن است. اگر به‌جای 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 ، Real یا Text و....

۲ــ۲ــ۱ــ رکورد[1]: به مجموعه‌ی فیلدهای مرتبط، یک رکورد می‌گوییم.

به‌عنوان مثال، قالب رکورد مربوط به یک دانش‌آموز می‌تواند شامل نام فیلدهای زیر باشد :

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

محتوای رکورد دانش‌آموزان می‌تواند برای دو دانش‌آموز همکلاسی به‌صورت زیر باشد :

۱ــ مجید، شجاعی، ۱۳۷۰، اول دبیرستان، کلاس 1 الف و...

۲ــ حسن، کمالیان، ۱۳۷۱، اول دبیرستان، کلاس 1 الف و...

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

۳ــ۲ــ۱ــ فایل[2]: مجموعه‌ای از رکوردها، تشکیل فایل را می‌دهند. به‌عنوان مثال : فایل تحصیلی دانش‌آموزان، محتوای فیلدهای رکوردهای مربوط به دانش‌آموزان را دارا می‌باشد.

۴ــ۲ــ۱ــ کلید[3]: به فیلد یا زیرمجموعه‌ای از فیلدها، که باعث یکتایی هریک از رکوردها شود، کلید گویند. به‌وسیله‌ی کلید می‌توان به یک رکورد مشخص دسترسی داشت. به‌عنوان مثال، به رکوردهای زیر توجه کنید :

ردیف	شماره دانش‌آموزی	نام	نام‌خانوادگی
۱	۱۰۰	علی	حسینی
۲	۱۱۰	حسن	احمدی
۳	۱۰۸	علی	علوی
۴	۹۴	رضا	علوی

۱ــ Record ۲ــ File ۳ــ Key

فیلد نام نمی‌تواند کلید باشد زیرا دو مقدار مشابه «علی» در رکوردهای ردیف ۱ و ۳ وجود دارد. هم‌چنین فیلد نام‌خانوادگی نمی‌تواند کلید باشد زیرا مقادیر «علوی» در رکوردهای ردیف ۳ و ۴ 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 نام‌خانوادگی نیز با هم می‌توانند کلید باشند. اگرچه در یک محیط عملیاتی مدرسه‌ی واقعی، ممکن است دو دانش‌آموز نام و نام‌خانوادگی یکسان داشته باشند.

نکته: شماره دانش‌آموزی یک صفت مجازی است که به دانش‌آموزان به‌صورت منحصر به‌فرد اختصاص می‌یابد و نقش کلید را دارد.

کنجکاوی

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

در انتخاب کلید موارد زیر قابل توجه است:

۱ــ کلید می‌تواند ساده یا مرکب باشد.

۲ــ ممکن است کلیدهای مختلفی اعم از ساده یا مرکب بتوان تشخیص داد ولی یکی از آن‌ها را با توجه به ملاحظاتی به‌عنوان کلید اصلی[۱] درنظر می‌گیریم.

۳ــ در برخی موارد به‌جای استفاده از فیلدها و صفات مورد نیاز شناسایی شده، به‌صورت مجازی فیلدی با مقدار یکتا، به‌عنوان کلید اضافه می‌شود. مانند شماره دانش‌آموزی. مجازی بودن یعنی یک فرد زمانی که در یک مدرسه ثبت‌نام می‌کند و دانش‌آموز آن مدرسه می‌شود، یک شماره‌ی دانش‌آموزی می‌گیرد و زمانی که فارغ‌التحصیل می‌شود دیگر این شماره را از دست می‌دهد.

۴ــ مقدار فیلدی که کلید درنظر گرفته می‌شود، تهی (خالی) نیست و طبق تعریف کلید، مقدار تکراری ندارد.

۱ــ Primary Key

۳ــ۱ــ عملیات روی رکوردها

این عملیات عبارت است از :

۱ــ بازیابی رکوردها	Retrieve
۲ــ درج رکوردها	Insert
۳ــ حذف رکوردها	Delete
۴ــ به هنگام سازی رکوردها	Update
۵ــ تغییر ساختار	Restructure

۱ــ۳ــ۱ــ بازیابی رکوردها: منظور از بازیابی به‌دست آوردن رکورد یا رکوردهایی است که دارای شرایط خاصی هستند. به‌عنوان مثال می‌خواهیم اطلاعات دانش‌آموزانی که معدل آن‌ها بیش از ۱۶ است را به‌دست آوریم. به این کار بازیابی گفته می‌شود. ممکن است نتیجه‌ی بازیابی یک یا چند رکورد باشد.

به‌طور کلی برای دستیابی و بازیابی رکوردها دو روش وجود دارد :

۱ــ ترتیبی

۲ــ مستقیم

در روش ترتیبی، رکوردها از ابتدا یکی پس از دیگری بررسی می‌شوند تا «محتوای» موردنظر بازیافت شود. مثلاً فرض کنید می‌خواهیم منزل شخصی به‌نام علی علوی را در یک ساختمان چند طبقه پیدا کنیم. برای یافتن آپارتمان موردنظر باید مثلاً از طبقه‌ی اول، زنگ واحدها را پشت سر هم بزنیم تا به منزل این شخص برسیم. این روش کُند و زمان‌بر است.

در روش مستقیم با توجه به محتوا، «آدرس» رکورد به‌دست می‌آید و رکورد موردنظر به‌طور مستقیم با آن آدرس بازیابی می‌شود. روش مستقیم به‌وسیله‌ی ساختارهای مختلف پیاده‌سازی می‌شود مثل شاخص[۱]. در مثال بالا اگر به دفتر سرایدار آپارتمان مراجعه کنیم، وی می‌گوید منزل این شخص مثلاً طبقه‌ی چهارم، واحد ۲ است. ما می‌توانیم از آسانسور استفاده کنیم و سریع به منزل این شخص برسیم. حتی اگر آسانسور در دسترس نباشد از راه‌پله‌ها خود را به طبقه چهارم می‌رسانیم بدون آن که در طبقه‌ها وقت صرف جستجو کنیم[۲].

۱ــ Index

۲ــ اگرچه از نظر زمانی نسبت به روش ترتیبی سریع‌تر است اما مصرف حافظه در آن بیشتر خواهد بود.

۲ــ۳ــ۱ــ درج رکوردها: منظور، افزودن رکوردی است که تاکنون در سیستم وجود نداشته است. درج می‌تواند در انتهای فایل یا محل منطقی آن رکورد صورت گیرد. محل منطقی، محلی است که نظم فایل را حفظ می‌کند.

به‌عنوان مثال فرض کنید، فایلی به‌صورت زیر وجود دارد. اگر بخواهیم دانش‌آموزی با معدل ۱۵ را اضافه کنیم و نظم فایل که به‌صورت صعودی مقادیر معدل مرتب است حفظ شود، باید این رکورد بین رکورد اول و دوم قرار گیرد. یعنی رکوردی بین اول و دوم اضافه کنیم و اطلاعات دانش‌آموز جدید را در آن ثبت کنیم.

	سایر فیلدها ...	فیلد معدل
رکورد اول		۱۴
رکورد دوم		۱۵
رکورد سوم		۱۷
رکورد چهارم		۱۸
رکورد پنجم		۲۰

درج رکوردی با معدل ۱۵ ← در محل منطقی

	سایر فیلدها ...	فیلد معدل
رکورد اول		۱۴
رکورد دوم		۱۷
رکورد سوم		۱۸
رکورد چهارم		۲۰

شکل ۲ــ۱ــ درج در محل منطقی (نظم صعودی فیلد معدل حفظ شده است)

۳ــ۳ــ۱ــ حذف رکوردها: درصورتی که به رکورد یا رکوردهایی نیاز نداشته باشیم، آن را از فایل مربوطه حذف می‌کنیم.

۴ــ۳ــ۱ــ به هنگام‌سازی رکوردها: اصلاح مقادیر فیلد یا فیلدهایی از رکورد، در اصطلاح به‌هنگام‌سازی نام دارد. به‌عنوان مثال تغییر نمره‌ی یک درس دانش‌آموز به‌هنگام‌سازی محسوب می‌شود.

۵ــ۳ــ۱ــ تغییر ساختار: اگر به هر دلیل در ساختار رکوردها تغییر ایجاد شود مثلاً اندازه‌ی فیلد تغییر کند یا فیلد جدیدی اضافه شود، تغییر ساختار صورت می‌گیرد.

۴ــ۱ــ سیستم فایلینگ

همان‌طورکه گفتیم سیستم فایلینگ، اولین نوع سیستم ذخیره و بازیابی اطلاعات است. در این سیستم نرم‌افزارهای 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 به‌وسیله‌ی برنامه‌سازی و پیاده‌سازی الگوریتم‌ها انجام می‌شود. زمانی که سیستم بزرگ و پیچیده می‌شود، هر تغییر جزیی در خواسته‌ها، منجر به کدنویسی می‌گردد.

به علت وجود فایل‌های مختلف، کنترل رعایت استاندارد در زمان پیاده‌سازی سخت است. به‌عنوان مثال فیلد نام‌خانوادگی در دو فایل ممکن است اندازه‌های مختلف داشته باشد. در سیستم فایلینگ، ممکن است «ناسازگاری داده‌ها» به‌وجود آید. درنظر بگیرید شخصی در زیر سیستم اول خود را «علی علوی تهرانی» و در زیر سیستم دوم «علی علوی» معرفی کند. بدین‌ترتیب مثلاً در مقایسه‌ی دو زیرسیستم، این دو مقدار یکسان نیستند و این به معنی ناسازگاری داده است. هم‌چنین احتمال بروز افزونگی[1] نیز وجود دارد. یعنی محتویات بعضی از فیلدها به‌صورت 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، می‌تواند ناشی از طراحی نامناسب باشد که در سایر سیستم‌های ذخیره و بازیابی نظیر بانک اطلاعاتی 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 به‌وسیله‌ی سایر سیستم‌های فایلینگ نیستند.

به‌طورکلی معایب روش فایلینگ را می‌توان به‌صورت زیر خلاصه کرد :

۱ــ احتمال بروز افزونگی (تکرار در ذخیره‌سازی)

۲ــ حجم زیاد کدنویسی

۳ــ احتمال بالای ناسازگاری داده‌ها

۴ــ سخت بودن اعمال استاندارد در سیستم‌ها

۵ــ۱ــ پایگاه داده‌ها

سیستم مدیریت پایگاه داده، یکی از سیستم‌های توسعه‌یافته‌ی ذخیره و بازیابی اطلاعات است. اگرچه این مفهوم بسیار شناخته شده و رایج است، اما واقعیت آن است که تعریف کاملاً واحدی در

۱ــ Redundancy

متون معتبر برای آن ذکر نشده است.

در نگرش بانک اطلاعاتی، طراحی به‌صورت یکپارچه انجام می‌گیرد. از آن‌جایی که یک محیط عملیاتی می‌تواند دارای زیرمحیط‌های مختلفی باشد، نگرش هنرجو باعث می‌شود برخی معایب روش فایلینگ نظیر ناسازگاری داده‌ها و اعمال نشدن استانداردها رفع گردد، مثلاً برای فیلد نام در تمام محیط، مقدار یکسانی از حافظه اختصاص می‌یابد. حال آن‌که در روش فایلینگ به علت مجزا بودن فایل‌ها به سختی و با صرف هزینه‌ی بالا می‌توان نگرش هنرجو داشت.

یکپارچه‌سازی و مدیریت متمرکز باعث جلوگیری از تکرار در ذخیره‌سازی (افزونگی) خواهد شد و درنتیجه از حافظه، استفاده بهتری صورت می‌گیرد. در مجموع می‌توان پایگاه داده را به‌صورت زیر تعریف کرد :

«مجموعه‌ای از داده‌ها به‌صورت یکپارچه با حداقل افزونگی، تحت کنترل یک سیستم متمرکز و در چارچوب یک مدل داده‌ای»

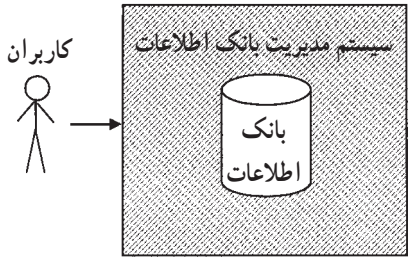
با ترکیب دست‌آوردهای شبکه‌ای، پایگاه داده می‌تواند به‌وسیله‌ی چند کاربر و همزمان استفاده شود.

کنجکاوی

تعریف‌های مختلف پایگاه داده را بررسی کنید.

۶ــ۱ــ سیستم مدیریت پایگاه داده[۱]

اصلی‌ترین تفاوت روش پایگاه داده با روش‌های قبلی مثل سیستم فایلینگ، وجود حصاری به‌نام «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 تأیید، کار خواسته شده انجام می‌گیرد.

۱ــ Database Management System (DBMS)

۱ــ۶ــ۱ــ معماری پایگاه داده: سؤال این است که چگونه می‌توان در یک سیستم بزرگ و پیچیده‌ی 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.

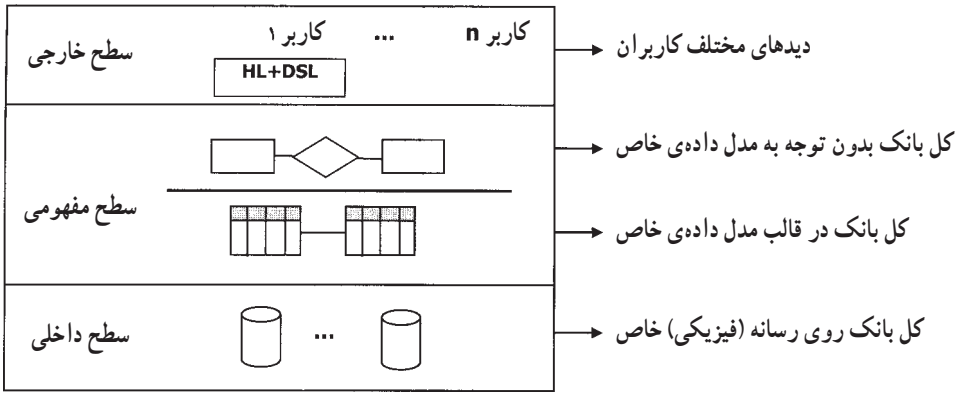
پاسخ این سؤال در معماری سه لایه‌ای است که به‌وسیله‌ی کمیته ANSI/SPARC[1] عرضه شده است. در ایجاد این معماری سطوح به‌گونه‌ای از هم مجزا شده‌اند که برای رسیدن به هدف با هم به‌خوبی مراوده و تعامل کنند و هم به اندازه‌ی کافی مجزا و مستقل باشند که تغییر در یک سطح به سطوح دیگر تسری پیدا نکند. این سه لایه عبارت است از :

۱ــ سطح خارجی[2]　　۲ــ سطح مفهومی[3]　　۳ــ سطح داخلی[4]

● ***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.

● ***سطح مفهومی:*** ساختار پایگاه داده است. تمام موجودیت‌ها و ارتباطات بین آن‌ها را شامل می‌شود.

● ***سطح داخلی:*** 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 داخلی، سطح فایلینگ است.



ساختار سه لایه‌ای معماری پایگاه داده

۱ــ ANSI Standard Planning and Requirements Committee

۲ــ External Level　　۳ــ Conceptual Level　　۴ــ Internal Level

نکته:

HL: زبان برنامه‌نویسی غیربانکی است. مثل Visual Basic و...

DSL: زبان برنامه‌نویسی بانکی یا به عبارتی زبان فرعی داده‌ای است. مثل SQL و...

۲ــ۶ــ۱ــ 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 :

۱ــ رابطه‌ای Relational Data Structure

۲ــ سلسله مراتبی Hierarchical Data Structure

۳ــ شبکه‌ای Network Data Structure

کنجکاوی

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 ارایه 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، ماجول‌ها و... برای این منظور ایجاد شده است که در بخش‌های بعدی بیشتر با آن آشنا می‌شویم.

۷ــ۱ــ مفاهیم پایه‌ی پایگاه داده‌ی رابطه‌ای

۱ــ۷ــ۱ــ محیط عملیاتی[1] و تعریف موجودیت: محیط عملیاتی، محیطی است که می‌خواهیم یک سیستم ذخیره و بازیابی برای آن ایجاد کنیم. مانند : مدرسه یا دانشگاه، بانک، تعمیرگاه، فروشگاه و...

در هر محیط عملیاتی تعدادی موجودیت[2] وجود دارد. موجودیت، هر مفهوم و یا شیء در محیط عملیاتی است. مثلاً دانش‌آموز، معلم، کارمند مدرسه، کلاس و درس موجودیت‌های محیط عملیاتی مدرسه هستند. تمام موجودیت‌های یک محیط عملیاتی مهم نیستند بلکه موجودیت‌هایی برای ما مهم هستند که می‌خواهیم در مورد آن‌ها اطلاع داشته باشیم. به‌عنوان یک مثال ساده در محیط عملیاتی مدرسه موجودیت‌های دانش‌آموز، معلم و درس برای سیستم فرضی ثبت‌نام مهم و قابل‌توجه هستند.

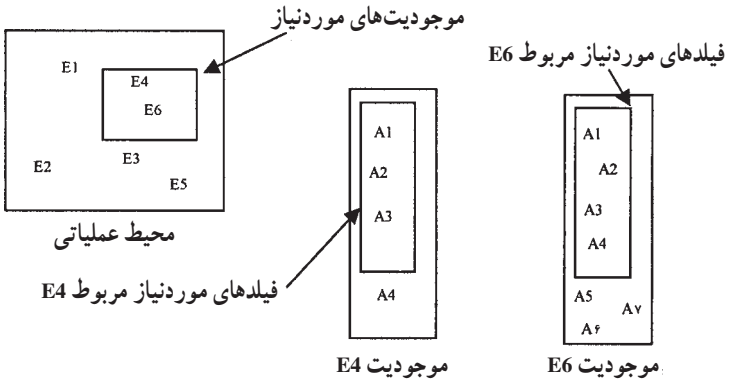
در هر محیط عملیاتی مجموعه‌ای از موجودیت‌های مناسب و مرتبط براساس مرز و محدوده‌ی سیستم مورد طراحی، انتخاب می‌شوند.

تمرین ۱ــ۱: در محیط‌های عملیاتی بانک، خانواده و باشگاه ورزشی چه موجودیت‌هایی قابل تشخیص است؟

پرسش: در چه مواقعی از فیلدهای خود موجودیت برای تعریف کلید استفاده نمی‌شود؟

۲ــ۷ــ۱ــ ویژگی‌های هر موجودیت: صفات و ویژگی‌های هر موجودیت درواقع همان فیلدها هستند. در مورد انتخاب فیلدهای یک موجودیت نیز مانند انتخاب موجودیت‌های محیط عملیاتی فقط صفات مورد نیاز و مرتبط به‌عنوان فیلد درنظر گرفته می‌شوند و تمام صفات یک موجودیت درنظر گرفته نمی‌شوند. به‌عنوان مثال، در محیط عملیاتی مدرسه در مورد اطلاعات مربوط به دانش‌آموز ویژگی‌هایی نظیر قد، وزن، رنگ چشم درنظر گرفته نمی‌شود چرا که در محیط عملیاتی موردنظر به آن‌ها نیاز نیست، ولی ممکن است در محیط عملیاتی دیگر مناسب و مفید باشند.

۱ــ Operational Environment ۲ــ Entity



شکل ۱ــ۱ــ زیرمجموعه موجودیت‌ها و فیلدهای انتخاب شده برحسب نیاز

کنجکاوی

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

تمرین ۲ـ۱: در سه محیط عملیاتی تمرین ۱ــ۱ موجودیت‌های مرتبط به همراه فیلدهای موردنیاز را شناسایی کنید.

مثال ۱ـ۱

محیط عملیاتی مدرسه را درنظر 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.

۳ــ۷ــ۱ــ ارتباط[1]: پیش از این عنوان شد که در محیط عملیاتی، موجودیت‌هایی وجود دارند. معمولاً بین موجودیت‌ها ارتباط یا ارتباط‌هایی وجود دارد که قابل شناسایی است. در محیط عملیاتی مدرسه، بین موجودیت‌های درس، دانش‌آموز و معلم ارتباط‌های زیر قابل تعریف است :

۱ــ معلم درس ارایه می‌کند.

۲ــ دانش‌آموز هر ترم چندین درس ثبت‌نام می‌کند.

۳ــ دانش‌آموز برای هریک از درس‌های ثبت‌نام شده نمره می‌گیرد.

همان‌طورکه ملاحظه می‌شود :

- در هر ارتباط موجودیت‌هایی شرکت دارند.
- هر ارتباط دارای یک مفهوم یا عملکرد است. مثلاً در ارتباط اول «ارایه کردن» عملکرد می‌باشد.
- هر ارتباط دارای «ماهیتی» خاص است. مثلاً در ارتباط دوم ماهیت ارتباط می‌تواند یک به چند باشد. یعنی یک دانش‌آموز چند درس اخذ می‌کند.

کنجکاوی

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

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

۴ــ۷ــ۱ــ انواع ماهیت در ارتباط[2]: تناظر بین موجودیت‌ها، ماهیت ارتباط است و به یکی از سه نوع زیر وجود دارد :

۱ــ یک به یک 1:1

۲ــ یک به چند 1:N

۳ــ چند به چند M:N

در ارتباط یک به یک، یک نمونه از موجودیت اول فقط با یک نمونه از موجودیت دیگر

۱ــ Relation ۲ــ Cardinality

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

تمرین ۳ـ۱ : مثالی از حالت چند به چند ارایه کنید.

کنجکاوی

حالتی را مثال بزنید که موجودیت با خودش مرتبط باشد.

مثال ۲ـ۱

در محیط عملیاتی مدرسه، ماهیت ارتباط‌ها به‌صورت زیر است :

۱ـ یک معلم چند درس ارایه می‌کند و هر درس به‌وسیله‌ی یک معلم ارایه می‌شود. (n : ۱)

۲ـ دانش‌آموز (در هر ترم) چندین درس ثبت‌نام می‌کند و یک درس به‌وسیله‌ی چندین دانش‌آموز انتخاب می‌شود. (m:n)

۳ـ یک دانش‌آموز برای هر درس ثبت‌نام شده، یک نمره می‌گیرد. یک نمره به یک درس داده می‌شود. (n : ۱)

۵ـ۷ـ۱ـ نمودار ارتباط ـ موجودیت /ER[1]: 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 :

۱ـ موجودیت‌ها

۲ـ عملکرد ارتباط

۱ـ Entity Relationship Diagram

۳ــ ماهیت ارتباط

۴ــ ویژگی‌های موجودیت

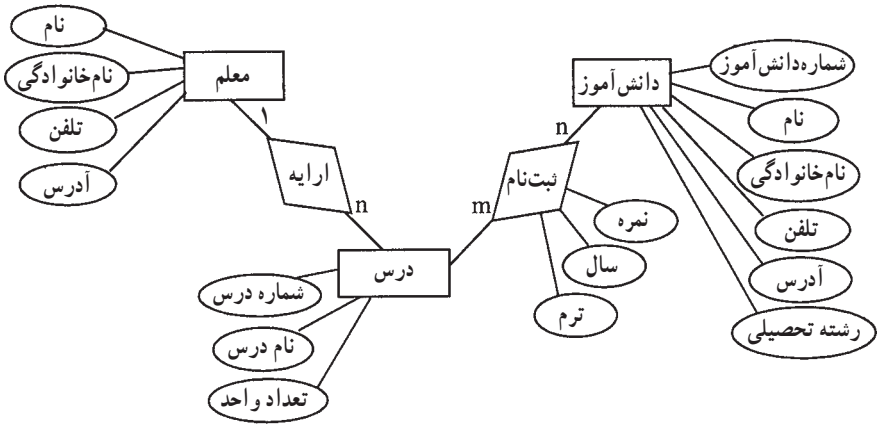
برای ترسیم نمودار، از شکل‌هایی به‌صورت زیر استفاده می‌شود[۱] :

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.

مثال۳ــ۱

نمودار ER محیط عملیاتی مدرسه به‌صورت زیر است :



۱ــ شکل‌ها به‌صورت قراردادی است و در منابع مختلف متفاوت است.

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

۶ــ۷ــ۱ــ رابطه / جدول: مدل رابطه‌ای بر مبنای یک مفهوم ریاضی به‌نام «رابطه»[1] تعریف شده است. استحکام نظریه مجموعه‌ها و سهولت پیاده‌سازی آن باعث رواج این مدل داده‌ای شده 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 ۱ــ۱ را دوباره ملاحظه نمایید. طبق مدل داده‌ای رابطه‌ای جدول‌های زیر را می‌توان درنظر گرفت :

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

Course:

شماره‌ی درس	نام درس	تعداد واحد

Student:

شماره‌ی دانش‌آموزی	نام	نام‌خانوادگی	تلفن	نشانی

Teacher:

آدرس	تلفن	نام‌خانوادگی	نام

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

۱ــ رابطه، زیرمجموعه‌ای از حاصل‌ضرب دکارتی مجموعه‌ها است. مفاهیم مرتبط در این نظریه به‌ترتیب عضو، مجموعه حاصل‌ضرب دکارتی، رابطه و تابع است.

نکته: محاسن مدل داده‌ای رابطه‌ای :

۱ــ از نظر نمایش، ساده و قابل فهم است و تنها از یک مفهوم اصلی به‌نام رابطه یا جدول استفاده می‌کند.

۲ــ انواع ماهیت‌های ارتباط را پشتیبانی می‌کند.

۳ــ مبنای ریاضی دارد.

هنگامی که طراح، موجودیت‌های مورد نیاز 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.

۷ــ۷ــ۱ــ تبدیل ER به جدول

ــ هر موجودیت مستقل به یک جدول تبدیل می‌شود.

ــ صفات موجودیت‌ها به فیلدهای جدول مربوطه تبدیل می‌شوند.

ــ موجودیت‌هایی که ارتباط ۱ :۱ دارند به یک جدول تبدیل می‌شوند. [اگرچه ممکن است تداخل اطلاعاتی رخ دهد]. در مواردی نیز مانند ۱ :n یکی از جدول‌ها به دیگری منتقل می‌شود.

ــ در ارتباط ۱ :n، کلید اصلی جدول طرف ۱ به جدول طرف n اضافه می‌شود.

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

۸ــ۱ــ 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 ــ 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، معمولاً ابتدا تحلیل فرایندی صورت می‌گیرد و پردازش‌ها و گردش اطلاعات شناسایی می‌شوند. با مشخص شدن پردازش‌ها، داده‌هایی که از طریق فرم‌ها رد و بدل 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 مورد نیاز 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 قبلاً توضیح داده‌ایم.

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 :

کنجکاوی

یک روش و متدولوژی طراحی نام برده و خصوصیات آن را به‌طور 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 محصولات 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. به‌طورکلی سه روش وجود دارد :

فرایندگرا (Process Oreinted)، داده‌گرا (Data Oriented) و شیءگرا (Object Orienter). روش فرایندگرا مانند SSADM، Jackson، Urdon و غیره. داده‌گرا مانند روش Information Engineering (IE) و روش شیءگرا مانند RUP (UML).

۱ــ 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 مورد نیاز، اطلاعات جمع‌آوری کنید.

۲ــ نمودار ER را طراحی کنید.

موجودیت‌ها، ویژگی‌ها و ارتباط بین آن‌ها را شناسایی کنید و نمودار ارتباط ــ موجودیت‌ها (ER) را ترسیم کنید.

برای هر موجودیت کلید اصلی را تعیین یا ایجاد کنید.

> **نکته:** به‌خاطر بسپارید که یک موجودیت نباید اطلاعات افزونه داشته باشد یعنی هر موجودیت باید اطلاعات یک موضوع را نگهداری کند و از اختلاط اطلاعاتی پرهیز شود.

۳ــ جدول‌ها و فیلدهای آن‌ها را مشخص کنید.

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

۴ــ فرم‌های مورد نیاز را تعریف کنید.

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

۵ــ گزارش‌های مورد نیاز 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.

۶ــ 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.

کنجکاوی

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 مسأله نشان‌دهنده‌ی غلط بودن طراحی نیست. بلکه برحسب مفروضات، دقت کار و محدوده‌ی سیستم کاملاً طبیعی است.

مثال ٤ـ١

محیط عملیاتی مدرسه را طراحی کنید.

نمودار ER رسم شده است.

١ـ نمودار ER به جدول‌های زیر تبدیل می‌شود :

جدول معلم	کد معلم	نام	نام‌خانوادگی	تلفن	آدرس
Tbl_Teacher	Teacher Code	Teacher FName	Teacher LName	Teacher Tel	Teacher Address

* در جدول دانش‌آموز، فیلد کد معلم را با عنوان کلید اصلی اضافه می‌کنیم.

جدول‌دانش‌آموز	شماره‌ی دانش‌آموز	نام	نام‌خانوادگی	تلفن	آدرس	سال
Tbl_Student	Student Code	Student FName	Student LName	Student Tel	Student Address	Year

جدول درس	شماره‌ی درس	نام درس	تعداد واحد	(کد معلم)
Tbl_Course	Course Code	Course Name	Course Unit	Teacher Code

* ارتباط ١ : n بین درس و معلم، کلید جدول معلم (کد معلم) به جدول درس اضافه می‌شود. دقت کنید که کد معلم در جدول درس بخشی از کلید اصلی نیست.

جدول ثبت‌نام	سال	ترم	شماره‌ی درس	شماره‌ی دانش‌آموز	نمره‌ی درس
Tbl_Register	Year	Term	Course Code	Student Code	Course Grade

* ارتباط n:m بین دانش‌آموز و درس به جدول ثبت‌نام تبدیل می‌شود و کلید اصلی جدول درس (شماره‌ی درس) و کلید اصلی جدول دانش‌آموز (شماره‌ی دانش‌آموز) به آن اضافه می‌شود.

چون ارتباط n:m مفروض خود دارای ویژگی‌هایی است پس این دو کلید به‌تنهایی 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 نرمال‌سازی خواهیم دید، جدول ثبت‌نام به دو جدول زیر تبدیل می‌شود :

جدول ثبت‌نام	سال	ترم	شماره‌ی دانش‌آموز	سریال ثبت‌نام
Tbl_Register	Register Year	Term	Student Code	Register Code

* ما شماره‌ی سریال ثبت‌نام را به‌عنوان کلید اصلی اضافه می‌کنیم. می‌توانستیم سال و ترم را هم کلید بگیریم ولی بهتر است از کلید ترکیبی به علت پیچیدگی‌هایی که ایجاد می‌کنند استفاده نکنیم.

جدول دروس ثبت‌نام شده	سریال ثبت‌نام	شماره‌ی درس	نمره‌ی درس
Tbl_Register	Register Code	Course Code	Greade

* دقت کنید که در جدول دروس ثبت‌نام شده شماره‌ی درس بخشی از کلید اصلی است.

شکل [ ] نشان‌دهنده‌ی کلید اصلی و شکل ( ) نشان‌دهنده‌ی کلید خارجی اضافه شده از جدول دارای ارتباط است.

۲ــ فرم‌های مورد نیاز :

فرم تعریف معلم

کد معلم:
نام:
نام‌خانوادگی:
تلفن:
آدرس:
توضیحات:

فرم تعریف درس

شماره‌ی درس:
نام درس:
تعداد واحد:
نام معلم:

فرم تعریف دانش‌آموزان

شماره‌ی دانش‌آموز:
نام:
نام‌خانوادگی
تلفن:
آدرس:
رشته‌ی تحصیلی:

فرم ورود نمره

سال: ترم: نام دانش‌آموز:

شماره‌ی درس	نام درس	تعداد واحد	نمره‌ی نهایی

فرم ثبت‌نام

سال: ترم: نام دانش‌آموز:

شماره‌ی درس	نام درس	تعداد واحد

۳ــ گزارش‌های مورد نیاز :

گزارش لیست معلمان مدرسه

کد معلم	نام معلم	تلفن	آدرس

گزارش ثبت‌نام

سال: ترم:

نام دانش‌آموز: شماره دانش‌آموز:

دروس ثبت‌نام شده

شماره‌ی درس	نام درس	تعداد واحد	نام معلم

گزارش کارنامه دانش‌آموزان

نام و نام‌خانوادگی:

تاریخ گزارش:

رشته: سال: ترم:

شماره‌ی درس	نام درس	تعدادواحد	نمره

جمع واحد:

معدل:

گزارش لیست حضور و غیاب

سال: ترم:

نام معلم: درس:

تعداد واحد: ساعت تشکیل کلاس:

شماره‌ی دانش‌آموز	نام دانش‌آموز									

خلاصه‌ی فصل

سیستم ذخیره و بازیابی از مفهوم ساده‌ی فایلینگ توسعه پیدا کرده است. فیلد کوچک‌ترین واحد ذخیره‌ی داده است. رکورد مجموعه‌ی فیلدهای مرتبط است و مجموعه‌ی 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.

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 برقرار کردن ارتباط بین حجم بالای داده‌های ذخیره شده‌ی روی رسانه و کاربران معماری سه لایه‌ای مطرح شده است : سطح خارجی، سطح مفهومی، سطح داخلی.

نرم‌افزاری 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 به‌نام 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 ــ موجودیت (ER) و روش تبدیل این نمودار به جدول‌های پایگاه داده مطرح شده است.

خودآزمایی

۱ــ دلایل متغیر شدن طول رکورد چیست؟

۲ــ معایب روش فایلینگ چیست؟

۳ــ کدام‌یک از اجزای محیط ذخیره و بازیابی بیشتر اهمیت دارد؟ چرا؟

۴ــ هدف از ایجاد سیستم ذخیره و بازیابی چیست؟

۵ــ کلید را تعریف کنید و انواع آن را با ذکر مثال توضیح دهید.

۶ــ معماری پایگاه داده را توضیح دهید.

۷ــ مدل داده‌ای را تعریف کنید و توضیح دهید.

۸ــ ماهیت ارتباط را تعریف کنید و در یک محیط عملیاتی مثال بزنید.

۹ــ مراحل طراحی پایگاه داده را توضیح دهید.

۱۰ــ لزوم طراحی قبل از پیاده‌سازی و شروع کار با نرم‌افزار Access را بیان کنید و مزایا و معایب احتمالی آن را بررسی کنید.