به نام خدا

پایگاه داده ها

مروری بر ویژگی های نسخه ۵,۰,۱ بانک اطلاعاتی MySQL

محمد ممبینی

Mombaini2001@Yahoo.Com
www.mombaini2006.blogsky.com

بانک اطلاعاتی معروف MySQL که اپن‌سورس است، طرفداران زیادی در بین برنامه‌نویسان دارد. MySQL علاوه بر آن‌که یکی از نمونه‌های بسیار موفق نرم‌افزارهای منبع باز محسوب می‌شود، مثالی از نرم‌افزاری است که در اروپا (سوئد) پا به عرصه گذاشته است. نخستین نسخه این بانک اطلاعاتی توسط دو نفر از برنامه‌نویسان سوئدی نوشته شد و از آن پس بر اساس مدل نرم‌افزارهای منبع باز توسعه یافت. البته در میانه راه (یعنی در زمان عرضه نسخه سوم این نرم‌افزار) بر اساس توافقی میان شرکت MySQL AB از یک سو و شرکت معظم SAP (که دارای خط تحقیق و توسعه بانک اطلاعاتی اختصاصی خود بود) از طرف دیگر، نسخه ویژه‌ای موسوم به MaxDB از تلفیق تکنولوژی‌های این دو شرکت در فهرست محصولات MySQL قرار گرفت. اما شرکت MySQL همچنان کار توسعه نسخه قبلی و منبع باز نرم‌افزار MySQL را تا این لحظه به موازات MaxDB ادامه داده است.

طرف‌داران فلسفه نرم‌افزارهای منبع باز همواره از MySQL به عنوان نمونه‌ای یاد می‌کنند که توانسته است به‌خوبی با نمونه‌های بانک‌های اطلاعاتی اختصاصی همانندSQL server که سرمایه‌گذاری‌های سنگینی برای توسعه آن انجام می‌شود، رقابت کند (هر چند که مایکروسافت بر اساس یک سیاست کلی تنها رقیب خود را در زمینه بانک‌های اطلاعاتی، محصولات اوراکل می‌داند و اصولاً توجهی به نرم‌افزارهای منبع باز نمی‌کند). اما واقعیت آن است که در دنیای برنامه‌نویسی این دو نرم افزار در رقابت تنگاتنگ قرار دارند.

آخرین نسخه MySQL همراه مجموعه‌ای از قابلیت‌ها و امکانات عرضه شده است که آنرا بیش از پیش به‌سمت کاربردهای <بزرگ مقیاس> سوق می‌دهد. برخی از ناظران معتقدند هدف اولیه تغییرات و پیشرفت‌های ارائه شده در نسخه ۵ این نرم‌افزار، بیش از هر چیز دیگری معطوف برنامه‌نویسی پیشرفته در کاربردهای گسترده و بزرگ است. یکی از منابع مورداستفاده در تهیه این مقاله در این زمینه چنین بیان می‌کند: <آیا زمان آن فرا رسیده است که اوراکل باز گردد و به پشت‌سر خود نگاه کند؟> جالب است که نویسنده در نوشته خود اصلاً اسمی از مایکروسافت و SQL server نیاورده است!

بانک اطلاعاتی سرور از نوع EMBEDDED

اگرچه قابلیت استفاده از این بانک اطلاعاتی به صورت سرور Embedded موضوع جدیدی محسوب نمی‌شود و از نسخه MySQL ۴ به بعد همواره وجودداشته است، اما این ویژگی از سوی کاربران همچنان نسبتاً ناشناخته باقی‌مانده است. استفاده از موتور نرم‌افزاری این بانک اطلاعاتی به صورت Embedded با توجه به آن‌که این نرم‌افزار از نظر توابع API به‌طور کامل با مدل Client/Server سازگاری دارد، بسیار ساده است. در واقع برای به‌کارگیری این نرم‌افزار به صورت Embedded تنها کافی است تا یک تغییر کوچک در سورس کد (نسبت به روش معمول) اعمال شود.

نمونه‌ای از یک قطعه کد به زبان C (که در گوشه و کنار اینترنت به فراوانی یافت می‌شود) این مطلب را به‌خوبی نمایش می‌دهد. (به قطعه کد شماره یک مراجعه نمایید)

```
#include <mysql.h>
#include <stdlib.h>
static char *serverargs[] = {
 "argv0",
 "--datadir = /var/db/embedded"
};

int main(int argc, char** argv) {
  MYSQL* dbb;

 /*For embedded
 if (mysql_server_init(2,serverargs,NULL))
  exit(1);

 /*Normal state on local host
 db = mysql_init(NULL);
 mysql_real_connect(db,NULL, "user" , "secret" , "db" , 0 , NULL , 0);

 /*client api calls

 /*For embdded
mysql_server_end();
}
```

کد ۱

یونیون ها

یونیون‌ها موجوداتی هستند که امکان ترکیب دو یا چند Query را در یک DataSet فراهم می‌کنند (البته با فرض این که اسامی ستون، نوع داده و ترتیب فیلد مطابقت داشته باشند). یونیون‌ها مکانیسم بسیار قدرتمندی برای انواع گوناگون جستجوهای پیشرفته محسوب می‌شوند. به‌طور معمول زمانی یونیون‌ها به‌کار برنامه‌نویسان می‌آیند که در کاربردهای مورد نظر آن‌ها، دو جدول شامل اطلاعات نسبتاً مرتبط وجود داشته باشند. به عنوان مثال، در صورتی‌که در ساختار یک بانک اطلاعاتی، یک جدول شامل اطلاعات فروشندگان باشد و جدول دیگری برای ثبت داده‌های شرکت‌های فروشنده لوازم ساختمانی به‌کار رفته باشد، می‌توان از یونیون برای جستجوی همزمان در دو جدول و به‌دست آوردن یک مجموعه پاسخ یا result set بهره گرفت.

SUB QUERY ها

از SubQueryها و جداول مشتق شده برای قرار دادن عبارات انتخابی در یک Statement SQL دیگر استفاده می‌شود. مثلاً اگر در بخش FROM عبارت جستجوی خود از یک عبارت SELECT دیگر استفاده کنید، در این صورت، عبارت SELECT خارجی از نتایج به‌دست آمده از عبارت SELECT به‌کار رفته در بخش FROM کلی جستجو، استفاده خواهد کرد. این ویژگی هم یکی از ابزارهای قدرتمندی است که در کاربردهای پیشرفته به کمک برنامه‌نویسان می‌آید.

عبارات از قبل آماده

برنامه‌نویسان آشنا با گرامر عبارات ODBC ازپیش آماده (statement ODBC Prepared) از این پس می‌توانند این ویژگی را در مجموعه API بانک اطلاعاتی MySQL که به زبان C نوشته است، نیز بیابند.

مثلاً:

```
AND ? < annual_sales SELECT * FROM customer WHERE region =?
```

اصطلاح عبارت جستجوی فوق آماده یا prepare شد، برنامه‌نویس می‌تواند توسط توابع API نرم‌افزار MySQL مقادیر گوناگونی را به علامت‌های سؤال‌های به‌کار رفته در عبارت، متصل یا Bind کنند. مزیت عمده پشتیبانی از چنین قابلیتی در آن نهفته است که دیگر برای هر عبارت جستجویی که به ازای هر یک از مقادیر متغیر علامت سؤال ایجاد می‌شود، نیازی به تولید مجدد query نخواهد بود. اهمیت این موضوع زمانی بیشتر آشکار می‌گردد که قرار باشد یک عبارت جستجوی از پیش آماده مکرراً اجرا شود. در این وضعیت به‌دلیل آن‌که queryها فقط یک‌بار ساخته و بهینه‌سازی می‌شوند، سرعت اجرای نرم‌افزار به طرز محسوسی بالا خواهد بود (در واقع سرعت اجرای نرم‌افزار در قیاس با حالت معمول، دچار افت قابل توجهی نخواهد شد).

چندین DATASET در یک فراخوانی

از زمان عرضه نسخه ۴,۱ نرم‌افزار MySQL، برنامه‌نویسان می‌توانسته‌اند توسط یک فراخوانی، چندین query را بر روی سرور به اجرا بگذارند. این مطلب به معنی آن است که نرم‌افزار Client قادر به دریافت چندین resultSet خواهد بود. این قابلیت در مواقعی که برنامه‌نویس از پیش می‌داند که چندین جستجو مستقل و ناوابسته به یکدیگر باید در کاربرد مشخصی به اجرا گذاشته شوند، عامل صرفه‌جویی بسیار مفیدی محسوب می‌شود. علاوه بر مواردی که در بالا مورد بررسی قرار دادیم، این قابلیت زمانی که آن‌را در کنار ویژگی جدید نسخه ۵ این نرم‌افزار یعنی پشتیبانی از Stored Procedureها مورد بررسی قرار دهیم، اهمیت دوچندان خواهد یافت. زیرا می‌دانیم که هر Stored Procedure ممکن است منجر به تولید و بازگشت دادن چندین resultSet شود.

VIEW ها

Viewها برای نخستین بار در نسخه جدید MySQL مورد پشتیبانی قرار گرفته‌اند. Viewها این امکان را برای کاربر فراهم می‌آورند که بتواند داده‌های مجموعه‌ای از جداول را همانند اطلاعات یک جدول ببیند. واقعیت آن است که Viewها از دید برنامه‌نویسان از اهمیت چندانی برخوردار نیستند. اما برای مدیران بانک‌های اطلاعاتی، view یک ابزار کارآمد محسوب می‌شود. در آخرین نسخه بانک‌اطلاعاتی MySQL، viewها دارای قابلیت Update شدن هستند.

procedure Stored ها و توابع

طرفداران MySQL تا همین اواخر از این‌که نرم‌افزار بانک‌اطلاعاتی محبوبشان از procedure Storedها پشتیبانی نمی‌کند، دلخور می‌شدند. اما از زمان عرضه نسخه ۴، شرکت MySQL AB وعده داده بود که از این قابلیت در نسخه ۵ پشتیبانی خواهد کرد. بدین ترتیب برای نخستین بار Stored procedure ها در نسخه‌های ۵ و ۵,۰,۱۰ به کار گرفته شدند.

یک stored procedure همان‌طور که از نام آن مشخص می‌شود، دستور فرایند یا تابعی است که در محل خودِ بانک اطلاعاتی ذخیره‌سازی می‌شود. stored procedure از چندین منظر دارای اهمیت فراوان است. اصولاً یک stored procedure تابعی است که برنامه‌نویس از آن برای انجام عملیات منطقی پیچیده بر روی داده‌های بانک اطلاعاتی استفاده می‌کند. البته باید توجه داشته باشید که چنین تابعی در خود بانک اطلاعاتی ذخیره می‌شود. بدین ترتیب خواص متعددی که به آن اشاره کردیم، در این شرایط تحقق می‌یابند. نخست آن‌که در یک مدل برنامه‌های کاربردی از نوع Client/server، انجام چنین عملیات پیچیده‌ای به هیچ وجه به سکویی که بخش Client بر روی آن در حال اجرا خواهد بود، وابسته نیست. نکته دیگر آن‌که در سیستم‌های شبکه، انجام عملیات پیچیده و سنگین تحت هیچ شرایطی منجر به ایجاد ترافیک در سطح شبکه و در نتیجه ایجاد تأخیر نخواهد شد. زبان مورد استفاده برای نوشتن توابع ذخیره شده در بانک procedure Stored، زبان استانداردی است که SQL ۲۰۰۳ نام دارد. این زبان همان زبانی است که بسیاری از بانک‌های اطلاعاتی از آن برای چنین منظورهایی استفاده می‌کنند. به عنوان مثال بانک اطلاعاتی معتبر و معروف IBM به نام DB2 هم از همین زبان برای بیان روتین‌های توابع خود بهره می‌گیرد.

مطلب دیگری که نباید آن‌را فراموش کرد آن است که در نسخه آخر MySQL قابلیت مهمی گنجانده شده است که امکان بازگرداندن نتایج یک عبارت جستجوی SELECT را به‌سمت کلاینت، فراهم می‌کند. در صورتی که نرم‌افزار MySQL از چنین قابلیتی پشتیبانی نمی‌کرد و برنامه‌نویسان ناگزیر می‌شدند تا یک cursor بر روی query باز کنند و نهایتاً یک resultSet بسازند. هرچند که انجام چنین عملیاتی برای برنامه‌نویسان بانک‌های اطلاعاتی کار دشواری محسوب نمی‌شود، اما با این حال کاری است که نیاز به توجه کامل برنامه‌نویس برای برگرداندن نتیجه یک query به کاربر دارد. از آنجایی که در کاربردهای مشخصی، امکان بازگشت چندین resultSet به کاربر متصور است، اهمیت ویژگی مورد بحث در چنین شرایط خاصی بیشتر نمایان خواهد شد. بر همین اساس توصیه می‌شود در چنین شرایطی، حدالمقدور از نسخه‌های بالاتر از نسخه ۴,۱استفاده شود.

در پروژه‌های بزرگ و پیچیده‌تر،stored procedure ها نقش دیگری نیز می‌یابند. معمولاً مرسوم است که در پروژه‌های بزرگ، یک یا دو نفر از برنامه‌نویسان خبره بانک‌های اطلاعاتی وظیفه می‌یابند تا به حل مسائل پیچیده‌تر بپردازند و حاصل کار خود را به‌صورت یک عبارت SQL و در قالب یک Stored procedure تحویل دهند. بدین ترتیب مابقی برنامه‌نویسان می‌توانند به پیاده‌سازی بخش‌های دیگر پروژه بپردازند.

REPLICATION

اگرچه این ویژگی از دیرباز در نرم‌افزار MySQL مورد توجه بوده است، در نسخه جدید MySQL پیشرفت‌های قابل توجهی در ارتباط با این موضوع از جهات سرعت اجرا و قابلیت اطمینان صورت گرفته است. ویژگی replication از چندین جهت مهم تلقی می‌شود. نخست آن‌که استفاده از replication تقریباً در اکثر نرم‌افزارهای کاربردی بزرگ مقیاس (که کاربران متعددی در حال قرائت اطلاعات از جداول هستند)، روش متداولی به‌شمار می‌رود. در چنین شرایطی به‌طور معمول کاهش سرعت اجرای نرم‌افزار کاربردی، هیچ چاره‌ای به‌جز بهره جستن از منابع سخت‌افزاری سریع‌تر نخواهد داشت. اما یک راه‌حل هوشمندانه دیگر نیز برای جبران مسأله سرعت اجرا وجود دارد. در این روش می‌توان یک بانک اطلاعاتی را فقط برای خواندن از اطلاعات به‌کار گرفت و از چندین بانک‌های اطلاعاتی دیگر برای نوشتن و ذخیره اطلاعات استفاده کرد. کاربرد متداول دیگر replication، ایجاد یک پشتیبان زنده یا Hot Backup از بانک‌های اطلاعاتی به اصطلاح mission critical است. (اصطلاح mission critical به وضعیت‌هایی گفته می‌شود که در

آن‌ها حتی توقف‌های لحظه‌ای نرم‌افزار، منجر به بروز خسارت خواهد شد. به عنوان مثال، نرم‌افزارهای کنترلی دمای راکتور یک نیروگاه هسته‌ای، تشکیل چنین حالتی دارند. زیرا توقف لحظه‌ای فرایند کنترل دما در چنین کاربردهایی مسلماً بسیار خطرآفرین محسوب می‌شود) سناریوی استفاده از replication در تهیه نسخه‌های <پشتیبان زنده> در کاربردهای حساس چنان است که به محض از کارافتادگی بانک اطلاعاتی اصلی، نسخه‌های پشتیبان، قابلیت در مدار قرار دادن اطلاعات و جایگزین بانک اطلاعاتی اصلی شدن را دارا باشند. در یک طراحی مناسب، استفاده از replication می‌تواند زمان‌های توقف یا Downtime را به حداقل برساند.

نتیجه گیری

مسیر تغییر و تحولات و توسعه‌ای که نرم‌افزار MySQL از نسخه ۳ به بعد تا این لحظه طی کرده است، مسیری دشوار و طولانی محسوب می‌شود. اما تنها از طریق طی کردن چنین مسیرهایی یک نرم‌افزار، خصوصاً از نوع بانک‌های اطلاعاتی می‌تواند خود را برای کاربرد در سطوح سازمانی بزرگ (Enterprise) آماده سازد.

محمد ممبینی دانشجوی رشته کامپیوتر

Mombaini2001@yahoo.com

www.mombaini2006.blogsky.com

منبع مجله دنیای کامپیوتر