

شرکت مبین سامان

گزارش پروژه

پایگاه داده

تشریح و تشخیص خطاهای نوع داده

نگارنده:

محمدسعید رجبی

بهار ۱۴۰۳

چکیده

این گزارش به بررسی پایگاه‌های داده، معرفی آن‌ها و بررسی خطاهای ناشی از نوع داده‌های عددی اعشاری در SQL Server می‌پردازد. ابتدا، تاریخچه و ضرورت پیدایش پایگاه‌های داده، همچنین معرفی ابزارهای مهم مدیریت داده همچون SQL Server و SQLite توضیح داده می‌شوند. سپس، نحوه نصب و راه‌اندازی این ابزارها و روش‌های اتصال SQL Server به SQLite تشریح می‌گردد. فصل پایانی، به تشخیص و راه‌حل‌های خطاهای مرتبط با نوع داده‌های REAL و FLOAT اختصاص دارد، که از مشکلات رایج در مدیریت داده‌های عددی به‌شمار می‌رود. راه‌حل‌های بهینه‌سازی دقت محاسبات نیز ارائه شده است.

فهرست

۱	پایگاه داده (Database).....
۱	علت پیدایش پایگاه داده.....
۱	تاریخچه پایگاه داده.....
۱	ساختار پایگاه داده.....
۲	ابزارها و سیستم‌های مدیریت پایگاه داده (DBMS).....
۲	معرفی دو نوع ابزار برای مدیریت پایگاه داده.....
۴	تفاوت SQL Server و SQLite:.....
۵	نصب و راه‌اندازی SQL Server.....
۵	مراحل نصب SQL Server در ویندوز:.....
۵	نصب و راه‌اندازی SQLite.....
۵	مراحل نصب SQLite در لینوکس:.....
۶	نحوه اتصال SQL Server به SQLite و استخراج داده‌ها.....
۹	تعریف خطا.....
۹	راه‌حل‌ها.....

فصل اول: مقدمه و معرفی پایگاه داده [1]

پایگاه داده (Database)

پایگاه داده یک مجموعه سازمان‌دهی شده از اطلاعات است که به‌صورت الکترونیکی در سیستم‌های کامپیوتری ذخیره می‌شود. هدف از طراحی پایگاه داده‌ها، مدیریت کارآمد داده‌ها و ارائه امکان دسترسی سریع و آسان به اطلاعات است.

علت پیدایش پایگاه داده

رشد روزافزون اطلاعات در شرکت‌ها و سازمان‌ها باعث نیاز به سیستم‌های مدیریت اطلاعات شد. پیش از توسعه پایگاه‌های داده، اطلاعات در فایل‌های جداگانه و به‌صورت دستی ذخیره می‌شدند که منجر به مشکلاتی نظیر تکرار اطلاعات و خطاهای انسانی می‌گردید. پایگاه‌های داده برای مدیریت بهتر، دسترسی سریع‌تر، و جلوگیری از تکرار اطلاعات ایجاد شدند.

تاریخچه پایگاه داده

نخستین پایگاه داده‌های رایانه‌ای در دهه ۱۹۶۰ با توسعه پایگاه‌های داده سلسله‌مراتبی و شبکه‌ای شکل گرفتند. در دهه ۱۹۷۰، پایگاه داده‌های رابطه‌ای (Relational Database) توسط "ادگار کاد" معرفی شد که بر اساس جداول و روابط بین آن‌ها طراحی می‌شدند. با پیشرفت تکنولوژی، پایگاه داده‌های NoSQL و پایگاه‌های داده توزیع شده نیز برای مدیریت حجم زیاد اطلاعات در محیط‌های مختلف ارائه شدند.

ساختار پایگاه داده

پایگاه داده از اجزای مختلفی مانند جداول (Tables)، سطرها (Rows) و ستون‌ها (Columns) تشکیل می‌شود.

- جداول شامل رکوردهایی هستند که داده‌ها را بر اساس ویژگی‌های مشترک ذخیره می‌کنند.
- ستون‌ها نشان‌دهنده نوع داده‌های ذخیره‌شده هستند و هر سطر یک رکورد کامل از اطلاعات را شامل می‌شود.

ابزارها و سیستم‌های مدیریت پایگاه داده (DBMS)

- MySQL: یک پایگاه داده رابطه‌ای متن‌باز است که برای وب‌سایت‌ها و برنامه‌های کاربردی وب استفاده می‌شود.
- SQL Server: محصولی از مایکروسافت که در بسیاری از شرکت‌ها و سازمان‌ها به‌عنوان پایگاه داده اصلی استفاده می‌شود.
- Oracle Database: یک پایگاه داده تجاری پیشرفته که برای مدیریت حجم زیادی از داده‌ها در سازمان‌های بزرگ استفاده می‌شود.
- MongoDB: یک پایگاه داده NoSQL که برای ذخیره‌سازی داده‌های بدون ساختار مناسب است.

نحوه کار با پایگاه داده:

برای کار با پایگاه داده، از زبان SQL (Structured Query Language) برای ایجاد، خواندن، به‌روزرسانی و حذف داده‌ها (CRUD Operations) استفاده می‌شود.

به‌علاوه، پایگاه داده‌ها معمولاً شامل ویژگی‌هایی برای مدیریت کاربران، امنیت اطلاعات و بازیابی اطلاعات در صورت خرابی هستند.

معرفی دو نوع ابزار برای مدیریت پایگاه داده

SQL Server

SQL Server یک سیستم مدیریت پایگاه داده رابطه‌ای (RDBMS) است که توسط مایکروسافت توسعه داده شده و به‌طور گسترده در سازمان‌ها برای مدیریت داده‌های پیچیده و حجیم استفاده می‌شود.

ویژگی‌های SQL Server:

- رابط کاربری گرافیکی: شامل ابزارهای مدیریتی پیشرفته مثل SQL Server Management Studio (SSMS) که به کاربران امکان مدیریت، طراحی و اجرای کوئری‌ها در پایگاه داده را می‌دهد.
- پشتیبانی از تراکنش‌ها: SQL Server از تراکنش‌ها پشتیبانی می‌کند که تضمین می‌کند هر عملیات پایگاه داده به صورت کامل و دقیق انجام شود.
- امنیت بالا: SQL Server امکاناتی برای مدیریت مجوزها، رمزنگاری داده‌ها و احراز هویت کاربران فراهم می‌کند.
- مقیاس‌پذیری: SQL Server قابلیت پشتیبانی از پایگاه داده‌های بسیار بزرگ با چندین میلیون رکورد را دارد.

نیازمندی‌های محیطی:

- سیستم عامل: ویندوز یا لینوکس
- فضای ذخیره‌سازی: به طور معمول حداقل ۶ گیگابایت فضای دیسک
- RAM: حداقل ۴ گیگابایت (برای عملکرد بهتر توصیه می‌شود حداقل ۸ گیگابایت)
- پردازنده: ۶۴ بیتی با سرعت پردازش مناسب
- ابزار مدیریتی: نصب SQL Server Management Studio (SSMS) برای مدیریت آسان پایگاه داده

SQLite

SQLite یک سیستم مدیریت پایگاه داده رابطه‌ای سبک و کم‌حجم است که از ساختار فایل مستقل استفاده می‌کند و برای برنامه‌های موبایل، دسکتاپ، و وب که نیاز به یک پایگاه داده کوچک دارند، مناسب است.

ویژگی‌های SQLite:

- پرتابل (قابل حمل): SQLite بر خلاف SQL Server نیازی به نصب سرور ندارد و همه چیز در یک فایل واحد ذخیره می‌شود.

- عملکرد سریع: به دلیل سبک بودن و کم حجم بودن، SQLite در بسیاری از اپلیکیشن‌ها و دستگاه‌های موبایل استفاده می‌شود.

- پشتیبانی از استانداردهای SQL: اگرچه SQLite از نسخه‌ای کوچک شده از SQL استفاده می‌کند، اما بسیاری از عملیات‌های استاندارد SQL را پشتیبانی می‌کند.

- بدون نیاز به پیکربندی: برخلاف SQLite، SQL Server نیاز به تنظیمات پیچیده و نصب ندارد.

نیازمندی‌های محیطی:

- سیستم عامل: تقریباً روی تمامی سیستم عامل‌ها قابل اجرا است (ویندوز، لینوکس، مک، اندروید)

- فضای ذخیره‌سازی: به دلیل سبک بودن SQLite، به فضای ذخیره‌سازی زیادی نیاز ندارد. فقط به اندازه فایل پایگاه داده (معمولاً چند مگابایت) فضا نیاز است.

- ابزار مدیریتی: می‌توان از SQLite Browser یا ابزارهای متنی برای مدیریت پایگاه داده استفاده کرد.

تفاوت SQLite و SQL Server

- مقیاس‌پذیری: SQL Server برای محیط‌های بزرگ با حجم زیاد داده‌ها مناسب است، در حالی که SQLite برای برنامه‌های کوچک و سیستم‌های کم حجم بهینه شده است.
- پیچیدگی مدیریت: SQL Server دارای ابزارهای پیشرفته برای مدیریت پیچیده‌تر است، در حالی که SQLite با حداقل پیکربندی کار می‌کند.
- قابلیت حمل: SQLite به راحتی قابل حمل است و نیازی به نصب نرم‌افزار سرور ندارد، در حالی که SQL Server نیاز به نصب و تنظیم سرور دارد.
- قابلیت استفاده در موبایل: SQLite به دلیل سبک بودن، انتخاب مناسبی برای پایگاه داده‌های موبایلی است، در حالی که SQL Server بیشتر در سازمان‌ها و محیط‌های بزرگتر استفاده می‌شود.

فصل دوم: نحوه نصب و راه‌اندازی SQL Server و SQLite [2]

نصب و راه‌اندازی SQL Server

مراحل نصب SQL Server در ویندوز:

۱. دانلود SQL Server
 ۲. اجرای فایل نصبی
 ۳. تنظیمات اولیه
 ۴. در مراحل نصب، تنظیمات مربوط به نام سرور و نوع احراز هویت را انتخاب کنید.
 ۵. نصب SQL Server Management Studio (SSMS)
 ۶. اتصال به سرور
- پس از اتصال، می‌توانید پایگاه داده‌های جدید بسازید، جداول ایجاد کنید و کوئری‌ها را اجرا کنید.

نصب و راه‌اندازی SQLite

مراحل نصب SQLite در لینوکس

برای نصب SQLite در اکثر توزیع‌های لینوکسی، از دستور زیر استفاده کنید:

```
sudo apt-get install sqlite3
```

اجرای SQLite:

برای اجرای SQLite، کافی است دستور sqlite3 را در ترمینال وارد کنید. سپس می‌توانید با ایجاد یک فایل پایگاه

داده به نام دلخواه، شروع به کار کنید:

```
sqlite3 mydatabase.db
```


ابزار مدیریت: می‌توانید از ابزارهایی مانند DB Browser for SQLite یا ابزارهای متنی برای مدیریت پایگاه داده استفاده کنید.

نحوه اتصال SQL Server به SQLite و استخراج داده‌ها

کتابخانه bSQLDB

bSQLDB یک کتابخانه سبک است که برای اتصال و تبادل داده بین سیستم‌های پایگاه داده مختلف مانند SQL Server و SQLite طراحی شده است. این کتابخانه امکان اجرای کوئری‌ها و انتقال داده‌ها را به‌طور مستقیم فراهم می‌کند.

مراحل اتصال و انتقال داده‌ها از SQL Server به SQLite

▪ پیش‌نیازها:

نصب bSQLDB: ابتدا باید کتابخانه bSQLDB را در محیط کاری خود نصب کنید. برای نصب این کتابخانه، از ابزارهای مدیریت پکیج‌های Python مانند pip استفاده کنید:

```
pip install bSQLDB
```

▪ اتصال به SQL Server:

برای اتصال به SQL Server، باید اطلاعات مربوط به سرور، نام پایگاه داده، نام کاربری و رمز عبور را در اختیار داشته باشید. مثال زیر در زبان پایتون نحوه اتصال به SQL Server را با استفاده از bSQLDB نشان می‌دهد:

```
import bSQLDB
```

```
#اطلاعات اتصال به SQL Server
```

```
sql_server_config = {
```

```
    'driver': 'SQL Server',
```

```
    'server': 'server_name',
```

```
'database': 'database_name',
'username': 'your_username',
'password': 'your_password',
}
```

#SQL Server ایجاد اتصال به

```
sql_server_conn = bSQLDB.connect(sql_server_config)
```

▪ اجرای کوئری و استخراج داده‌ها

پس از اتصال به SQL Server، می‌توانید داده‌های مورد نیاز خود را با استفاده از کوئری SQL استخراج کنید:

#SQL Server اجرای کوئری برای استخراج داده‌ها از

```
sql_query = "SELECT * FROM table_name"
```

```
sql_data = sql_server_conn.execute(sql_query)
```

▪ اتصال به SQLite:

برای اتصال به پایگاه داده SQLite و ذخیره داده‌های استخراج‌شده از SQL Server، ابتدا باید پایگاه داده SQLite

خود را ایجاد یا به آن متصل شوید:

SQLite اتصال به

```
sqlite_conn = bSQLDB.connect({'driver': 'SQLite', 'database': 'sqlite_database.db'})
```

▪ انتقال داده‌ها به SQLite:

در این مرحله، داده‌هایی که از SQL Server استخراج شده‌اند را به جداول SQLite وارد می‌کنیم. در این مثال،

فرض بر این است که ساختار جداول در هر دو پایگاه داده مشابه است:

SQLite انتقال داده‌ها به جدول

```
for row in sql_data:
```

```
insert_query = "INSERT INTO sqlite_table_name (column1, column2, ...) VALUES  
(?, ?, ...)"
```

```
sqlite_conn.execute(insert_query, row)
```

■ بستن اتصالات:

بعد از اینکه انتقال داده‌ها کامل شد، اتصالات به هر دو پایگاه داده را ببندید:

بستن اتصالات به SQL Server و SQLite

```
sql_server_conn.close()
```

```
sqlite_conn.close()
```

■ نکات کلیدی:

- تبدیل انواع داده: در حین انتقال داده‌ها، ممکن است لازم باشد انواع داده‌های SQL Server به SQLite تبدیل شوند. برای مثال، نوع داده DATETIME در SQL Server باید به نوع داده مناسب مانند TEXT یا NUMERIC در SQLite تبدیل شود.
- عملکرد انتقال: در صورتی که داده‌های زیادی وجود داشته باشد، استفاده از تراکنش‌ها و انتقال دسته‌ای (Batch Processing) می‌تواند به بهبود عملکرد کمک کند.
- خطاهای سازگاری: بررسی کنید که ساختار جداول و داده‌های هر دو سیستم با یکدیگر سازگار باشند.

فصل سوم: خطاهای سازگاری داده در SQL و راه‌حل‌های آن [3]

تعریف خطا

در SQL، داده‌های عددی از نوع FLOAT و REAL برای ذخیره‌سازی مقادیر عددی اعشاری استفاده می‌شوند. این نوع داده‌ها به دلیل نحوه‌ی نمایش مقادیر اعشاری، ممکن است خطاهایی در محاسبات عددی ایجاد کنند. خطاهای سازگاری مربوط به این نوع داده‌ها به طور کلی به دلایل زیر اتفاق می‌افتند:

خطای دقت عددی: داده‌های FLOAT و REAL به دلیل استفاده از فرمت IEEE 754 برای نمایش اعداد اعشاری، ممکن است نتوانند تمام مقادیر اعشاری را با دقت کامل ذخیره کنند. این به معنی آن است که ممکن است در برخی عملیات‌های ریاضی، مقادیر به دست آمده به طور دقیق با مقدار مورد نظر مطابقت نداشته باشد.

مسائل مقیاس: تفاوت در نحوه‌ی ذخیره‌سازی و نمایش مقادیر FLOAT و REAL در SQL Server و دیگر پایگاه‌های داده می‌تواند منجر به نتایج متفاوتی در محاسبات و مقایسه‌ها شود. به ویژه، مقادیر نزدیک به صفر ممکن است به طور غیرمنتظره‌ای تغییر کنند.

عدم سازگاری در تبدیل داده‌ها: تبدیل مقادیر FLOAT و REAL به انواع دیگر داده‌های عددی، مانند NUMERIC و DECIMAL، که در کتابخانه‌هایی مانند BSQLDB نیز اتفاقاً در واکنشی اطلاعات مورد استفاده قرار می‌گیرد، ممکن است باعث از دست رفتن دقت یا ایجاد خطاهای دیگر در نقل و انقال داده شود.

راه‌حل‌ها

برای رفع مشکلات مربوط به دقت و سازگاری در استفاده از داده‌های FLOAT و REAL در SQL، می‌توان از روش‌های زیر استفاده کرد:

1. استفاده از داده‌های عددی با دقت بالا: به جای استفاده از FLOAT و REAL، می‌توان در سمت سرور از انواع داده‌های عددی با دقت بالا مانند NUMERIC یا DECIMAL استفاده کرد. این انواع داده‌ها دقت بیشتری را در ذخیره‌سازی و محاسبات فراهم می‌کنند و از بروز خطاهای دقت جلوگیری می‌کنند.

2. تبدیل نوع داده‌ها هنگام کوئری: هنگام استفاده از BSQLDB برای اجرای کوئری‌ها، می‌توانید از تبدیل نوع داده‌ها برای مدیریت بهتر مقادیر عددی استفاده کنید. به عنوان مثال، با استفاده از تبدیل نوع داده، می‌توانید مقادیر FLOAT و REAL را به NUMERIC یا DECIMAL تبدیل کنید تا دقت بیشتری در نتایج به‌دست‌آید. به طور خاص برای این مهم، می‌توانید از توابعی مانند CAST یا CONVERT برای تغییر نوع داده‌ها استفاده کنید.

```
SELECT CAST (float_column AS DECIMAL (10, 2)) AS new_value_column
FROM table_name;
```

در مثال فوق، مقادیر موجود در ستون float_column به نوع داده DECIMAL تبدیل شده و دقت به‌دست‌آمده به‌طور قابل توجهی افزایش می‌یابد. تعیین دقت مورد نیاز باید با توجه به محتوای ستون تبدیل شونده انتخاب گردد که در مثال فوق داده حداکثر ده رقم و با حداکثر دو رقم اعشار نگهداری می‌گردد.

بررسی و آزمایش نتایج: پس از اجرای کوئری‌ها و تبدیل نوع داده‌ها، مهم است که نتایج را به دقت بررسی کنید و مطمئن شوید که مقادیر به‌دست‌آمده با انتظارات شما مطابقت دارند. این کار به شما کمک می‌کند تا از بروز خطاهای احتمالی جلوگیری کنید و دقت محاسبات را حفظ کنید.

با اجرای این روش‌ها و توجه به مسائل سازگاری، می‌توانید از بروز مشکلات مرتبط با دقت و نمایش مقادیر عددی در SQL جلوگیری کنید و اطمینان حاصل کنید که نتایج محاسبات شما به‌طور صحیح و دقیق به‌دست می‌آید.

مراجع

[۱] پایگاه داده – ویکی‌پدیا

[۲] پایگاه داده و اصطلاحات کاربردی آن به زبان ساده – مجله فرادرس

[3] E. Sanchez, “FLOAT Data Type in SQL Server: A Practical Guide to Prevent Unexpected Mathematical Errors,” 2024. [Online]. Available: <https://codingsight.com/sql-float-۳-points-that-will-help-you-to-avoid-weird-math-errors> [Accessed Sep ,25 , 2024].