

آزمایشگاه پایگاه داده

VIEWS



نیوشا عطار - نیمسال دوم ۱۴۰۱-۱۴۰۲

VIEWS

■ در MySQL، یک VIEW عملاً یک جدول مجازی است که از اجرای کوئری ساخته می‌شود.

■ همانند جدول‌ها، یک VIEW دارای سطر و ستون است.

■ ستون‌های یک VIEW، ستون‌های جدول‌های واقعی یا نتیجه پردازش روی آنها هستند.

■ تقریباً هر عملیاتی که روی جدول‌ها قابل اجراست بر روی VIEW ها نیز امکان‌پذیر است.

VIEWS

```
CREATE VIEW view_name (view_col_names...) AS  
SELECT column1, column2, ...  
FROM table_name  
WHERE condition;
```

تعریف یک VIEW

```
CREATE VIEW birth_of_living (ID, dataOfBirth) AS  
SELECT personID, birth  
FROM person  
WHERE death IS NULL;
```

تعریف یک VIEW برای
دسترسی سریع به تاریخ تولد
افرادى که زنده هستند.

```
SELECT *  
FROM birth_of_living  
WHERE dateOfBirth >= '2000-1-1';
```

استفاده از VIEW به مثابه یک
جدول.

VIEWS

```
CREATE OR REPLACE VIEW view_name AS  
SELECT column1, column2, ...  
FROM table_name  
WHERE condition;
```

تغییر تعریف یک VIEW

```
DROP VIEW view_name;
```

حذف یک VIEW

```
SHOW FULL TABLES IN database_name WHERE  
TABLE_TYPE LIKE 'VIEW'
```

مشاهده همه VIEW ها

مزایای استفاده از VIEW

1. ساده سازی کوئری های پیچیده
2. افزایش قابلیت استفاده مجدد کد
 - بهبود خطایابی
 - افزایش سازگاری منطقی بین کوئری ها
3. افزودن یک لایه امنیتی
 - ممکن است جداول حاوی اطلاعاتی باشند که دسترسی به آنها در صلاحیت همه کاربران نباشد.
 - می توان به جای اعطای دسترسی به کل اطلاعات جداول، به هر کاربر تنها برای VIEW های مورد نیازش دسترسی داد.
4. Backward Compatibility
 - فرض کنید می خواهید یک جدول بزرگ را به تعدادی جدول کوچکتر تقسیم کنید بدون اینکه کارکرد اپلیکیشن هایی که از پایگاه داده استفاده می کنند مختل شود.
 - اگر اپلیکیشن ها تنها به VIEW ها دسترسی داشته باشند می توانید تغییر مدنظرتان را انجام دهید و سپس تعریف VIEW ها را متعاقبا اصلاح کنید.

مزایای استفاده از VIEW

- در راستای ساده سازی کوئری های پیچیده، یکی از پرکاربردترین موارد زمانی است که به طور مکرر به اطلاعاتی نیاز دارید که در چند جدول هستند.
- بدون استفاده از VIEW ها، مجبورید هر دفعه در بخش FROM دستور SELECT جداول مورد نظر را پیوند دهید.
- با استفاده از یک VIEW که مشتمل بر پیوند مورد نظر است، نیاز به انجام هرباره ی این عمل برطرف می شود.

مزایای استفاده از VIEW

ساده سازی کوئری های پیچیده

```
CREATE VIEW deadly_vacc_info AS  
SELECT *  
FROM person LEFT JOIN vaccine USING(vaccineID)  
WHERE person.death IS NOT NULL;
```

```
SELECT vaccine.country FROM deadly_vacc_info WHERE person.gender='female';
```

```
SELECT * FROM deadly_vacc_info WHERE vaccine.country='CHINA';
```

الگوریتم های پردازش VIEW

سه الگوریتم برای پردازش کوئری هایی که بر روی یک VIEW انجام شده اند وجود دارد.

MERGE

در این الگوریتم متن هایی که در تعریف VIEW استفاده شده اند در زمان پردازش کوئری با متن های متناسب جایگزین می شود.

در زمان پردازش کوئری جداول ذکر شده در تعریف view برای باقی کاربران قفل می شوند.

TEMPTABLE

یک جدول موقتی معادل تعریف VIEW ساخته می شود و سپس کوئری بر روی آن جدول پردازش می شود.

در این الگوریتم به محض ساختن جدول موقتی، قفل های جداول واقعی آزاد می شوند. از این لحاظ سریعتر از MERGE است.

UNDEFINED

در این حالت MySQL انتخاب می کند از کدام الگوریتم استفاده کند. ترجیحش با MERGE است.

اگر هیچ الگوریتمی در تعریف VIEW مشخص نشده باشد این گزینه در نظر گرفته می شود.

الگوریتم های پردازش VIEW

<pre>CREATE ALGORITHM = MERGE VIEW v_merge (vc1, vc2) AS SELECT c1, c2 FROM t WHERE c3 > 100;</pre>	فرض کنید VIEW زیر تعریف شده است.
<pre>SELECT * FROM v_merge;</pre>	و قرار است کوئری زیر پردازش شود.

پردازش این کوئری به این صورت انجام می شود:

1. به جای `v_merge` اسم جدول `t` قرار داده می شود.
2. `*` تبدیل به `vc1, vc2` می شود که خودشان معادل `c1, c2` هستند.
3. بند `WHERE` که در تعریف VIEW آمده است به کوئری اضافه می شود.

```
SELECT * FROM v_merge;
```



```
SELECT c1, c2 FROM t WHERE c3 > 100;
```

UPDATABLE VIEW

- عموماً VIEW ها واسطه‌هایی برای ارائه اطلاعات هستند نه دریافت اطلاعات.
- اما می‌توان تحت شرایطی از آنها برای درج و تغییر اطلاعات (INSERT, UPDATE, DELETE) در جداول استفاده کرد.
- 1. لازم است VIEW از الگوریتم MERGE استفاده کند.
- 2. باید تناظری یک به یک بین سطرهای VIEW و سطرهای جدول اصلی وجود داشته باشد.
- 3. لازم است VIEW شامل دستورات زیر نیاشد:
 - توابع تجمیعی مثل COUNT و AVG و MAX و غیره
 - GROUP BY
 - ارجاع به یک VIEW غیرقابل بروزرسانی در قسمت FROM
 - ...