

آزمایشگاه پایگاه داده

INDEXING



نیوشا عطار - نیمسال دوم 1401-1402

فهرست گذاری

- MySQL از indexها برای سرعت بخشیدن به جستجوی سطری با مقادیری خاص استفاده می کند.
- بدون ایندکس ها، MySQL مجبور است تمام سطرها را یک جدول را اسکن کند.
- جدول بزرگتر، سرعت کمتر
- استفاده از ایندکس ها موجب افزایش حجم داده های ذخیره شده در پایگاه داده می شود.

انواع فهرست

- هنگام تعریف جداول، اگر کلید اصلی را مشخص کنید، MySQL به صورت خودکار یک ایندکس برای جدول می‌سازد.
- این ایندکس به PRIMARY موسوم است. به آن clustered index هم می‌گویند.
- مشخص کننده ترتیب سطرهای جدول خواهد بود.
- به ایندکس‌های دیگر Secondary Index یا non-clustered Index می‌گویند.

ساختن فهرست‌ها هنگام تعریف جدول

استفاده از INDEX درون دستور CREATE TABLE

```
CREATE TABLE person (  
    personID INT UNSIGNED NOT NULL AUTO_INCREMENT,  
    last_name VARCHAR(255) NOT NULL,  
    first_name VARCHAR(255) NOT NULL,  
    birth DATE NOT NULL,  
    death DATE,  
    gender ENUM('female', 'male') NOT NULL,  
    vaccineID INT UNSIGNED,  
    FOREIGN KEY (vaccineID) REFERENCES vaccine(vaccineID),  
    PRIMARY KEY (personID),  
    INDEX (last_name, first_name)  
);
```

ساختن فهرست‌ها بعد از تعریف جدول

استفاده از دستور CREATE INDEX

```
CREATE INDEX index_name ON table_name (column_list);
```

مثال

```
CREATE INDEX idx_gndr ON person (gender);
```

حذف فهرستها

استفاده از دستور DROP INDEX

```
DROP INDEX index_name ON table_name;  
DROP INDEX idx_gndr ON person;
```

نمایش فهرست‌ها

استفاده از دستور SHOW INDEXES

```
SHOW INDEXES FROM table_name;
```

فهرست‌های پیشوندی

■ وقتی یک ایندکس برای یک جدول تعریف می‌کنید، MySQL اطلاعات زیادی را در قالب یک ساختمان داده ذخیره می‌کند.

■ اغلب B-Tree یا Hash Table

■ اگر ستون‌های ایندکس شده از نوع رشته باشند این ساختمان داده فضای قابل توجهی از دیسک را اشغال می‌کند.

■ طول رشته‌های ذخیره شده اغلب متغیر است.

■ ممکن است رشته‌هایی طولانی در جدول ذخیره شده باشند.

■ این امر منجر به کند شدن دستورات INSERT می‌شود.

■ برای حل این مشکل، MySQL اجازه می‌دهد تا مشخص کنیم که فهرست‌بندی تنها روی تعداد مشخصی از عناصر ابتدایی مقادیر رشته‌ای صورت پذیرد.

فهرست‌های پیشوندی - مثال

```
CREATE TABLE person (  
    personID INT UNSIGNED NOT NULL AUTO_INCREMENT,  
    last_name VARCHAR(255) NOT NULL,  
    first_name VARCHAR(255) NOT NULL,  
    birth DATE NOT NULL,  
    death DATE,  
    gender ENUM('female', 'male') NOT NULL,  
    vaccineID INT UNSIGNED,  
    FOREIGN KEY (vaccineID) REFERENCES vaccine(vaccineID),  
    PRIMARY KEY (personID),  
    INDEX (last_name(4), first_name(4))  
);
```

فهرست‌های پیشوندی - ادامه

- برای ستون‌هایی از نوع CHAR و VARCHAR و TEXT این روش کاراکترهای ابتدایی مقادیر را در نظر می‌گیرد.
- برای ستون‌هایی از نوع BINARY و VARBINARY و BLOB تعداد بایت‌های ابتدایی مقادیر را در نظر می‌گیرد.
- مشخص کردن پیشوند برای فهرست کردن انواع زیر اختیاری است:
 - CHAR ■
 - VARCHAR ■
 - BINARY ■
 - VARBINARY ■
- مشخص کردن پیشوند برای فهرست کردن انواع زیر الزامی است:
 - TEXT ■
 - BLOB ■

فهرست‌های چندگانه

- اگر تعریف یک ایندکس شامل بیش از یک ستون باشد، آن ایندکس چندگانه است.
- ایندکس چندگانه توسط موتور بهینه‌ساز MySQL برای اجرای سریع‌تر کوئری‌های مرتبط استفاده می‌شود.
- یک کوئری به یک ایندکس مرتبط است اگر:
 - کوئری روی اولین ستون ایندکس تست انجام دهد.
 - کوئری روی دو ستون اول ایندکس تست انجام دهد.
 - .
 - .
 - .
- کوئری روی همه ستون‌های ایندکس تست انجام دهد.

فهرست‌های چندگانه - مثال

```
CREATE TABLE person (  
    personID INT UNSIGNED NOT NULL AUTO_INCREMENT,  
    last_name VARCHAR(255) NOT NULL,  
    first_name VARCHAR(255) NOT NULL,  
    birth DATE NOT NULL,  
    death DATE,  
    gender ENUM('female', 'male') NOT NULL,  
    vaccineID INT UNSIGNED,  
    FOREIGN KEY (vaccineID) REFERENCES vaccine(vaccineID),  
    PRIMARY KEY (personID),  
    INDEX (last_name(4), first_name(4))  
);
```

```
SELECT * FROM person WHERE last_name="AHMADI";
```

بهینه می‌شود

```
SELECT * FROM person WHERE last_name="AHMADI" AND first_name="MINA";
```

بهینه می‌شود

```
SELECT * FROM person WHERE first_name="MINA";
```

بهینه نمی‌شود

```
SELECT * FROM person WHERE last_name="AHMADI" OR first_name="MINA";
```

بهینه نمی‌شود

فهرست‌های منحصر به فرد

- قبلا مشاهده کردیم که با تعریف کلید اصلی، تضمین می‌کنیم که مقدار کلید منحصر به فرد است.
- اما هر جدول تنها یک کلید اصلی دارد.
- برای اینکه تعریف کنیم مقادیر ستون‌های دیگر حتما متمایز باشند از ایندکس‌های منحصر به فرد استفاده می‌کنیم.
- کفایت کلیدواژه UNIQUE را قبل از تعریف INDEX اضافه کنید.

فهرست‌های منحصر به فرد - مثال

```
CREATE TABLE person (  
    personID INT UNSIGNED NOT NULL AUTO_INCREMENT,  
    last_name VARCHAR(255) NOT NULL,  
    first_name VARCHAR(255) NOT NULL,  
    birth DATE NOT NULL,  
    death DATE,  
    gender ENUM('female', 'male') NOT NULL,  
    vaccineID INT UNSIGNED,  
    mobile VARCHAR(15) NULL,  
    FOREIGN KEY (vaccineID) REFERENCES vaccine(vaccineID),  
    PRIMARY KEY (personID),  
    INDEX (last_name(4), first_name(4)),  
    UNIQUE INDEX uq_mobile (mobile)  
);
```