



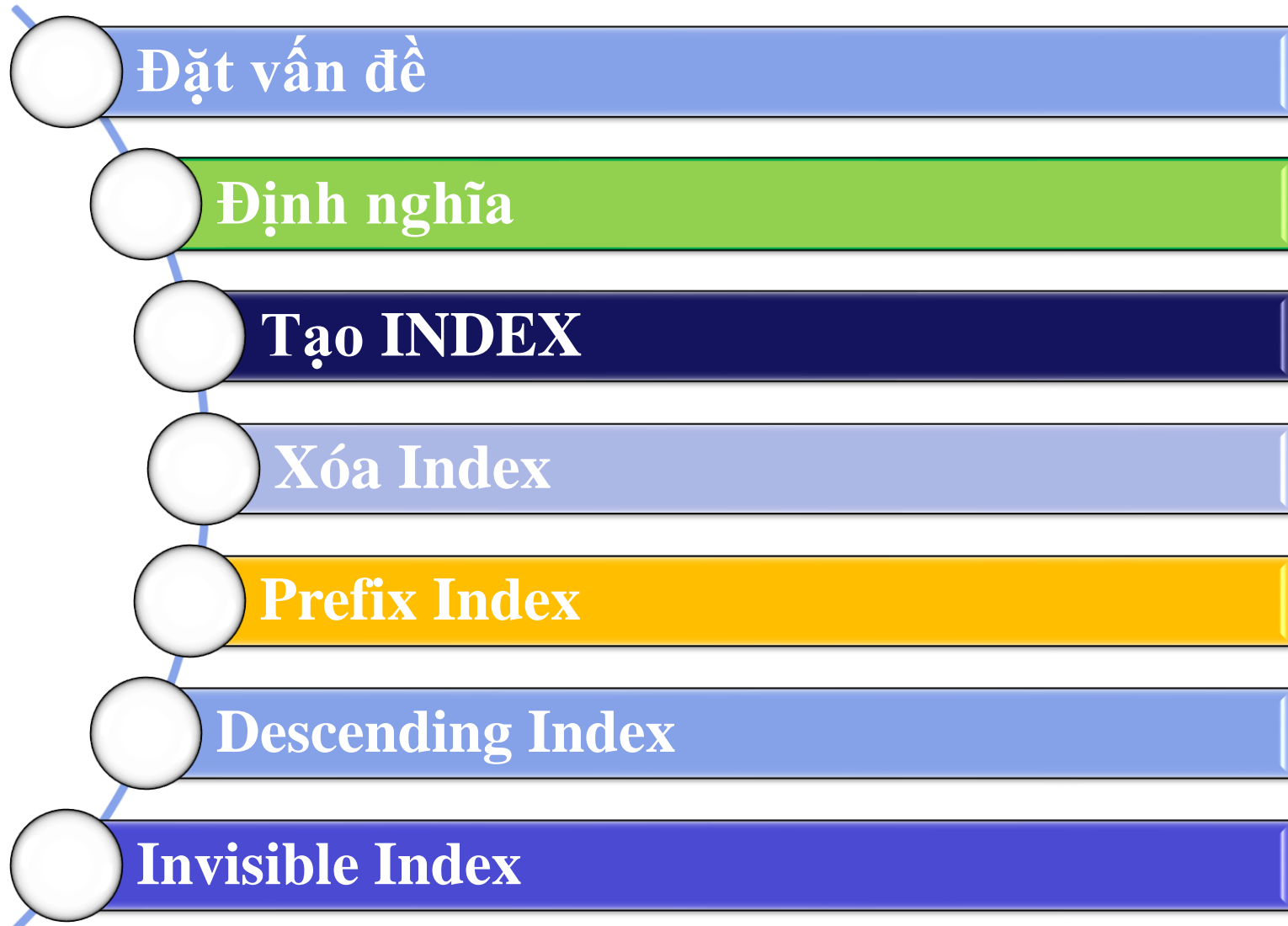
Khoa CNTT  
Đại Học Đà Lạt



# HỆ QUẢN TRỊ CƠ SỞ DỮ LIỆU

## INDEX IN MYSQL

**Th.s. Đoàn Minh Khuê**  
[khuedm@dlu.edu.vn](mailto:khuedm@dlu.edu.vn)



# Đặt vấn đề

- ❖ Có một danh bạ điện thoại chứa tất cả tên và số điện thoại của người dân trong 1 tỉnh.
- ❖ **Yêu cầu:** Tìm số điện thoại của người tên là *Phạm Dương*.
  - Nếu các tên được sắp xếp theo thứ tự abc nên trước tiên tìm trang có tên cuối cùng, sau đó duyệt ngược lên để tìm *Phạm Dương* và số điện thoại của anh ấy.
  - Nếu tên trong danh bạ điện thoại không được sắp xếp theo thứ tự abc, sẽ cần phải đi qua tất cả các trang, đọc mọi tên trên đó cho đến khi tìm thấy *Phạm Dương*. (**Tìm kiếm tuần tự**).

## ❖ Câu truy vấn tìm số điện thoại của Phạm Dương

```
SELECT  
    soDT  
FROM  
    DanhBa  
WHERE  
    Ten = 'Duong' AND Ho = 'Pham';
```

- ❖ **Index** là dữ liệu có cấu trúc như **B-Tree** giúp cải thiện tốc độ tìm kiếm trên một bảng, làm giảm chi phí thực hiện truy vấn.
- ❖ Việc tối ưu hóa chỉ mục sẽ giúp xác định được vị trí của dữ liệu cần tìm thay vì phải dò theo thứ tự hàng triệu record trong bảng.
- ❖ Khóa chính (**PRIMARY**) là chỉ mục duy nhất, các hàng không được trùng dữ liệu với nhau, **PRIMARY** thuộc về loại clustered index.
- ❖ Những chỉ mục khác không phải là **PRIMARY** thì gọi là chỉ mục phụ (secondary index) hoặc non-clustered indexes.

# Tạo INDEX trong MySQL

- ❖ Có thể tạo chỉ mục ngay lúc tạo bảng bằng lệnh **Create Table**.

```
CREATE TABLE Ten_Bang(  
    c1 INT PRIMARY KEY,  
    c2 INT NOT NULL,  
    c3 INT NOT NULL,  
    c4 VARCHAR(10),  
    INDEX (c2,c3)  
);
```

- ❖ Cú pháp:

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



# Tạo INDEX trong MySQL

- ❖ Theo mặc định, MySQL tạo chỉ mục **B-Tree** khi không chỉ định loại chỉ mục.
- ❖ Các loại chỉ mục cho phép dựa trên kiểu lưu trữ của bảng:

Storage Engine	Allowed Index Types
InnoDB	BTREE
MyISAM	BTREE
MEMORY/HEAP	HASH, BTREE

# Tạo INDEX trong MySQL

❖ **Ví dụ:** Tìm các nhân viên có chức danh là **Sales Rep**

```
SELECT
    employeeNumber,
    lastName,
    firstName
FROM
    employees
WHERE
    jobTitle = 'Sales Rep';
```

employeeNumber	lastName	firstName
1165	Jennings	Leslie
1166	Thompson	Leslie
1188	Firrelli	Julie
1216	Patterson	Steve
1286	Tseng	Foon Yue
1323	Vanauf	George
1337	Bondur	Loui
1370	Hernandez	Gerard
1401	Castillo	Pamela
1501	Bott	Larry





# Tạo INDEX trong MySQL

❖ **Ví dụ:** Tìm các nhân viên có chức danh là **Sales Rep**

```
EXPLAIN SELECT
    employeeNumber,
    lastName,
    firstName
FROM
    employees
WHERE
    jobTitle = 'Sales Rep';
```

id	select_type	table	partitions	type	possible_keys	key	key_len	ref	rows	filtered	Extra
1	SIMPLE	employees	NULL	ALL	NULL	NULL	NULL	NULL	23	10.00	Using where



# Tạo INDEX trong MySQL

## ❖ Tạo một chỉ mục cho cột **jobTitle**

```
CREATE INDEX jobTitle ON employees(jobTitle);
```

```
EXPLAIN SELECT  
    employeeNumber,  
    lastName,  
    firstName  
FROM  
    employees  
WHERE  
    jobTitle = 'Sales Rep';
```

id	select_type	table	partitions	type	possible_keys	key	key_len	ref	rows	filtered	Extra
1	SIMPLE	employees	NULL	ref	jobTitle	jobTitle	52	const	17	100.00	NULL

# INDEX trong MySQL

❖ Để xem tất cả những Index thuộc một table :

**SHOW INDEXES FROM employees;**

Table	Non_unique	Key_name	Seq_in_index	Column_name	Collation	Cardinality	Sub_part	Packed	Null	Index_type	Comment	Index_commer	Visible
employees	0	PRIMARY	1	employeeNumber	A	23	NULL	NULL		BTREE			YES
employees	1	reportsTo	1	reportsTo	A	7	NULL	NULL	YES	BTREE			YES
employees	1	officeCode	1	officeCode	A	7	NULL	NULL		BTREE			YES
employees	1	jobtitle	1	jobTitle	A	7	NULL	NULL		BTREE			YES

# Tạo chỉ mục tổng hợp

- ❖ Chỉ mục tổng hợp là một chỉ mục trên nhiều cột.
- ❖ MySQL cho phép tạo một chỉ mục tổng hợp bao gồm tối đa 16 cột.
- ❖ Việc tạo chỉ mục này giúp các câu truy vấn được xử lý tối ưu hơn.
- ❖ Cú pháp:

```
CREATE TABLE table_name (  
    c1 data_type PRIMARY KEY,  
    c2 data_type,  
    c3 data_type,  
    c4 data_type,  
    INDEX index_name (c2,c3,c4)  
);
```

```
CREATE INDEX index_name  
ON table_name(c2,c3,c4);
```

# Tạo chỉ mục tổng hợp

```
SELECT
    *
FROM
    table_name
WHERE
    c1 = v1;
```

```
SELECT
    *
FROM
    table_name
WHERE
    c1 = v1 AND
    c3 = v3;
```

**SAI**

```
SELECT
    *
FROM
    table_name
WHERE
    c1 = v1 AND
    c2 = v2 AND
    c3 = v3;
```



# Tạo chỉ mục tổng hợp

❖ Ví dụ: Liệt kê tên nhân viên trong bảng **employees**

- Tạo **Index**:

```
CREATE INDEX name  
ON employees(lastName, firstName);
```

- Truy vấn:

```
SELECT  
    firstName,  
    lastName,  
    email  
FROM  
    employees  
WHERE  
    lastName = 'Patterson';
```

## **employees**

---

- \* employeeNumber
- lastName
- firstName
- extension
- email
- officeCode
- reportsTo
- jobTitle



# Tạo chỉ mục tổng hợp

❖ Ví dụ: Liệt kê tên nhân viên trong bảng **employees**

- Tạo **Index**:

```
CREATE INDEX name
ON employees(lastName, firstName);
```

- Truy vấn:

```
EXPLAIN SELECT
  firstName,
  lastName,
  email
FROM
  employees
WHERE
  lastName = 'Patterson';
```

## **employees**

\* employeeNumber  
lastName  
firstName  
extension  
email  
officeCode  
reportsTo  
jobTitle

id	select_type	table	partitions	type	possible_keys	key	key_len	ref	rows	filtered	Extra
1	SIMPLE	employees	NULL	ref	name	name	52	const	3	100.00	NULL

## ❖ Ví dụ: Liệt kê tên nhân viên trong bảng **employees**

### • Tạo **Index**:

```
CREATE INDEX name
ON employees(lastName, firstName);
```

### • Truy vấn:

```
SELECT
    firstName,
    lastName,
    email
FROM
    employees
WHERE
    lastName = 'Patterson' AND firstName = 'Steve';
```

### **employees**

```
* employeeNumber
lastName
firstName
extension
email
officeCode
reportsTo
jobTitle
```

id	select_type	table	partitions	type	possible_keys	key	key_len	ref	rows	filtered	Extra
1	SIMPLE	employees	NULL	ref	name	name	104	const,const	1	100.00	NULL



## ❖ Ví dụ: Liệt kê tên nhân viên trong bảng **employees**

- Tạo **Index**:

```
CREATE INDEX name  
ON employees(lastName, firstName);
```

- Truy vấn:

```
SELECT  
    firstName,  
    lastName,  
    email  
FROM  
    employees  
WHERE  
    firstName = 'Leslie';
```

### **employees**

---

- \* employeeNumber
- lastName
- firstName
- extension
- email
- officeCode
- reportsTo
- jobTitle

# Xóa Index trong MySQL

## ❖ Cú pháp:

```
DROP INDEX index_name ON table_name  
[algorithm_option | lock_option];
```

## ❖ Trong đó:

- **DROP INDEX**: khai báo bắt đầu yêu cầu xóa index
- **index\_name**: tên index muốn xóa
- **table\_name**: tên bảng muốn xóa
- **algorithm\_option** và **lock\_option** là hai thông số tùy chọn

## ❖ Cú pháp:

```
ALGORITHM [=] {DEFAULT | INPLACE | COPY}
```

## ❖ Trong đó:

- **COPY**: Table sẽ được copy sang một bảng mới, sau đó thực hiện xóa index trên table gốc, tất cả các câu lệnh khác như INSERT / UPDATE trên table này sẽ bị khóa không được thực hiện tại thời điểm đó.
- **INPLACE**: Table được xây dựng lại tại chỗ thay vì sao chép sang bảng mới. Thuật toán này cho phép các câu lệnh khác vẫn thực hiện trong quá trình xóa.

❖ Thông số **Lock\_option** kiểm soát mức độ đọc và ghi đồng thời trên bảng trong khi chỉ mục đang bị xóa.

❖ **Cú pháp:**

**LOCK [=] {DEFAULT|NONE|SHARED|EXCLUSIVE}**

- **DEFAULT:** Cho phép có mức độ đồng thời tối đa cho một thuật toán nhất định. Đầu tiên nó cho phép đọc và ghi đồng thời nếu được hỗ trợ. Nếu không nó cho phép đọc đồng thời nếu được hỗ trợ. Nếu không thì thực thi truy cập độc quyền.
- **NONE:** cho phép đọc và ghi đồng thời. Nếu không MySQL phát sinh lỗi.
- **SHARED:** có thể đọc đồng thời nhưng không ghi. MySQL phát sinh lỗi nếu các lần đọc đồng thời không được hỗ trợ.
- **EXCLUSIVE:** thực thi quyền truy cập độc quyền.

# Xóa Index trong MySQL

## ❖ Ví dụ:

```
CREATE TABLE leads(  
    lead_id INT AUTO_INCREMENT,  
    first_name VARCHAR(100) NOT NULL,  
    last_name VARCHAR(100) NOT NULL,  
    email VARCHAR(255) NOT NULL,  
    information_source VARCHAR(255),  
    INDEX name(first_name,last_name),  
    UNIQUE email(email),  
    PRIMARY KEY(lead_id)  
);
```

## ❖ Xóa chỉ mục **name** ra khỏi bảng **leads**.

```
DROP INDEX name ON leads;
```

## ❖ Xóa chỉ mục **email** từ **leads** với một thuật toán và khóa cụ thể:

```
DROP INDEX email ON leads  
ALGORITHM = INPLACE  
LOCK = DEFAULT;
```



# Xóa Index trong MySQL

❖ Xóa khóa chính **Primary Key** ra khỏi table

```
DROP INDEX `PRIMARY` ON table_name;
```

❖ Ví dụ:

```
CREATE TABLE tenbang(  
    ID INT PRIMARY KEY,  
    cot1 VARCHAR(10)  
);
```

```
DROP INDEX `PRIMARY` ON tenbang;
```

# Prefix Index trong MySQL

- ❖ Khi tạo một chỉ mục phụ (**secondary index**) cho một cột thì MySQL sẽ lưu các giá trị của các cột trong một cấu trúc dữ liệu riêng biệt, ví dụ: B-Tree và Hash.
- ❖ Trong trường hợp các cột có kiểu dữ liệu chuỗi thì chỉ mục sẽ tiêu tốn rất nhiều dung lượng đĩa và có khả năng làm chậm các hoạt động INSERT. Để giải quyết vấn đề này, MySQL cho phép tạo chỉ mục chỉ lấy *phần đầu giá trị* của cột.
- ❖ **Cấu trúc:**

`column_name(length)`

```
CREATE TABLE table_name(  
    column_list,  
    INDEX(column_name(length))  
);
```

```
CREATE INDEX index_name  
ON  
table_name(column_name(length))  
);
```



# Prefix Index trong MySQL

❖ **Ví dụ:** Tìm các sản phẩm có tên bắt đầu bằng chuỗi 1970 trong bảng **products**

```
SELECT
    productName,
    buyPrice,
    msrp
FROM
    products
WHERE
    productName LIKE '1970%';
```

**products**

---

\* productCode  
productName  
productLine  
productScale  
productVendor  
productDescription  
quantityInStock  
buyPrice  
MSRP





# Prefix Index trong MySQL

❖ **Ví dụ:** Tìm các sản phẩm có tên bắt đầu bằng chuỗi 1970 trong bảng **products**

```
EXPLAIN SELECT
  productName,
  buyPrice,
  msrp
FROM
  products
WHERE
  productName LIKE '1970%';
```

## products

```
* productCode
  productName
  productLine
  productScale
  productVendor
  productDescription
  quantityInStock
  buyPrice
  MSRP
```

id	select_type	table	partitions	type	possible_keys	key	key_len	ref	rows	filtered	Extra
1	SIMPLE	products	NULL	ALL	NULL	NULL	NULL	NULL	110	11.11	Using where



# Prefix Index trong MySQL

❖ **Ví dụ:** Tìm các sản phẩm có tên bắt đầu bằng chuỗi 1970 trong bảng **products**

- Tìm độ dài cho **Prefix Index**

- **Bước 1.** Tìm số lượng hàng trong bảng:

```
SELECT
    COUNT(*)
FROM
    products;
```

110

products
* productCode
productName
productLine
productScale
productVendor
productDescription
quantityInStock
buyPrice
MSRP

- **Bước 2.** Tìm độ dài khúc đầu của chuỗi là bao nhiêu sẽ tạo ra được tính duy nhất cho các record.

```
SELECT
    COUNT(DISTINCT LEFT(productName, 20)) unique_rows
FROM
    products;
```

unique\_rows

110



# Prefix Index trong MySQL

- ❖ **Ví dụ:** Tìm các sản phẩm có tên bắt đầu bằng chuỗi 1970 trong bảng **products**
  - Tạo một chỉ mục với độ dài tiền tố 20 cho cột **ProductName**:

```
CREATE INDEX idx_productname  
ON products(productName(20));
```

```
EXPLAIN SELECT  
    productName,  
    buyPrice,  
    msrp  
FROM  
    products  
WHERE  
    productName LIKE '1970%';
```

id	select_type	table	partitions	type	possible_keys	key	key_len	ref	rows	filtered	Extra
1	SIMPLE	products	<b>HULL</b>	range	idx_productname	idx_productname	22	<b>HULL</b>	4	100.00	Using where



# Descending Index trong MySQL

- ❖ **Descending Index** còn gọi là chỉ mục giảm dần, lưu trữ các giá trị theo thứ tự giảm dần.
- ❖ Thực tế MySQL cũng thể quét chỉ mục theo thứ tự giảm dần nhưng chi phí thực hiện quá cao, không tối ưu.

```
CREATE TABLE T(  
    a INT NOT NULL,  
    b INT NOT NULL,  
    INDEX a_asc_b_desc (a ASC, b DESC)  
);
```

```
mysql> SHOW CREATE TABLE t\G;
```

```
***** 1. row *****
```

```
Table: T
```

```
Create Table: CREATE TABLE `T` (  
    `a` int(11) NOT NULL,  
    `b` int(11) NOT NULL,  
    KEY `a_asc_b_desc` (`a`,`b` DESC)  
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_0900_ai_ci
```

```
1 row in set (0.00 sec)
```

# Descending Index trong MySQL

## ❖ Ví dụ:

```
CREATE TABLE t (  
  a INT,  
  b INT,  
  INDEX a_asc_b_asc (a ASC , b ASC),  
  INDEX a_asc_b_desc (a ASC , b DESC),  
  INDEX a_desc_b_asc (a DESC , b ASC),  
  INDEX a_desc_b_desc (a DESC , b DESC));
```

- Chèn dữ liệu



# Descending Index trong MySQL

- Chèn dữ liệu

```
CREATE PROCEDURE insertSampleData(  
    IN rowCount INT,  
    IN low INT,  
    IN high INT  
)  
BEGIN  
    DECLARE counter INT DEFAULT 0;  
    REPEAT  
        SET counter := counter + 1;  
        -- thêm du lieu  
        INSERT INTO t(a,b)  
        VALUES(  
            ROUND((RAND() * (high-low))+high),  
            ROUND((RAND() * (high-low))+high)  
        );  
    UNTIL counter >= rowCount  
    END REPEAT;  
END$$
```

- Chạy lệnh sau để insert:

```
CALL insertSampleData(10000,1,1000);
```

# Descending Index trong MySQL

❖ **Yêu cầu 1:** Sắp xếp tăng dần theo hai cột a và b

```
EXPLAIN SELECT
    *
FROM
    t
ORDER BY a , b; -- sử dụng index a_asc_b_asc
```

id	select_type	table	partitions	type	possible_keys	key	key_len	ref	rows	filtered	Extra
1	SIMPLE	t	NULL	index	NULL	a_asc_b_asc	10	NULL	10157	100.00	Using index

❖ **Yêu cầu 2:** Sắp xếp cột a tăng dần, cột b giảm dần

```
EXPLAIN SELECT
    *
FROM
    t
ORDER BY a , b DESC; -- sử dụng index a_asc_b_desc
```

id	select_type	table	partitions	type	possible_keys	key	key_len	ref	rows	filtered	Extra
1	SIMPLE	t	NULL	index	NULL	a_asc_b_desc	10	NULL	10157	100.00	Using index



# Descending Index trong MySQL

❖ **Yêu cầu 3:** Sắp xếp cột a giảm dần, cột b tăng dần.

```
EXPLAIN SELECT
    *
FROM
    t
ORDER BY a DESC , b; -- sử dụng index a_desc_b_asc
```

id	select_type	table	partitions	type	possible_keys	key	key_len	ref	rows	filtered	Extra
1	SIMPLE	t	NULL	index	NULL	a_desc_b_asc	10	NULL	10157	100.00	Using index



# Invisible Index trong MySQL

- ❖ Invisible Index là trạng thái chỉ mục ẩn
- ❖ Muốn bật chỉ mục thì thay đổi trạng thái từ **INVISIBLE** sang **VISIBLE**.
- ❖ Cú pháp:
- ❖ Cú pháp thay đổi trạng thái của một chỉ mục từ **INVISIBLE** sang **VISIBLE** và ngược

```
CREATE INDEX index_name  
ON table_name( c1, c2, ...) INVISIBLE;
```

```
ALTER TABLE table_name  
ALTER INDEX index_name [VISIBLE | INVISIBLE];
```



# Invisible Index trong MySQL

❖ Ví dụ:

```
CREATE INDEX extension  
ON employees(extension) INVISIBLE;
```

```
ALTER TABLE employees  
ALTER INDEX extension VISIBLE;
```



# Invisible Index trong MySQL

## ❖ Kiểm tra trạng thái của chỉ mục

```
SELECT
    index_name,
    is_visible
FROM
    information_schema.statistics
WHERE
    table_schema = 'classicmodels'
    AND table_name = 'employees';
```

```
SHOW INDEXES FROM employees;
```

INDEX_NAME	IS_VISIBLE
extension	NO
officeCode	YES
PRIMARY	YES
reportsTo	YES

## ❖ Tạo bảng **discounts**

```
CREATE TABLE discounts (  
    discount_id INT NOT NULL,  
    name VARCHAR(50) NOT NULL,  
    valid_from DATE NOT NULL,  
    valid_to DATE NOT NULL,  
    amount DEC(5 , 2 ) NOT NULL DEFAULT 0,  
    UNIQUE discount_id(discount_id)  
);
```

## ❖ Đổi trạng thái **UNIQUE** sang **Invisible**

```
ALTER TABLE discounts  
ALTER INDEX discount_id INVISIBLE;
```

Error Code: 3522. A primary key index cannot be invisible

