



Manipulating data

Ba Nguyễn



Data types

Một số kiểu dữ liệu cơ bản:

- **CHAR(len)** - chuỗi có độ dài cố định
- **VARCHAR(len)** - chuỗi có độ dài linh động
- **TINYINT - UNSIGNED TINYINT** - 1 Byte (2^8)
- **INT - UNSIGNED INT** - 4 Bytes (2^{32})
- **DECIMAL(p, s)** - Số thực, p = số chữ số tối đa, s = số chữ số phần thập phân
- **BOOLEAN**, giá trị **TRUE, FALSE**
- **DATE, TIME, DATETIME**



Constraints

Constraints (ràng buộc) là các quy tắc được áp dụng cho các cột dữ liệu trong bảng.

Constraints sử dụng để kiểm tra tính hợp lệ của dữ liệu, đảm bảo tính tin cậy, chính xác, và toàn vẹn dữ liệu.

Dữ liệu sẽ không thể thêm vào CSDL nếu không phù hợp với các constraint được chỉ định cho cột



Constraints

NOT NULL

Ràng buộc đảm bảo dữ liệu một cột không được phép để trống

DEFAULT

Ràng buộc chỉ định một giá trị mặc định cho cột

ZEROFILL

Ràng buộc sử dụng với các cột số, hiển thị số lượng chữ số nhất định



Constraints

CHECK (exprssion)

Ràng buộc kiểm tra tính hợp lệ của dữ liệu, có thể sử dụng các hàm, toán tử, hay **REGEXP** với **CHECK**

UNIQUE

Ràng buộc đảm bảo giá trị cho một cột (hoặc tập hợp các cột) là duy nhất, chấp nhận giá trị **NULL**. Một bảng có thể có nhiều cột **UNIQUE**



Constraints

PRIMARY KEY

Ràng buộc áp dụng cho 1 hoặc tập hợp các cột trong bảng, **PRIMARY KEY** sử dụng để định danh dữ liệu cho các hàng trong bảng. Mỗi bảng chỉ có 1 **PRIMARY KEY** duy nhất và không cho phép giá trị **NULL**.

💡 Có thể sử dụng **AUTO_INCREMENT** để thiết lập giá trị tự động cho cột **PRIMARY KEY** có kiểu số

💡 Có thể tùy chỉnh giá trị cho **AUTO_INCREMENT** để bắt đầu từ một số cụ thể



Constraints

FOREIGN KEY

Ràng buộc áp dụng cho 1 hoặc tập hợp các cột để liên kết đến dữ liệu trong các bảng khác, thường là **PRIMARY KEY**. Một bảng có thể có nhiều FOREIGN KEY, liên kết đến nhiều bảng khác nhau.

Ngoài ra, **FOREIGN KEY** còn đặt các ràng buộc với dữ liệu từ các bảng liên quan, trong trường hợp các bảng liên quan có sự thay đổi về dữ liệu, điều này đảm bảo tính toàn vẹn dữ liệu.



Database

-- Tạo CSDL

CREATE DATABASE IF NOT EXISTS db_name;

Options: CHARACTER SET, COLLATE

-- Xóa CSDL

DROP DATABASE IF EXISTS db_name;



Table

-- Tạo bảng

```
CREATE TABLE IF NOT EXISTS table_name (  
    column_definition,  
    constraints  
) table_options;
```

-- Xóa bảng

```
DROP TABLE IF EXISTS table_name;
```



INSERT

Câu lệnh **INSERT** sử dụng để thêm mới một (hoặc nhiều) records vào một bảng

```
INSERT INTO table_name (col_name, col2_name)
```

```
VALUES
```

```
    (value1, value2),
```

```
    (value1, value2)
```

```
    # ...
```



UPDATE

Câu lệnh **UPDATE** được sử dụng để cập nhật dữ liệu trong một bảng, cũng có thể sử dụng **JOIN** để cập nhật dữ liệu trên cùng lúc nhiều bảng

```
UPDATE table_name  
SET column = value | expr  
    column = value | expr  
WHERE search_condition;
```



DELETE

Câu lệnh **DELETE** được sử dụng để xóa dữ liệu từ một hoặc nhiều bảng (JOIN)

```
DELETE FROM table_name  
WHERE search_condition;
```

```
-- Sử dụng truncate để xóa toàn bộ dữ liệu  
TRUNCATE table_name;ALTER
```