Trường ĐH Khoa Học Tự Nhiên Tp. Hồ Chí Minh
TRUNG TÂM TIN HỌC

DATABASE AND SQL FOR DATA SCIENCE

Basic SQL

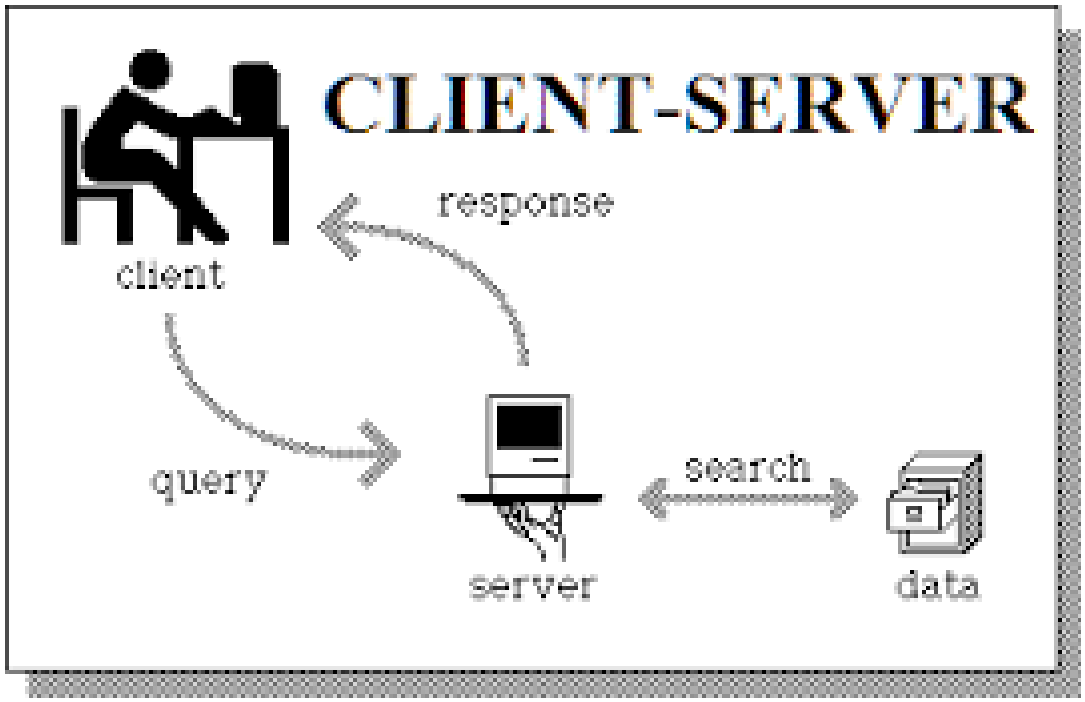
Phòng LT & Mạng

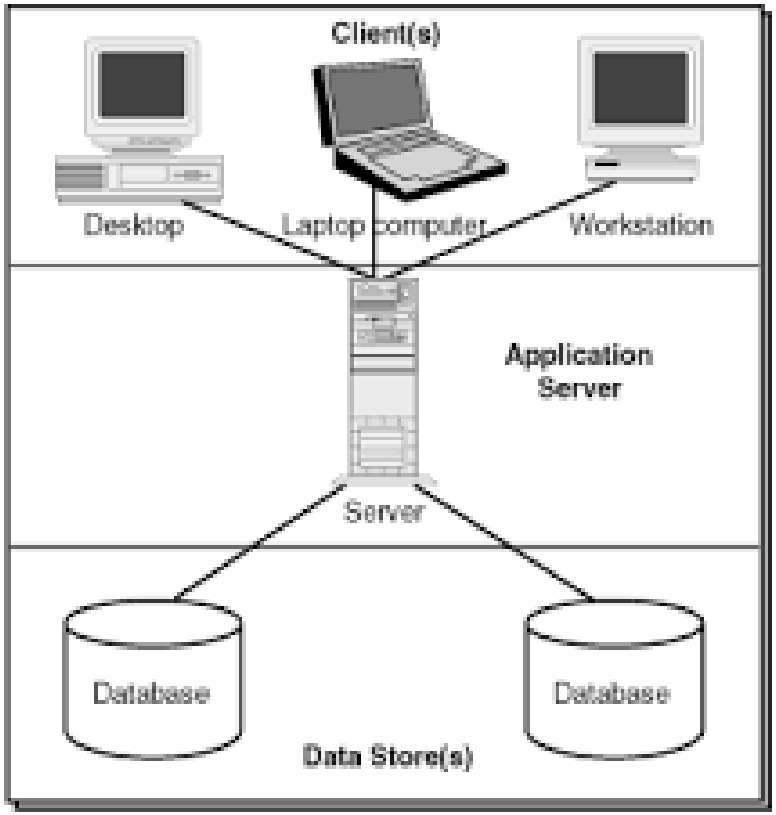
http://csc.edu.vn/kiem-thu-phan-mem

2019

Data Science

Mô hình Client-Server



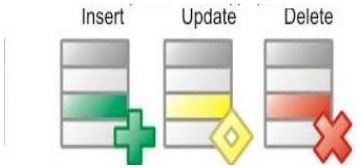


4 nhóm lệnh

Data Science

SQL (Structured Query Language)

DML (Data Manipulation Language)



DDL (Data Definition Language)

DCL (Data Control Language)

NỘI DUNG

1. CREATE/DROP TABLE ← DDL
2. SELECT ← SQL
3. COUNT/DISTINCT/LIMIT ← SQL
4. INSERT ← DML
5. UPDATE ← DML
6. DELETE ← DML
7. CONSTRAINT
8. WHERE AND/OR ← SQL

CREATE/DROP TABLE

- CREATE TABLE: Dùng để tạo cấu trúc bảng dữ liệu
- Cú pháp chung:

```
create table TABLENAME (
    COLUMN1 datatype,
    COLUMN2 datatype,
    COLUMN3 datatype,
        ...
    ) ;
```

CREATE/DROP TABLE

Data Science

- Các kiểu dữ liệu (data type) thường dùng:
  - Kiểu số
    - SMALLINT: số nguyên 2 bytes, từ -32768 to +32767
    - INTEGER hoặc INT: số nguyên 4 bytes, từ -2147483648 to +2147483647
    - BIGINT: số nguyên 8 bytes, từ -9223372036854775808 to +9223372036854775807.
    - FLOAT: số thực 4 bytes, từ -7.2E+75 to 7.2E+75
    - DOUBLE: số thực 8 bytes

CREATE/DROP TABLE

- Các kiểu dữ liệu (data type) thường dùng:
  - Kiểu chuỗi
    - CHAR(n): Chuỗi có độ dài cố định n bytes (với n từ 1 → 255)
    - VARCHAR(n): Chuỗi có độ dài khác nhau với tối đa n bytes (với n từ 1→ 32704)
    - BINARY(n): Chuỗi nhị phân có độ dài cố định n bytes (với n từ 1 → 255)
    - VARBINARY(n): Chuỗi nhị phân có độ dài khác nhau với tối đa n bytes (với n từ 1→ 32704)

CREATE/DROP TABLE

- Các kiểu dữ liệu (data type) thường dùng:
  - Kiểu ngày giờ
    - DATE: Kiểu ngày với 3 giá trị năm, tháng, ngày (từ 0001-01-01 đến 9999-12-31)
    - TIME: Kiểu giờ với 3 giá trị giờ, phút, giây (từ 00.00.00 đến 24.00.00)
    - TIMESTAMP: Kiểu ngày giờ với bảy giá trị năm, tháng, ngày, giờ, phút, giây và micro giây (từ 0001-01-01-00.00.00.000000000 đến 9999-12-31-24.00.00.000000000)

CREATE/DROP TABLE

- Các kiểu dữ liệu (data type) thường dùng:
  - Kiểu dữ liệu lớn (LOB large object)
    - CLOB(n): Chuỗi có độ dài khác nhau với tối đa n ký tự (n < 2.147.483.647)
    - DBCLOB(n): Chuỗi các ký tự 2 bytes có độ dài khác nhau với tối đa n ký tự (n < 1.073.741.824)
    - BLOB(n): Chuỗi nhị phân có độ dài khác nhau với tối đa n ký tự (n < 2.147.483.647)

CREATE/DROP TABLE

- Ví dụ 1:
  - Tạo bảng COUNTRY với 3 cột: ID, CountryCode và CountryName

```
create table COUNTRY (
    ID int,
    CountryCode char(3),
    CountryName varchar(60)
);
```

CREATE/DROP TABLE

Data Science

- Ví dụ 2:
  - Tạo bảng COUNTRY với 3 cột: ID, CountryCode và CountryName. Trong đó ID là khóa chính (primary key)

```
create table COUNTRY (
    ID int,
    CountryCode char(3),
    CountryName varchar(60),
    PRIMARY KEY (ID)
);
```

CREATE/DROP TABLE

- Ví dụ 2:
  - Tạo bảng COUNTRY với 3 cột: ID, CountryCode và CountryName. Trong đó ID là khóa chính (primary key)

```
create table COUNTRY (
    ID integer PRIMARY KEY NOT NULL,
    CountryCode char(3) NOT NULL,
    CountryName varchar(60) NOT NULL
);
```

CREATE/DROP TABLE

- DROP TABLE: Dùng xóa bảng dữ liệu
- Cú pháp chung:

```
Drop table TABLENAME;
```

- Ví dụ: Xóa bảng COUNTRY

```
Drop table COUNTRY;
```

CREATE/DROP TABLE

- Lưu ý:
  - Not Null: dữ liệu của cột không được phép Null
  - Tên bảng và Tên cột không có khoảng trắng và các ký tự đặc biệt, không phân biệt hoa thường.
  - Tên bảng và Tên cột phải được bắt đầu bằng ký tự chữ hoặc _

CREATE/DROP TABLE

- Một số lỗi thường gặp
  - Bảng đã tồn tại khi tạo bảng, ví dụ:
    - Table 'COUNTRY' already exists
    - The name of the object to be created is identical to the existing name "BMR71783. COUNTRY“
  - Bảng không tồn tại khi xóa bảng, ví dụ:
    - Unknown table 'TestDB.COUNTRY‘
    - "BMR71783.COUNTRY" is an undefined name.

NỘI DUNG

1. CREATE/DROP TABLE
2. SELECT
3. COUNT/DISTINCT/LIMIT
4. INSERT
5. UPDATE
6. DELETE
7. CONSTRAINT
8. WHERE AND/OR

SELECT

- Dùng để truy vấn (lựa chọn) dữ liệu từ các bảng dữ liệu
- Cú pháp chung:

```
select *| COLUMN1, COLUMN2, ...| EXPRESSION1, EXPRESSION2, …
from TABLENAME;

select COLUMN1, COLUMN2, ...
from TABLENAME
where CONDITION ;
```

SELECT

Data Science

- Ví dụ 1:
  - Liệt kê danh sách tất cả các Quốc gia bao gồm Id và Tên của mỗi Quốc gia

```
select ID, NAME from COUNTRY;
```

  - Liệt kê danh sách tất cả các Quốc gia bao gồm tất cả thông tin của mỗi Quốc gia

```
select * from COUNTRY;
```

SELECT

- Ví dụ 2:
  - Liệt kê danh sách tất cả các Quốc gia có Id nhỏ hơn hoặc bằng 5

```
select *
from COUNTRY
where ID <= 5;
```

SELECT

Data Science

- Ví dụ 3:
  - Lựa chọn thông tin Quốc gia có CountryCode là VN

```
select *
from COUNTRY
where CountryCode = 'VIE';
```

SELECT

Data Science

- Lưu ý:
  - Dùng ký tự * để thay thế cho tất cả các cột (Column) trong bảng
  - Có thể đặt bí danh cho cột thông qua từ khóa As

```
select ID, NAME as TenQuocGia
from COUNTRY;
```

  - Các phép toán so sánh trong mệnh đề điều kiện
    - =, >, >=, <, <=, != (hoặc <>)
    - Between … and…
    - IN (list)

NỘI DUNG

1. CREATE/DROP TABLE
2. SELECT
3. **COUNT/DISTINCT/LIMIT**
4. INSERT
5. UPDATE
6. DELETE
7. CONSTRAINT
8. WHERE AND/OR

COUNT/DISTINCT/LIMIT

- Dùng để tính toán trong câu lệnh truy vấn dữ liệu từ các bảng dữ liệu
- COUNT: Dùng để đếm số lượng mẩu tin (dòng) được truy vấn

```
select COUNT(*)
from tablename
```

COUNT/DISTINCT/LIMIT

- Ví dụ 1:
  - Cho biết tổng cộng có bao nhiêu Quốc gia trong bảng COUNTRY

```
select COUNT(*)
from COUNTRY;
```

COUNT/DISTINCT/LIMIT

- Ví dụ 2:
  - Cho biết có bao nhiêu Tỉnh/TP của Quốc gia có Code là VN

```
select COUNT(*)
from CITIES
where CountryCode = 'VIE';
```

COUNT/DISTINCT/LIMIT

- DISTINCT: Dùng loại bỏ các dòng dữ liệu trùng trong câu lệnh truy vấn

```
select DISTINCT columnname
from tablename
```

COUNT/DISTINCT/LIMIT

- Ví dụ: xét bảng dữ liệu MEDALS như sau:

```
Create TABLE MEDALS (
    Id Int PRIMARY KEY,
    Year Int,
    City Varchar(30),
    Sport Varchar(30),
    Discipline Varchar(30),
    Athlete Varchar(60),
    CountryCode Char(3),
    Gender Varchar(5),
    Event Varchar(80),
    Medal Varchar(8)
)
```

COUNT/DISTINCT/LIMIT

Data Science

- Ví dụ: xét bảng dữ liệu MEDALS như sau:

Id	Year	City	Sport	Discipline	Athlete	CountryCode	Gender	Event	Medal
1	1896	Athens	Aquatics	Swimming	HAJOS, Alfred	HUN	Men	100M Freestyle	Gold
2	1896	Athens	Aquatics	Swimming	HERSCHMANN, Otto	AUT	Men	100M Freestyle	Silver
3	1896	Athens	Aquatics	Swimming	DRIVAS, Dimitrios	GRE	Men	100M Freestyle For Sailors	Bronze
4	1896	Athens	Aquatics	Swimming	MALOKINIS, Ioannis	GRE	Men	100M Freestyle For Sailors	Gold
5	1896	Athens	Aquatics	Swimming	CHASAPIS, Spiridon	GRE	Men	100M Freestyle For Sailors	Silver
6	1896	Athens	Aquatics	Swimming	CHOROPHAS, Efstathios	GRE	Men	1200M Freestyle	Bronze
7	1896	Athens	Aquatics	Swimming	HAJOS, Alfred	HUN	Men	1200M Freestyle	Gold
8	1896	Athens	Aquatics	Swimming	ANDREOU, Joannis	GRE	Men	1200M Freestyle	Silver
9	1896	Athens	Aquatics	Swimming	CHOROPHAS, Efstathios	GRE	Men	400M Freestyle	Bronze
10	1896	Athens	Aquatics	Swimming	NEUMANN, Paul	AUT	Men	400M Freestyle	Gold

COUNT/DISTINCT/LIMIT

- Ví dụ 1: liệt kê danh sách các Quốc gia đạt huy chương vàng

```
select DISTINCT CountryCode
from MEDALS
where MEDAL = 'GOLD'
```

COUNT/DISTINCT/LIMIT

- Ví dụ 2: liệt kê tên các vận động viên đã từng giành được huy chương vàng

```
select DISTINCT Athlete
from MEDALS
where MEDAL = 'GOLD'
```

COUNT/DISTINCT/LIMIT

- LIMIT: Dùng giới hạn số dòng dữ liệu trong câu lệnh truy vấn

```
select *
from tablename
LIMIT Number_of_rows
```

COUNT/DISTINCT/LIMIT

- Ví dụ: liệt kê 10 Quốc gia đầu tiên có trong bảng COUNTRY

```
select *
from COUNTRY
LIMIT 10
```

NỘI DUNG

1. CREATE/DROP TABLE
2. SELECT
3. COUNT/DISTINCT/LIMIT
4. **INSERT**
5. UPDATE
6. DELETE
7. CONSTRAINT
8. WHERE AND/OR

INSERT

- Dùng để thêm mẩu tin (dòng dữ liệu) mới vào bảng dữ liệu
- Cú pháp chung:

```
Insert into tablename [([ColumnName1],
            [ColumnName2],…)]

Values([Value1], [Value2],…);
```

INSERT

- Ví dụ:

```
Insert into COUNTRY
Values(2, 'USA', 'United State');

Insert into COUNTRY (Id, CountryCode, CountryName)
Values(1, 'VIE', 'Việt Nam');

Insert into COUNTRY (CountryCode, CountryName)
Values('FRA', 'Pháp');
```

INSERT

- Có thể thêm nhiều dòng dữ liệu đồng thời trong một lệnh Insert
- Ví dụ:

```
Insert into COUNTRY (Id, CountryCode,
                                    CountryName)
Values
    (1, 'VIE', 'Việt Nam'),
    (2, 'USA', 'United State');
```

- Lưu ý thứ tự các cột khi sử dụng lệnh Insert

NỘI DUNG

1. CREATE/DROP TABLE
2. SELECT
3. COUNT/DISTINCT/LIMIT
4. INSERT
5. **UPDATE**
6. DELETE
7. CONSTRAINT
8. WHERE AND/OR

UPDATE

- Dùng để cập nhật (sửa) mẩu tin (dòng dữ liệu) trong bảng dữ liệu
- Cú pháp chung:

```
Update tablename
Set [[ColumnName1]=[Value1],
     [ColumnName1]=[Value1],… ]
[Where <Condition>];
```

UPDATE

- Ví dụ:

```
Update COUNTRY
Set CountryName = 'United State of American'
Where CountryCode = 'USA';
```

NỘI DUNG

1. CREATE/DROP TABLE
2. SELECT
3. COUNT/DISTINCT/LIMIT
4. INSERT
5. UPDATE
6. **DELETE**
7. CONSTRAINT
8. WHERE AND/OR

DELETE

- Dùng để xóa mẩu tin (dòng dữ liệu) trong bảng dữ liệu
- Cú pháp chung:

```
Delete From tablename
[Where <Condition>];
```

DELETE

- Ví dụ: xóa quốc gia có mã code là USA

```
Delete From COUNTRY
Where CountryCode = 'USA';
```

NỘI DUNG

1. CREATE/DROP TABLE
2. SELECT
3. COUNT/DISTINCT/LIMIT
4. INSERT
5. UPDATE
6. DELETE
7. **CONSTRAINT**
8. WHERE AND/OR

CONSTRAINT

- Constraint là các qui tắc được áp đặt cho các cột dữ liệu trên table.
- Constraint được sử dụng để giới hạn kiểu dữ liệu nhập vào một bảng. Điều này đảm bảo tính chính xác và tính đáng tin cậy cho dữ liệu trong Database.

CONSTRAINT

- Constraint có 2 cấp độ:
  - Column Level: chỉ được áp dụng cho cột
  - Table Level: áp dụng cho toàn bộ table

- Có thể khai báo constraint trong câu lệnh CREATE TABLE (tạo mới bảng) hoặc ALTER TABLE (Sửa đổi bảng)

CONSTRAINT

- Các loại Constraint phổ biến:
  - NOT NULL: Bảo đảm một cột không thể có giá trị NULL.
  - DEFAULT: Cung cấp một giá trị mặc định cho cột khi không được xác định.
  - UNIQUE: Bảo đảm tất cả giá trị trong một cột là khác nhau.
  - PRIMARY KEY: Xác định giá trị trên tập các cột làm khóa chính phải là duy nhất, không được trùng lặp. Việc khai báo ràng buộc khóa chính yêu cầu các cột phải NOT NULL.

CONSTRAINT

- Các loại Constraint phổ biến:
  - FOREIGN KEY: Dùng để thiết lập khóa ngoại trên bảng, tham chiếu đến bảng khác thông qua giá trị của cột được liên kết. Giá trị của cột được liên kết phải là duy nhất trong bảng kia.
  - CHECK: Bảo đảm tất cả giá trị trong cột thỏa mãn điều kiện nào đó

CONSTRAINT

- Ví dụ 1: thiết lập ràng buộc NOT NULL trên cột CountryName khi tạo Table

```
create table COUNTRY (
    ID int,
    CountryCode char(3),
    CountryName varchar(60) NOT NULL,
    ModifyDate date
);
```

CONSTRAINT

- Ví dụ 2: thiết lập ràng buộc NOT NULL trên cột CountryName khi sửa đổi Table

```
alter table COUNTRY
    MODIFY CountryName varchar(60) NOT NULL;
```

CONSTRAINT

- Ví dụ 3: thiết lập ràng buộc DEFAULT lấy giá trị mặc định là ngày hiện tại trên cột ModifyDate

```
create table COUNTRY (
    ID int,
    CountryCode char(3),
    CountryName varchar(60) NOT NULL,
    ModifyDate TimeStamp Default CURRENT_TIMESTAMP
);
```

CONSTRAINT

- Ví dụ 4: thiết lập ràng buộc DEFAULT lấy giá trị mặc định là ngày hiện tại trên cột ModifyDate khi sửa đổi table

```
alter table COUNTRY
   Modify ModifyDate Timestamp Default
                              CURRENT_TIMESTAMP;
```

CONSTRAINT

- Ví dụ 5: thiết lập ràng buộc PRIMARY KEY cho cột ID (Định nghĩa trực tiếp khi khai báo cột – Constraint mức cột)

```
create table COUNTRY (
    ID int PRIMARY KEY,
    CountryCode char(3),
    CountryName varchar(60) NOT NULL,
    ModifyDate date
);
```

CONSTRAINT

- Ví dụ 6: thiết lập ràng buộc PRIMARY KEY cho cột ID (Định nghĩa constraint mức bảng)

```
create table COUNTRY (
    ID int,
    CountryCode char(3),
    CountryName varchar(60) NOT NULL,
    ModifyDate date,
    CONSTRAINT pk_id PRIMARY KEY (ID)
);
```

CONSTRAINT

- Ví dụ 7: thiết lập ràng buộc PRIMARY KEY cho cột ID (Thiết lập ràng buộc khi sửa đổi table)

```
alter table COUNTRY
   Add Constraint pk_id PRIMARY KEY (ID);
```

CONSTRAINT

- Ví dụ 8: thiết lập ràng buộc FOREIGN KEY cho cột CountryID (Định nghĩa trực tiếp khi khai báo cột)

```
Create TABLE MEDALS (
    Id Int PRIMARY KEY,
    Year Int,
    City Varchar(30),
    Sport Varchar(30),
    Discipline Varchar(30),
    Athlete Varchar(60),
    CountryID Int FOREIGN KEY REFERENCES Country(ID),
    CountryCode Char(3),
    Gender Varchar(5),
    Event Varchar(80),
    Medal Varchar(8)
)
```

CONSTRAINT

- Ví dụ 9: thiết lập ràng buộc FOREIGN KEY cho cột CountryID (Định nghĩa constraint)

```
Create TABLE MEDALS (
    Id Int PRIMARY KEY,
    Year Int,
    City Varchar(30),
    Sport Varchar(30),
    Discipline Varchar(30),
    Athlete Varchar(60),
    CountryID Int,
    CountryCode Char(3),
    Gender Varchar(5),
    Event Varchar(80),
    Medal Varchar(8),
    CONSTRAINT FK_CountryID FOREIGN KEY (CountryID)
                                REFERENCES Country(ID)
)
```

CONSTRAINT

- Ví dụ 10: thiết lập ràng buộc FOREIGN KEY cho cột CountryID (Thiết lập ràng buộc khi sửa đổi table)

```
alter table MEDALS
    Add Constraint FK_CountryID FOREIGN KEY
            (CountryID) REFERENCES Country(ID);
```

CONSTRAINT

- Ví dụ 11: thiết lập ràng buộc CHECK cho các cột của bảng VayNo (Định nghĩa trực tiếp khi khai báo cột):

```
CREATE TABLE Vay (
    MaVay char(10) NOT NULL,
    MaKH char(10),
    MaTaiSan char(10),
    MaNV char(10),
    NgayVay date,
    ThoiHan int,
    LaiSuat float,
    SoTienVay float Check (SoTienVay > 0),
    NgayHetHan date
);
```

CONSTRAINT

- Ví dụ 12: thiết lập ràng buộc CHECK cho các cột của bảng VayNo (khi sửa đổi Table):
  - Tạo ràng buộc check trên cột SoTienVay > 0

```
ALTER TABLE VayNo
   ADD CONSTRAINT check_SoTienVay
                          CHECK (SoTienVay > 0);
```

  - Tạo ràng buộc check trên cột NgayHetHan phải lớn hơn NgayVay

```
ALTER TABLE VayNo
   ADD CONSTRAINT check_ngayhethan
                       CHECK (NgayHetHan > NgayVay);
```

CONSTRAINT

- Ví dụ 12: thiết lập ràng buộc CHECK cho các cột của bảng VayNo (khi sửa đổi Table):
  - Tạo ràng buộc kiểm tra trên cột ThoiHan nằm trong khoảng 1 đến 36 tháng

```
ALTER TABLE VayNo
    ADD CONSTRAINT check_thoihan
                CHECK (ThoiHan BETWEEN 1 AND 36);
```

CONSTRAINT

Data Science

❑ Xóa bỏ Constraint:

- Cú pháp chung

```
ALTER TABLE <tên table chứa ràng buộc>
    DROP CONSTRAINT <tên ràng buộc muốn xóa>
```

- Ví dụ: xóa bỏ ràng buộc check_SoTienVay trong bảng VayNo

```
ALTER TABLE VayNo
    DROP CONSTRAINT check_SoTienVay;
```

CONSTRAINT

Data Science

- Xóa bỏ Constraint:
  - Áp dụng MariaDB 10.x.x và MySQL 8.x.x
  - Xóa Primary Key

```
ALTER TABLE <tên table chứa ràng buộc>
    DROP PRIMARY KEY;
```

  - Ví dụ: xóa bỏ primary key trong bảng Country

```
ALTER TABLE Country
    DROP PRIMARY KEY;
```

CONSTRAINT

- Xóa bỏ Constraint:
  - Áp dụng MariaDB 10.x.x và MySQL 8.x.x
  - Xóa Foreign Key

```
ALTER TABLE <tên table chứa ràng buộc>
    DROP FOREIGN KEY <tên foreign key>;
```

  - Ví dụ: xóa bỏ foreign key frk_medal_country trong bảng Medals

```
ALTER TABLE Medals
    DROP FOREIGN KEY frk_medal_country;
```

CONSTRAINT

Data Science

Xóa bỏ Constraint:

- Áp dụng MariaDB 10.x.x và MySQL 8.x.x
- Xóa các constraint theo tên

```
ALTER TABLE <tên table chứa ràng buộc>
  DROP <Loại Constraint> <Tên Constraint>;
```

- Ví dụ: xóa bỏ check constraint thời hạn trong bảng VayNo

```
ALTER TABLE VayNo
  DROP CHECK check_thoihan;
```

CONSTRAINT

- Kích hoạt/Bỏ kích hoạt constraint:
  - Bỏ kích hoạt

```
ALTER TABLE <tên table chứa ràng buộc>
  NOCHECK CONSTRAINT <tên ràng buộc muốn
                      bỏ kích hoạt>
```

  - Kích hoạt

```
ALTER TABLE <tên table chứa ràng buộc>
  WITH CHECK CHECK CONSTRAINT <tên ràng
                      buộc muốn kích hoạt>
```

CONSTRAINT

- Ví dụ:
  - Bỏ kích hoạt ràng buộc trên cột SoTienVay

```
ALTER TABLE VayNo
    NOCHECK CONSTRAINT check_SoTienVay;
```

  - Kích hoạt lại ràng buộc trên cột SoTienVay

```
ALTER TABLE VayNo
    WITH CHECK CHECK CONSTRAINT check_SoTienVay;
```

CONSTRAINT

Data Science

- Kích hoạt/Bỏ kích hoạt constraint:
  - Áp dụng MariaDB 10.x.x và MySQL 8.x.x
  - Bỏ kích hoạt check constraint

```
ALTER TABLE <tên table chứa ràng buộc>
  ALTER CHECK <tên constraint> NOT ENFORCED;
```

  - Ví dụ: bỏ kích hoạt check check_SoTienVay trong bảng VayNo

```
ALTER TABLE VayNo
  ALTER CHECK check_SoTienVay NOT ENFORCED;
```

CONSTRAINT

- Kích hoạt/Bỏ kích hoạt constraint:
  - Áp dụng MariaDB 10.x.x và MySQL 8.x.x
  - Bỏ kích hoạt các constraint khác

```
ALTER TABLE <tên table chứa ràng buộc>
    ALTER CONSTRAINT<tên constraint> NOT ENFORCED;
```

  - Ví dụ: bỏ kích hoạt FK_CountryID trong bảng Medals

```
ALTER TABLE Medals
    ALTER CONSTRAINT FK_CountryID NOT ENFORCED;
```

CONSTRAINT

- Kích hoạt/Bỏ kích hoạt constraint:
  - Áp dụng MariaDB 10.x.x và MySQL 8.x.x
  - kích hoạt lại check constraint

```
ALTER TABLE <tên table chứa ràng buộc>
  ALTER CHECK <tên constraint> ENFORCED;
```

  - Ví dụ: kích hoạt lại check check_SoTienVay trong bảng VayNo

```
ALTER TABLE VayNo
  ALTER CHECK check_SoTienVay ENFORCED;
```

CONSTRAINT

Data Science

- Kích hoạt/Bỏ kích hoạt constraint:
  - Áp dụng MariaDB 10.x.x và MySQL 8.x.x
  - Kích hoạt lại các constraint khác

```
ALTER TABLE <tên table chứa ràng buộc>
    ALTER CONSTRAINT<tên constraint> ENFORCED;
```

  - Ví dụ: kích hoạt lại FK_CountryID trong bảng Medals

```
ALTER TABLE Medals
    ALTER CONSTRAITN FK_CountryID ENFORCED;
```

NỘI DUNG

1. CREATE/DROP TABLE
2. SELECT
3. COUNT/DISTINCT/LIMIT
4. INSERT
5. UPDATE
6. DELETE
7. CONSTRAINT
8. **WHERE AND/OR**

WHERE AND/OR

- Dùng để giới hạn số lượng mẩu tin (dòng dữ liệu) trả về trong câu truy vấn
  - **AND**: phối hợp hai biểu thức so sánh, kết quả trả về TRUE khi cả hai biểu thức so sánh đều trả về TRUE
  - **OR**: phối hợp hai biểu thức so sánh, kết quả trả về FALSE khi cả hai biểu thức so sánh đều trả về FALSE

WHERE AND/OR

- Ví dụ 1: cho biết danh sách các vận động viên đoạt huy chương vàng năm 2012

```
select Athlete
from MEDALS
where Medal = 'GOLD' AND Year = 2012
```

WHERE AND/OR

- Ví dụ 2: cho biết danh sách các vận động viên đoạt huy chương bạc năm 2012 và 2016

```
select DISTINCT Athlete
from MEDALS
where Medal = 'Silver' AND
              (Year = 2012 OR Year = 2016)
```