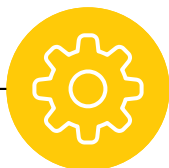
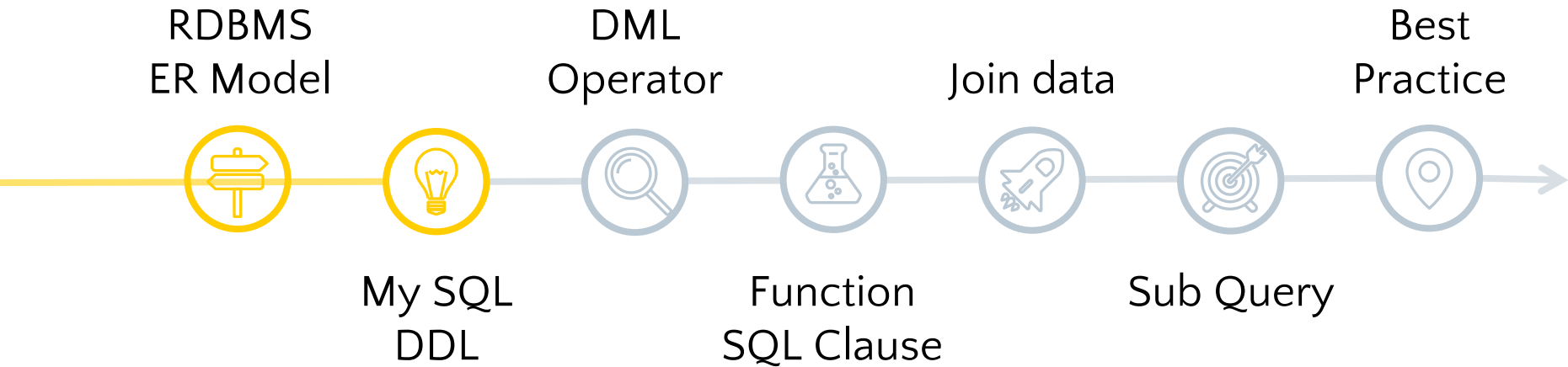


Welcome back



SQL *Essentials*

Roadmap



Previous lecture

- ER Model
- Entity
- Cardinality
- Relationships
- Convert ER model to schema
- Database
- Relational database
- DBMS vs RDBMS
- Schema

What we will explore today?

Data Definition Language

- Fun with database
- SQL data type
- Table In database
- Meaningful data with constraints
- SQL Process

MySQL Workbench

- Structure query language
- Take a look on UI
- First command
- SQL Components

- MySQL Server is an open-source relational database management system (RDBMS) developed by Oracle Corporation.
- It is one of the most popular and widely used database systems in the world.

What is SQL

- SQL stands for Structured Query Language. It's use to **store, manipulate, retrieve data**

SQL data type

INTEGER datatype

BIT(size)

- From 1 to 64
- Default value for **size** is 1

TINYINT(size)

- Signed: from **128** to **127**
- Unsigned: from **0** to **255**)

SMALLINT(size)

- Signed: from **-32768** to **32767**
- Unsigned: from **0** to **65535**)

MEDIUMINT(size)

- Signed: from **-8388608** to **8388607**
- Unsigned: from **0** to **16777215**

INT(size)

- Signed: from **-2147483648** to **2147483647**
- Unsigned range: from **0** to **4294967295**

Example

```
CREATE TABLE EmployeeData (  
    EmployeeID INT(10) UNSIGNED  
    AUTO_INCREMENT PRIMARY KEY,  
    FirstName VARCHAR(50),  
    LastName VARCHAR(50),  
    Age SMALLINT(5),  
    Department MEDIUMINT(8),  
    ExperienceYears INT(10)  
);
```

```
CREATE TABLE SensorData (  
    SensorID INT UNSIGNED  
    AUTO_INCREMENT PRIMARY KEY,  
    SensorName VARCHAR(100),  
    IsActive BIT(1),  
    SensorValue TINYINT  
);
```

```
CREATE TABLE ProductInventory (  
    ProductID INT UNSIGNED AUTO_INCREMENT PRIMARY KEY,  
    ProductName VARCHAR(100),  
    StockQuantity INT,  
    SalesCount MEDIUMINT,  
    Revenue BIGINT  
);
```

FLOAT and DOUBLE datatype

Float(size, d)

- A floating point number
- **size**: the total number of digits.
- **d**: the number of digits after the decimal point.

Double(size, d)

- A normal-size floating point number.
- **size**: the total number of digits.
- **d**: the number of digits after the decimal point.

Example

```
CREATE TABLE Measurements (  
    MeasurementID INT PRIMARY KEY,  
    Value FLOAT(10, 2)  
);
```

```
CREATE TABLE Prices (  
    ProductID INT,  
    UnitPrice DOUBLE(10, 4)  
);
```

```
CREATE TABLE FinancialData (  
    TransactionID INT,  
    Amount DECIMAL(12, 2)  
);
```

```
INSERT INTO FinancialData (TransactionID,  
Amount) VALUES (1, 1234.56),  
(2, 789.99),  
(3, 4567.33),  
(4, 10000.00),  
(5, 250.75);
```

DECIMAL datatype

- DECIMAL(size, d)
- An exact fixed-point number.
- size: The total number of digits
- d: The number of digits after the decimal point.
- The maximum number: for *size* is 65, for *d* is 30. The default value: for *size* is 10, for *d* is 0

```
CREATE TABLE FinancialData (  
    TransactionID INT,  
    Amount DECIMAL(12, 2)  
);
```

```
INSERT INTO FinancialData (TransactionID,  
Amount) VALUES (1, 1234.56),  
(2, 789.99),  
(3, 4567.33),  
(4, 10000.00),  
(5, 250.75);
```

String data types

CHAR(size)

- A FIXED length string, can contain letters, numbers, and special characters.
- **Size:** specifies the column length in characters - can be from 0 to 255. Default is 1

VARCHAR(size)

- A VARIABLE length string, can contain letters, numbers, and special characters.
- **size:** specifies the maximum column length in characters - can be from 0 to 65535

TINYTEXT(size)

- Holds a string with a maximum length of 255 characters

TEXT(size)

- Holds a string with a maximum length of 65,535 bytes

ENUM datatype

- `ENUM(val1, val2, val3, ...)`: A string object that can have only one value, chosen from a list of possible values.
- You can list up to 65535 values in an ENUM list.
- If a value is inserted that is not in the list, a blank value will be inserted.
- The values are sorted in the order you enter them

```
CREATE TABLE Students (  
    StudentID INT PRIMARY KEY,  
    FirstName VARCHAR(50),  
    LastName VARCHAR(50),  
    Gender ENUM('Male', 'Female', 'Other')  
);
```

```
-- Inserting data into the Students table  
INSERT INTO Students (  
    StudentID, FirstName, LastName, Gender)  
VALUES  
    (1, 'John', 'Doe', 'Male'),  
    (2, 'Jane', 'Smith', 'Female'),  
    (3, 'Alex', 'Taylor', 'Other');
```

NVARCHAR

- NVARCHAR is used to store variable-length Unicode character strings.
- It's particularly useful when you need to store multilingual data or data containing characters from different scripts.

```
CREATE TABLE Users (  
    UserID INT PRIMARY KEY,  
    FirstName NVARCHAR(50),  
    LastName NVARCHAR(50)  
);
```

```
-- Inserting data into the Users table with Unicode characters  
INSERT INTO Users (UserID, FirstName, LastName)  
VALUES  
    (1, N'John 🌟', N'Doe 🌟'),  
    (2, N'Jane 🌸', N'Smith 🌸'),  
    (3, N'Alex ☐', N'Taylor ☐);
```

Why use Unicode character?

- Single Byte Representation:
 - A single byte can represent 256 different values.
 - It can represent 256 characters or symbols in single-byte encoding.
- ASCII Encoding:
 - ASCII uses a single byte to represent English characters, digits, and some symbols.
 - It results in 256 possible values.
- Unicode Encoding:
 - Unicode provides a much larger character set than single-byte encodings.
 - Some Unicode encodings use multiple bytes per character, representing a vast range of characters from many languages and scripts.

Date and Time datatypes

DATETIME(fsp)

- Format: YYYY-MM-DD hh:mm:ss.
- Range is from '1000-01-01 00:00:00' to '9999-12-31 23:59:59'

TIMESTAMP(fsp)

- Values are stored as the number of seconds since the Unix epoch ('1970-01-01 00:00:00' UTC).
- Format: YYYY-MM-DD hh:mm:ss.
- Range: from '1970-01-01 00:00:01' UTC to '2038-01-09 03:14:07' UTC

TIME(fsp)

- Format: hh:mm:ss. The supported range is from '-838:59:59' to '838:59:59'

DATE

- Format: YYYY-MM-DD.
- Range is from '1000-01-01' to '9999-12-31'

YEAR

- A year in four-digit format.
- Four-digit format: 1901 to 2155, and 0000.

Example

```
CREATE TABLE Events (  
    EventID INT PRIMARY KEY,  
    EventName VARCHAR(100),  
    EventDate DATE,  
    EventTime TIME,  
    EventDateTime DATETIME,  
    CreatedTimestamp TIMESTAMP DEFAULT CURRENT_TIMESTAMP,  
    EventYear YEAR  
);
```

```
-- Inserting data into the Events table  
INSERT INTO Events (EventID, EventName, EventDate, EventTime, EventDateTime,  
EventYear)  
VALUES  
    (1, 'Birthday Party', '2023-10-15', '15:00:00', '2023-10-15 15:00:00', '2023'),  
    (2, 'Conference', '2023-11-05', '09:30:00', '2023-11-05 09:30:00', '2023'),  
    (3, 'Meeting', '2023-10-20', '14:15:00', '2023-10-20 14:15:00', '2023');
```

SQL Main Components

DDL (Data
Definition
Language)

used to define **data structures**: database, table, column, relationships, constraints etc..

DML (Data
Manipulation
Language)

used for **insert, delete, update** data in a DB

DCL (Data
Control
Language)

used to **control access** to data stored in a DB



Data Definition Language (DDL)



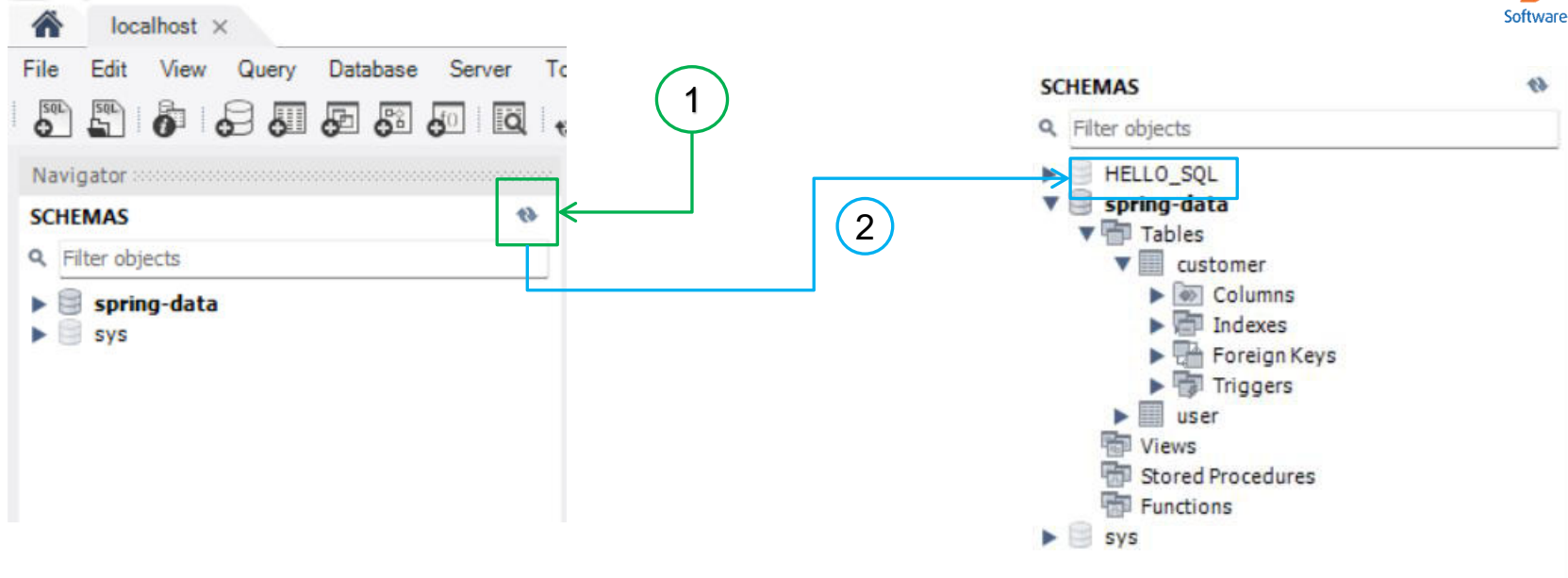
Database

Create database syntax

CREATE DATABASE data_base_name

EX1: CREATE DATABASE HELLO_SQL

Output					
Action Output					
#	Time	Action	Message	Duration / Fetch	
✓ 1	14:12:48	CREATE DATABASE HELLO_SQL	1 row(s) affected	0.046 sec	



SCHEMAS

Filter objects

HELLO_SQL

▼ spring-data

▼ Tables

customer

▶ Columns

▶ Indexes

▶ Foreign Keys

▶ Triggers

user

Views

Stored Procedures

Functions

▶ sys

Load Spatial Data

Set as Default Schema

Filter to This Schema

Schema Inspector

Table Data Import Wizard

Copy to Clipboard

Send to SQL Editor

Create Schema...

Alter Schema...

Drop Schema...

Search Table Data...

Refresh All

1

2

SCHEMAS

Filter objects

▼ HELLO_SQL

Tables

Views

Stored Procedures

Functions

▼ spring-data

▼ Tables

customer

▶ Columns

▶ Indexes

▶ Foreign Keys

▶ Triggers

user

Views

Stored Procedures

Functions

▶ sys

USE db_name;

Run the command again

Output				
Action Output				
#	Time	Action	Message	Duration / Fetch
✖ 1	14:10:33	CREATE DATABASE HELLO_SQL	Error Code: 1007. Can't create database 'HEL...	0.031 sec

Drop database

DROP DATABASE data_base_name

EX: DROP DATABASE HELLO_SQL

Output					
Action Output					
#	Time	Action	Message	Duration / Fetch	
✓ 1	15:50:31	DROP DATABASE HELLO_SQL	0 row(s) affected	0.531 sec	

Refresh database to see the result



- Can we just run the drop command again?

Output					
Action Output					
	#	Time	Action	Message	Duration / Fetch
✓	1	15:50:31	DROP DATABASE HELLO_SQL	0 row(s) affected	0.531 sec
✗	2	15:50:57	DROP DATABASE HELLO_SQL	Error Code: 1008. Can't drop database 'HELLO_SQL'; database doesn't exist	0.000 sec

Before delete

- Click on another database
- Close all the connection to the target delete database

DROP then CREATE

- Drop if exist

```
DROP DATABASE IF EXISTS HELLO_500AE -- xóa db nếu tồn tại
```

- Then create

```
CREATE DATABASE HELLO_500AE -- tạo ra db mới;
```



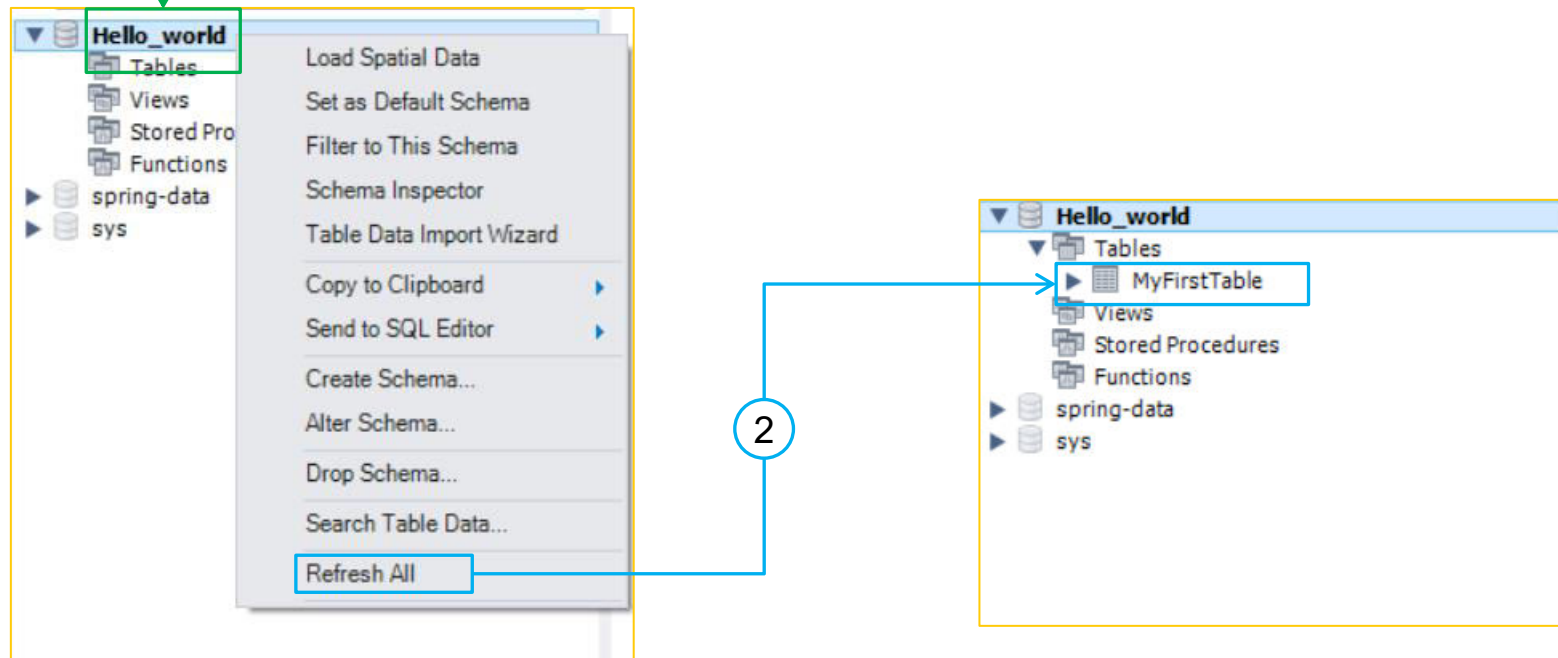
Tables

Create table

```
CREATE TABLE table_name (  
    column1 datatype,  
    column2 datatype,  
    column3 datatype,  
    ....  
);
```

```
CREATE TABLE MyFirstTable(  
    ID int,  
    FullName char(50),  
    Email varchar(20),  
    PhoneNumber varchar(10),  
    DateOfBirth date,  
    Wallet decimal(10,5)  
);
```

Action Output				
#	Time	Action	Message	Duration / Fetch
✓ 1	10:37:41	CREATE TABLE MyFirstTable(ID int, FullName c...	0 row(s) affected	0.063 sec



Modify column in table

```
-- add column in exists table
```

```
ALTER TABLE table_name  
ADD column_name datatype;
```

```
-- drop column in exists table
```

```
ALTER TABLE table_name  
DROP COLUMN column_name;
```

```
-- modified column in exists table
```

```
ALTER TABLE table_name  
MODIFY COLUMN column_name datatype;
```

View the design of table

SCHEMAS

Filter objects

- Hello_world
 - Tables
 - MyFirstTable**
 - Views
 - Stored Procedures
 - Functions
- spring-data
- sys

Table Name: MyFirstTable **Schema:** Hello_world

Column Name	Datatype	PK	NN	UQ	B	UN	ZF	AI	G	Default/Expression
ID	INT	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	NULL
FullName	CHAR(50)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	NULL
Email	VARCHAR(20)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	NULL
PhoneNumber	VARCHAR(10)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	NULL
DateOfBirth	DATE	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	NULL
Wallet	DECIMAL(10,5)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	NULL
gender	VARCHAR(10)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	NULL

Column Name: **Data Type:**

Charset/Collation:

Comments:

Default:

Storage: ☐ Virtual ☐ Stored

☐ Primary Key ☐ Not Null ☐ Unique

☐ Binary ☐ Unsigned ☐ Zero Fill

☐ Auto Increment ☐ Generated

Columns | Indexes | Foreign Keys | Triggers | Partitioning | Options

Apply **Revert**

Practice

- Add column **City** with varchar(100) into table MyFirstTable
- Modify column **City** datatype to varchar(500)
- Drop column City

Meaningful data

- SQL constraints are used to specify rules for the data in a table.
- This ensures the accuracy and reliability of the data in the table.
- If there is any violation between the constraint and the data action, the action is aborted.

Constraints

PRIMARY KEY

- Uniquely identifies each row in a table

FOREIGN KEY

- Links between tables

UNIQUE

- Ensures that all values in a column are different

DEFAULT

- Default value for a column if no value is specified

NOT NULL

- Ensures that a column cannot have a NULL value

CHECK

- Ensures that the values in a column satisfies a specific condition

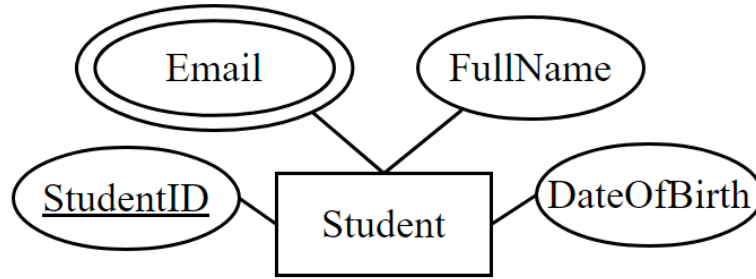
Primary key

- A **PRIMARY KEY** is a field or combination of fields which uniquely specify a row.
- Primary key values cannot be NULL.

Foreign key

- A **FOREIGN KEY** is a field (or collection of fields) in one table, that refers to the PRIMARY KEY in another table.
- The table with the **foreign key** is called the **child table**, and the table with the **primary key** is called the **referenced or parent table**

Foreign key




Student(StudentID, FullName, DateOfBirth)
↑
StudentEmail(StudentID, Email)

Example

<u>StudentID</u>	FullName	DateOfBirth
1	Snoop Dog	2/19/2000
2	The Rock	2/16/1999

<u>StudentID</u>	<u>Email</u>
1	snoop@high.com
1	snoop@low.com
2	power@man.com
2	supper@man.com

NULL	wrongdata@man.com
------	-------------------



Create table with constraints

```
CREATE TABLE MySecondTable (  
    ID INT PRIMARY KEY,  
    FullName CHAR(50) NOT NULL,  
    Email VARCHAR(20) UNIQUE,  
    PhoneNumber VARCHAR(10),  
    DateOfBirth DATE,  
    Wallet DECIMAL(10, 2) CHECK (Wallet > 0)  
);
```

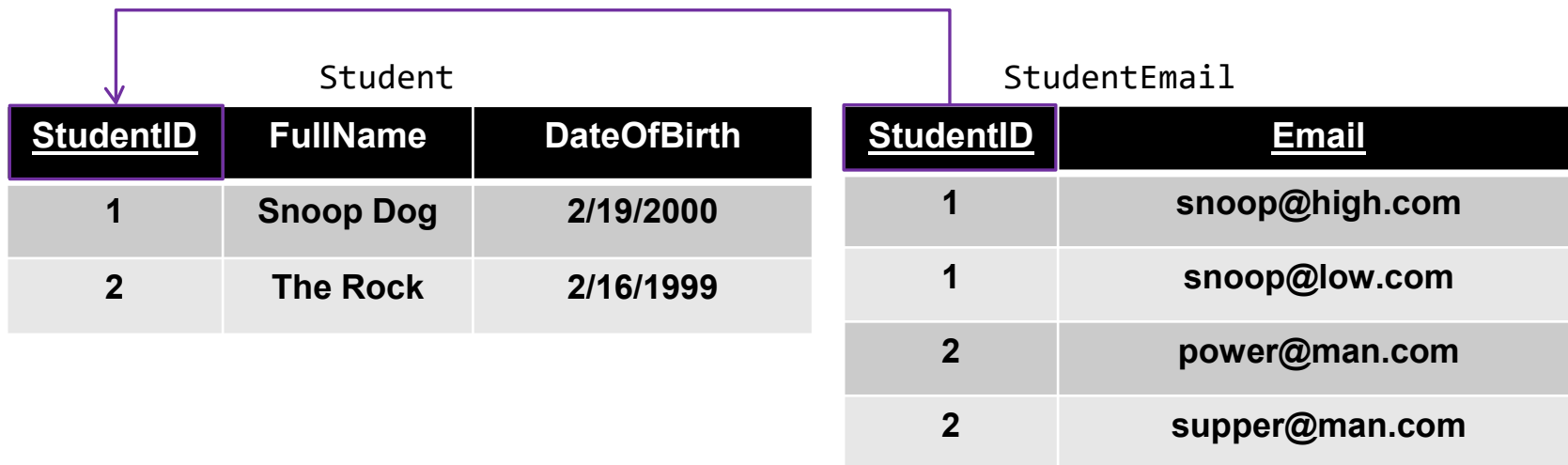
Add constraints to table

```
CREATE TABLE MySecondTableWithAlter (  
    ID INT,  
    FullName CHAR(50),  
    Email VARCHAR(20),  
    PhoneNumber VARCHAR(10),  
    DateOfBirth DATE,  
    Wallet DECIMAL(10, 2)  
);
```

```
-- Modify the ID column to be NOT NULL and add it as the primary key  
ALTER TABLE MySecondTableWithAlter  
MODIFY COLUMN ID INT NOT NULL,  
ADD PRIMARY KEY (ID);  
  
-- Modify the FullName column to be NOT NULL and increase its length  
to 60  
ALTER TABLE MySecondTableWithAlter  
MODIFY COLUMN FullName CHAR(60) NOT NULL;  
  
-- Add a UNIQUE constraint on the Email column  
ALTER TABLE MySecondTableWithAlter  
ADD UNIQUE (Email);  
  
-- Add a CHECK constraint to ensure Wallet is greater than 0  
ALTER TABLE MySecondTableWithAlter  
ADD CHECK (Wallet > 0);
```

Add foreign key

```
ALTER TABLE StudentEmail  
ADD FOREIGN KEY (StudentID) REFERENCES Student(StudentID);
```



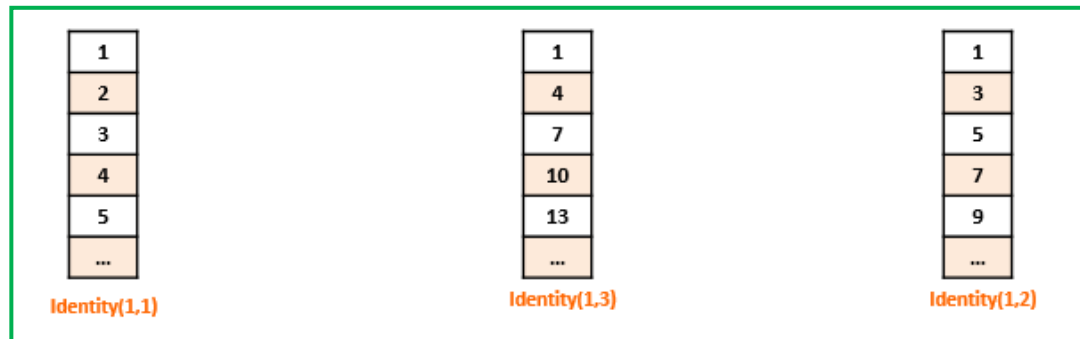
Drop table

DROP TABLE `table_name;`

ex: DROP TABLE `MySecondTableWithAlter;`

Auto generate Identity

```
CREATE TABLE StudentWithAutoIncreaseID (  
    StudentID INT AUTO_INCREMENT PRIMARY KEY,  
    FullName VARCHAR(50) NOT NULL,  
    DateOfBirth DATE DEFAULT CURRENT_TIMESTAMP  
);
```



Practice

- StudentID start from 1 and increase one by one.
- FullName is not nullable
- DateOfBirth default is '10/22/2000'
- Score between 0 and 100
- Email is unique
- in 'StudentEmail' table Primary key is StudentID & Email

Student(StudentID, FullName, DateOfBirth, Score)

StudentEmail(StudentID, Email)



Practice - Solution

```
• CREATE TABLE Student (  
    StudentID INT AUTO_INCREMENT PRIMARY KEY,  
    FullName VARCHAR(50) NOT NULL,  
    DateOfBirth DATE DEFAULT '2000-10-22',  
    Score INT,  
    CHECK (Score between 0 and 100)  
);  
  
• CREATE TABLE StudentEmail (  
    StudentID INT,  
    Email VARCHAR(50),  
    PRIMARY KEY (StudentID, Email),  
    FOREIGN KEY (StudentID) REFERENCES Student(StudentID)  
);
```


SQL convention

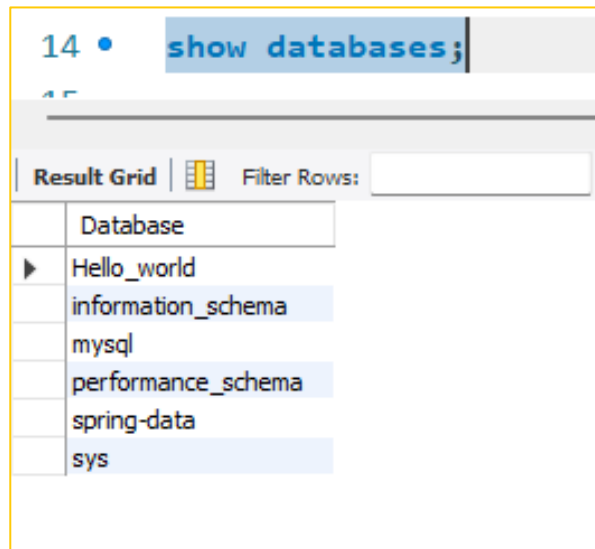
DO	TRY TO AVOID	Description
SELECT	select	SQL STATEMENT IS UPPER CASE
int	INT	Date type should be lowercase
EmployeeSalaryID	EmployeesalaryID	Follow Pascal case for variable, table, column
@studentCount	@@studentCount	Avoid @@ prefix
@studentCount	@sc	Clear meaning
EmployeeSalary	Employee Salary	Only use ([a-zA-Z][a-zA-Z0-9])

Naming Conventions

key word	prefix
PRIMARY KEY	PK_: Primary Key constraints
FOREIGN KEY	FK_: Foreign Key constraints
UNIQUE	UNI_: Unique constraints
DEFAULT	DF_: Default constraints
NOT NULL	Follow after column
CHECK	CHK_: Check constraints
View	view_: Views
Index	IX_: Indexes

See all databases

SHOW DATABASES;



Break the limit

```
CREATE TABLE BreakTheLimit(  
    Data char(8000) NOT NULL,  
    MoreData char(54) NOT NULL  
)
```



Thank you!



Any questions ?

Extra Resources

Name	Link
floating point	https://www.youtube.com/watch?v=L8OYx1l8qNg
sql datatype	https://www.w3schools.com/sql/sql_datatypes.asp
alter table	https://www.w3schools.com/sql/sql_alter.asp
SQL process	https://docs.oracle.com/database/121/TGSQL/tgsql_sqlproc.htm#TGSQL176
break limit	https://www.linkedin.com/pulse/page-size-sql-server-mohammad-mehrabani/
became SQL god?	https://www.w3schools.com/sql/default.asp