MySQL Database Learning Guide

IMPORTANT:

1. MySQL commands are not required to be on the same line
2. The command are not case-sensitive

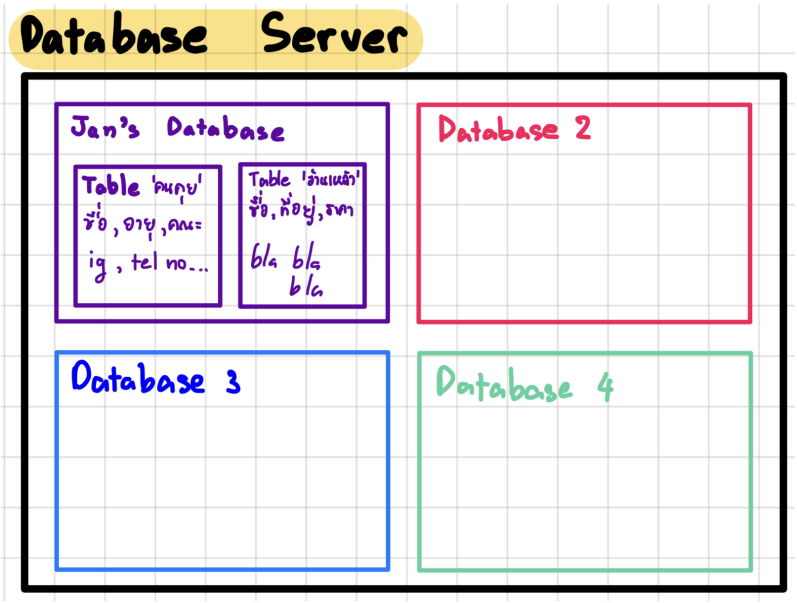
Day 1 - Introduction to SQL

SQL is a programming language for managing database. It is being stored in tabular form (table) with rows and column. (Looks like Excel)

```
+--------+----------------+------------+------+
| cat_id | name           | breed      | age  |
+--------+----------------+------------+------+
|      1 | Ringo          | Tabby      |    4 |
|      2 | Cindy          | Maine Coon |   10 |
|      3 | Dumbledore     | Maine Coon |   11 |
|      4 | Egg            | Persian    |    4 |
|      5 | Misty          | Tabby      |   13 |
|      6 | George Michael | Ragdoll    |    9 |
|      7 | Jackson        | Sphynx     |    7 |
+--------+----------------+------------+------+
```

Simple SQL Tables (From Terminal)

Database Server Structure:



In a single database server, there might be more than 1 or even hundreds of database stored.

As you can see, it structured like a silo (big -> small).

In a database, there might be a lot of table stored inside.

Every command in MySQL always end with [;] syntax, see example below.

Database Exercise: Type in these commands into terminal and paste the screenshots under the shown command

`show databases;` - to show list of databases

`CREATE DATABASE <database_name>;` - create a new database
****replace <database_name> with an actual name****

`DROP DATABASE <database_name>;` - to delete a database
****replace <database_name> with an actual name****

`USE <database_name>;` - to select that database for use
****replace <database_name> with an actual name****

Data Types

Just like in Python, this define data types that is going to be stored in the table. However, SQL has a lot more data type compared to Python.

String Data Types

Data type	Description
CHAR(size)	A FIXED length string (can contain letters, numbers, and special characters). The *size* parameter 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). The *size* parameter specifies the maximum string length in characters - can be from 0 to 65535
BINARY(size)	Equal to CHAR(), but stores binary byte strings. The *size* parameter specifies the column length in bytes. Default is 1
VARBINARY(size)	Equal to VARCHAR(), but stores binary byte strings. The *size* parameter specifies the maximum column length in bytes.
TINYBLOB	For BLOBs (Binary Large Objects). Max length: 255 bytes
TINYTEXT	Holds a string with a maximum length of 255 characters
TEXT(size)	Holds a string with a maximum length of 65,535 bytes
BLOB(size)	For BLOBs (Binary Large Objects). Holds up to 65,535 bytes of data
MEDIUMTEXT	Holds a string with a maximum length of 16,777,215 characters
MEDIUMBLOB	For BLOBs (Binary Large Objects). Holds up to 16,777,215 bytes of data
LONGTEXT	Holds a string with a maximum length of 4,294,967,295 characters
LONGBLOB	For BLOBs (Binary Large Objects). Holds up to 4,294,967,295 bytes of data
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
SET(val1, val2, val3, ...)	A string object that can have 0 or more values, chosen from a list of possible values. You can list up to 64 values in a SET list

Date and Time Data Types

Data type	Description
DATE	A date. Format: YYYY-MM-DD. The supported range is from '1000-01-01' to '9999-12-31'
DATETIME(*fsp*)	A date and time combination. Format: YYYY-MM-DD hh:mm:ss. The supported range is from '1000-01-01 00:00:00' to '9999-12-31 23:59:59'. Adding DEFAULT and ON UPDATE in the column definition to get automatic initialization and updating to the current date and time
TIMESTAMP(*fsp*)	A timestamp. TIMESTAMP 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. The supported range is from '1970-01-01 00:00:01' UTC to '2038-01-09 03:14:07' UTC. Automatic initialization and updating to the current date and time can be specified using DEFAULT CURRENT_TIMESTAMP and ON UPDATE CURRENT_TIMESTAMP in the column definition
TIME(*fsp*)	A time. Format: hh:mm:ss. The supported range is from '-838:59:59' to '838:59:59'
YEAR	A year in four-digit format. Values allowed in four-digit format: 1901 to 2155, and 0000. MySQL 8.0 does not support year in two-digit format.

Numeric Data Types

Data type	Description
BIT(*size*)	A bit-value type. The number of bits per value is specified in *size*. The *size* parameter can hold a value from 1 to 64. The default value for *size* is 1.
TINYINT(*size*)	A very small integer. Signed range is from -128 to 127. Unsigned range is from 0 to 255. The *size* parameter specifies the maximum display width (which is 255)
BOOL	Zero is considered as false, nonzero values are considered as true.
BOOLEAN	Equal to BOOL
SMALLINT(*size*)	A small integer. Signed range is from -32768 to 32767. Unsigned range is from 0 to 65535. The *size* parameter specifies the maximum display width (which is 255)
MEDIUMINT(*size*)	A medium integer. Signed range is from -8388608 to 8388607. Unsigned range is from 0 to 16777215. The *size* parameter specifies the maximum display width (which is 255)
INT(*size*)	A medium integer. Signed range is from -2147483648 to 2147483647. Unsigned range is from 0 to 4294967295. The *size* parameter specifies the maximum display width (which is 255)
INTEGER(*size*)	Equal to INT(size)
BIGINT(*size*)	A large integer. Signed range is from -9223372036854775808 to 9223372036854775807. Unsigned range is from 0 to 18446744073709551615. The *size* parameter specifies the maximum display width (which is 255)
FLOAT(*size*, *d*)	A floating point number. The total number of digits is specified in *size*. The number of digits after the decimal point is specified in the *d* parameter. This syntax is deprecated in MySQL 8.0.17, and it will be removed in future MySQL versions
FLOAT(*p*)	A floating point number. MySQL uses the *p* value to determine whether to use FLOAT or DOUBLE for the resulting data type. If *p* is from 0 to 24, the data type becomes FLOAT(). If *p* is from 25 to 53, the data type becomes DOUBLE()
DOUBLE(*size*, *d*)	A normal-size floating point number. The total number of digits is specified in *size*. The number of digits after the decimal point is specified in the *d* parameter
DOUBLE PRECISION(*size*, *d*)	
DECIMAL(*size*, *d*)	An exact fixed-point number. The total number of digits is specified in *size*. The number of digits after the decimal point is specified in the *d* parameter. The maximum number for *size* is 65. The maximum number for *d* is 30. The default value for *size* is 10. The default value for *d* is 0.
DEC(*size*, *d*)	Equal to DECIMAL(size,d)

อย่าเพิ่งท้อ!!!!

For us, the most used datatypes are
`INT` - for integer
`VARCHAR(character_length)` - for string (replace character_length by a number)
****use a single quote [‘] for defining string in MySQL****
but it maybe more in the future.

Data Types Exercise - Fill this table with data types (On the first row) and its corresponding data (next row)

Dont ask what is row and column
because I dont know either
กูก็ไม่รู้เหมือนกัน

This is a table where your tweets are being stored here

Username (Max 16 characters)	Tweet Content (Max 140 characters)	Likes	Retweet

Tables

Table is where the data is being stored. There are a lot of command for working with tables, we are going to cover most of them.

The concept is called ‘CRUD’
C - Create
R - Read
U - Update
D - Delete

C - Create

Table creation form looks like this:

```
CREATE TABLE <name> (
      <col1_name> <datatypes> <addi_arg>,
      <col2_name> <datatypes> <addi_arg>,
);
```

For example, creating table storing dogs information about name, age, and breed:

```
CREATE TABLE dogs (
      name VARCHAR(25),
      age INT,
      breed VARCHAR(50),
);
```

Or it can be shortened to a single line as MySQL does not care about lining

```
CREATE TABLE dogs (name VARCHAR(25), age INT, breed VARCHAR(50));
```

When created, we use the command `DESC <table_name>` to show the table properties

```
mysql> DESC dogs
    -> ;
+-------+-------------+------+-----+---------+-------+
| Field | Type        | Null | Key | Default | Extra |
+-------+-------------+------+-----+---------+-------+
| name  | varchar(25) | YES  |     | NULL    |       |
| age   | int         | YES  |     | NULL    |       |
| breed | varchar(50) | YES  |     | NULL    |       |
+-------+-------------+------+-----+---------+-------+
3 rows in set (0.02 sec)

mysql> desc dogs;
+-------+-------------+------+-----+---------+-------+
| Field | Type        | Null | Key | Default | Extra |
+-------+-------------+------+-----+---------+-------+
| name  | varchar(25) | YES  |     | NULL    |       |
| age   | int         | YES  |     | NULL    |       |
| breed | varchar(50) | YES  |     | NULL    |       |
+-------+-------------+------+-----+---------+-------+
3 rows in set (0.00 sec)

mysql> DeSc doGS;
+-------+-------------+------+-----+---------+-------+
| Field | Type        | Null | Key | Default | Extra |
+-------+-------------+------+-----+---------+-------+
| name  | varchar(25) | YES  |     | NULL    |       |
| age   | int         | YES  |     | NULL    |       |
| breed | varchar(50) | YES  |     | NULL    |       |
+-------+-------------+------+-----+---------+-------+
3 rows in set (0.00 sec)
```

Notice anything strange???

Additional Arguments

```
+------+-----+---------+-------+
| Null | Key | Default | Extra |
+------+-----+---------+-------+
| YES  |     | NULL    |       |
| YES  |     | NULL    |       |
| YES  |     | NULL    |       |
+------+-----+---------+-------+
```

These are <addi_arg> or additional arguments in the command.
Null - Allowing the field to be empty
Key - Special label that help us identify the row in a table Explore More about keys
Default - default value
Extra - Additional condition

Add these command to the <addi_arg>

List of additional arguments

NOT NULL	Preventing the inserted data to be empty
DEFAULT <value>	Adding default value. It could be string or integer DEFAULT 'BCC' or DEFAULT 168
PRIMARY KEY	Set the column to be primary key. Can be used 2 ways, adding to <addi_arg> or on another line: PRIMARY KEY <row_num>
AUTO_INCREMENT	Used for automatically increase the value. ONLY APPLICABLE TO A KEY COLUMN.

Example: we are going to create a table storing student data. The table should have student id****, first name*, last name*, age*, status**, additional information***.

* is for required field, must be fill no matter what
** is to have a default value
*** is allow to be empty
**** is going to be automatically assign without receiving input

```
CREATE TABLE StudentData (
      student_id INT AUTO_INCREMENT PRIMARY KEY,
      first_name VARCHAR(100) NOT NULL,
      last_name VARCHAR(100) NOT NULL,
      status VARCHAR(10) DEFAULT 'Student',
      Additional_info VARCHAR(100)
);
```

```
mysql> DESC StudentData;
+-----------------+--------------+------+-----+---------+----------------+
| Field           | Type         | Null | Key | Default | Extra          |
+-----------------+--------------+------+-----+---------+----------------+
| student_id      | int          | NO   | PRI | NULL    | auto_increment |
| first_name      | varchar(100) | NO   |     | NULL    |                |
| last_name       | varchar(100) | NO   |     | NULL    |                |
| status          | varchar(10)  | YES  |     | Student |                |
| Additional_info | varchar(100) | YES  |     | NULL    |                |
+-----------------+--------------+------+-----+---------+----------------+
5 rows in set (0.00 sec)
```

Exercise: Create a table that stores list of คนคุย
Requirement:

1. ลำดับที่ (ตั้งให้มากขึ้นเรื่อยๆแบบอัตโนมัติ)
2. ชื่อเล่น (ห้ามเว้น)
3. คณะ (ห้ามเว้น)
4. ปี (ห้ามเว้น)
5. สถานะ (ให้กำหนดว่าเลิกคุยแล้วโดยอัตโนมัติ)
6. Ig ห้ามเว้น
7. Line id เว้นได้

เสร็จแล้วแคปจอ desc มาแปะข้างล่างนี้

Inserting Data

Now we have created an empty table ready to store the data. Insert data into the table by using command:

```
INSERT INTO <table_name> (<col1>,<col2>)
VALUES (<col1_data>,<col2_data>);
```

Multi-Inserts

```
INSERT INTO <table_name> (<col1>,<col2>)
VALUES
(<col1_data1>,<col2_data1>)
(<col1_data2>,<col2_data2>)
(<col1_data3>,<col2_data3>);
```

Example: Adding 2 students data to the table

First Student:

- First Name: Capy
- Last Name: Bara
- Status: Alumni (จบแล้ว)
- Additional Info: -

Second Student:

- First Name: Bombastic
- Last Name: Sideeye
- Status: -
- Additional Info: Criminal Offensive

```
INSERT INTO StudentData(first_name, last_name, status,additional_info)
VALUE
('Capy','Bara','Alumni',NULL),
('Bombastic','Sideeye',DEFAULT,'CriminalOffensive');
```

The result:

```
mysql> SELECT * FROM StudentData;
+------------+------------+-----------+---------+--------------------+
| student_id | first_name | last_name | status  | Additional_info    |
+------------+------------+-----------+---------+--------------------+
|          1 | Capy       | Bara      | Alumni  | NULL               |
|          2 | Bombastic  | Sideeye   | Student | Criminal Offensive |
+------------+------------+-----------+---------+--------------------+
2 rows in set (0.00 sec)
```

Notice that to show data that we input to the table we use the command

```
SELECT * FROM <table_name>;
```

Exercise: Input the data to your newly created table called คนคุย and print out the data

R - Read

From previous section, we already learn that to print out data from the table we will be using

```
SELECT <col> FROM <table_name>;
```

There are more arguments that we can add to the command for outputting certain data.

Given the table:

```
mysql> SELECT * FROM cats;
+--------+---------------+------------+------+
| cat_id | name          | breed      | age  |
+--------+---------------+------------+------+
|      1 | Ringo         | Tabby      |    4 |
|      2 | Cindy         | Maine Coon |   10 |
|      3 | Dumbledore    | Maine Coon |   11 |
|      4 | Egg           | Persian    |    4 |
|      5 | Misty         | Tabby      |   13 |
|      6 | George Michael | Ragdoll    |    9 |
|      7 | Jackson       | Sphynx     |    7 |
+--------+---------------+------------+------+
```

To print out only certain column from the table we modify <col> to be that column name

```
mysql> SELECT name FROM cats;
+----------------+
| name           |
+----------------+
| Ringo          |
| Cindy          |
| Dumbledore     |
| Egg            |
| Misty          |
| George Michael |
| Jackson        |
+----------------+
```

```
mysql> SELECT breed FROM cats;
+------------+
| breed      |
+------------+
| Tabby      |
| Maine Coon |
| Maine Coon |
| Persian    |
| Tabby      |
| Ragdoll    |
| Sphynx     |
+------------+
```

```
mysql> SELECT name, breed FROM cats;
+----------------+------------+
| name           | breed      |
+----------------+------------+
| Ringo          | Tabby      |
| Cindy          | Maine Coon |
| Dumbledore     | Maine Coon |
| Egg            | Persian    |
| Misty          | Tabby      |
| George Michael | Ragdoll    |
| Jackson        | Sphynx     |
+----------------+------------+
```

The command does not limit how many column are being print out at a time.

We can add another column by adding comma **[,]** and another column name in the back

```
SELECT <col1>, <col2> FROM <table_name>;
```

WHERE argument

Works similarly to if-statement in python. Imagine we want to print out some data that match a certain aspect.

```
SELECT <col1> FROM <table_name> WHERE <condition>;
```

Example 1: Only print out name of tabby cat

```
mysql> SELECT name FROM cats WHERE breed='Tabby';
+-------+
| name  |
+-------+
| Ringo |
| Misty |
+-------+
```

Example 2: Only print out cat whose age is more than or equal to 10

```
mysql> SELECT name, age FROM cats WHERE age>=10;
+------------+------+
| name       | age  |
+------------+------+
| Cindy      |   10 |
| Dumbledore |   11 |
| Misty      |   13 |
+------------+------+
```

Example 3: Only print out cat that their age match their id

```
mysql> SELECT * FROM cats WHERE cat_id=age;
+--------+---------+---------+------+
| cat_id | name    | breed   | age  |
+--------+---------+---------+------+
|      4 | Egg     | Persian |    4 |
|      7 | Jackson | Sphynx  |    7 |
+--------+---------+---------+------+
```

Exercise:

1. Print ชื่อของคนคุยที่อยู่ปี 3
2. Print ig คนคุยที่อยู่ปี 4
3. Print ชื่อเล่นกับปีของคนที่อยู่ปี 2 ขึ้นไป
4. ปริ้น ig ของคนที่แก่กว่า

ALIASES or AS

Sometimes column name are too long or inconvenient to understand, we use command

```
SELECT <col> AS <preferred_name> FROM <table_name>;
SELECT <col1> AS <preferred_name>, <col2> FROM <table_name>;
```

Example: cat_id in my table is quite long, I want to temporary change it to id for the sake of reading the table

```
mysql> SELECT cat_id, name FROM cats;
+--------+----------------+
| cat_id | name           |
+--------+----------------+
|      1 | Ringo          |
|      2 | Cindy          |
|      3 | Dumbledore     |
|      4 | Egg            |
|      5 | Misty          |
|      6 | George Michael |
|      7 | Jackson        |
+--------+----------------+
```

```
mysql> SELECT cat_id AS id, name FROM cats;
+----+----------------+
| id | name           |
+----+----------------+
|  1 | Ringo          |
|  2 | Cindy          |
|  3 | Dumbledore     |
|  4 | Egg            |
|  5 | Misty          |
|  6 | George Michael |
|  7 | Jackson        |
+----+----------------+
```

U - Update

What if we do want to change the data value in the table, we this command to update the data

```
UPDATE <table_name> SET <col1=value> <additional_argument>;
UPDATE <table_name> SET <col1=value> WHERE <col2=value>;
```

Value can be string or integer.

Example: Changing Ringo breed to Siamese and age to 5

```
mysql> SELECT * FROM cats;
+--------+----------------+------------+------+
| cat_id | name           | breed      | age  |
+--------+----------------+------------+------+
|      1 | Ringo          | Tabby      |    4 |
|      2 | Cindy          | Maine Coon |   10 |
|      3 | Dumbledore     | Maine Coon |   11 |
|      4 | Egg            | Persian    |    4 |
|      5 | Misty          | Tabby      |   13 |
|      6 | George Michael | Ragdoll    |    9 |
|      7 | Jackson        | Sphynx     |    7 |
+--------+----------------+------------+------+
```

```
mysql> UPDATE cats SET breed='Siamese' WHERE name='Ringo';
Query OK, 1 row affected (0.01 sec)
Rows matched: 1  Changed: 1  Warnings: 0

mysql> SELECT * FROM cats;
+--------+----------------+------------+------+
| cat_id | name           | breed      | age  |
+--------+----------------+------------+------+
|      1 | Ringo          | Siamese    |    4 |
|      2 | Cindy          | Maine Coon |   10 |
```

```
mysql> UPDATE cats SET age=5 WHERE name='Ringo';
Query OK, 1 row affected (0.00 sec)
Rows matched: 1  Changed: 1  Warnings: 0

mysql> SELECT * FROM cats;
+--------+----------------+------------+------+
| cat_id | name           | breed      | age  |
+--------+----------------+------------+------+
|      1 | Ringo          | Siamese    |    5 |
|      2 | Cindy          | Maine Coon |   10 |
```

Exercise: เปลี่ยนสถานะคนคุยให้กลับมาคุยกันคนนึง

D - Delete

When we want to delete the data in the table we use this command

```
DELETE FROM <table_name>;
DELETE FROM <table_name> WHERE <col=value>;
```

The first command is for removing everything in the table
The 2nd command is for removing specific row with specific condition

Example 1: Delete all row from a table

```
mysql> SHOW TABLES;
+--------------------+
| Tables_in_petshop  |
+--------------------+
| cats               |
| dogs               |
| Employees          |
| shops              |
| StudentData        |
+--------------------+
```

```
mysql> SELECT * FROM shops;
+-----------------+
| name            |
+-----------------+
| shoe emporium   |
| mario's pizza   |
| mario's pizza2  |
+-----------------+
```

Now we are going to empty this table data

```
mysql> DELETE FROM shops;
Query OK, 3 rows affected (0.01 sec)

mysql> SELECT * FROM shops;
Empty set (0.00 sec)
```

Using the 1st command, shops table is now empty

Example 2: Delete a certain row from a certain value (Using cats table)

```
mysql> DELETE FROM cats WHERE cat_id=6;
Query OK, 1 row affected (0.01 sec)

mysql> SELECT * FROM cats;
+--------+------------+------------+------+
| cat_id | name       | breed      | age  |
+--------+------------+------------+------+
|      1 | Ringo      | Siamese    |    5 |
|      2 | Cindy      | Maine Coon |   10 |
|      3 | Dumbledore | Maine Coon |   11 |
|      4 | Egg        | Persian    |    4 |
|      5 | Misty      | Tabby      |   13 |
|      7 | Jackson    | Sphynx     |    7 |
+--------+------------+------------+------+
```

Now the cat with cat_id = 6 is not present in the table anymore.

Exercise: ลบคนคุยที่ไม่ชอบที่สุดออก

Exercise 1: CRUD

This exercise aims for making you familiar with CRUD command

Screen Capture each question result and paste it under the space provided

1. Create a new database named **shirts_db**

2. Create new table **shirts**

3. Insert these data into the table (Primary Key is shirt_id)

shirt_id	article	color	shirt_size	last_worn
1	t-shirt	white	S	10
2	t-shirt	green	S	200
3	polo shirt	black	M	10
4	tank top	blue	S	50
5	t-shirt	pink	S	0
6	polo shirt	red	M	5
7	tank top	white	S	200
8	tank top	blue	M	15

4. Add another shirt with this description:
   Purple polo shirt with size M last worn 50 days ago

5. Print out article and color of all shirts

6. Print out everything of all shirt with medium size except shirt id

7. Change all polo shirt size to L

8. Update the last worn shirt 15 days ago to 0 day

9. Update all white shirt size to XS and color to off white

10. Delete old shirts thats over 200 days last worn

11. Delete all tank tops

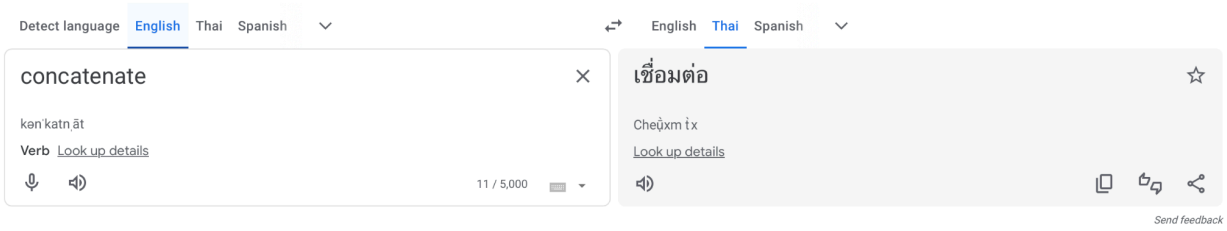
12. Delete all shirt

13. Drop the shirt table

Day 2 - Advanced String Functions

You may know about using SELECT function from previous class, but this time we will dive deeper to using advanced SELECT function for extra special result.

CONCAT, CONCAT_WS - Concatenating String



การเชื่อมต่อ string เข้าด้วยกัน

```
mysql> SELECT CONCAT('H','E','L','L','O');
+-----------------------------+
| CONCAT('H','E','L','L','O') |
+-----------------------------+
| HELLO                       |
+-----------------------------+
```

```
mysql> SELECT CONCAT_WS(' - ','H','E','L','L','O');
+--------------------------------------+
| CONCAT_WS(' - ','H','E','L','L','O') |
+--------------------------------------+
| H - E - L - L - O                    |
+--------------------------------------+
```

```
SELECT CONCAT( <col1> , <col2> );
SELECT CONCAT_WS( <connecting_string>, <col1>, <col2>);
```

CONCAT จะเป็นการเชื่อมต่อ string เข้ามาด้วยกันเฉยๆ
CONCAT_WS จะเป็นการเชื่อมต่อ string เข้ามาด้วยกันและเพิ่มตัวเชื่อมระหว่าง string เข้าไปด้วย

Example: CONCAT

```
mysql> select * from books;
+---------+-----------------------------------------------------+--------------+-----------------+---------------+----------------+-------+
| book_id | title                                               | author_fname | author_lname    | released_year | stock_quantity | pages |
+---------+-----------------------------------------------------+--------------+-----------------+---------------+----------------+-------+
|       1 | The Namesake                                        | Jhumpa       | Lahiri          |          2003 |             32 |   291 |
|       2 | Norse Mythology                                     | Neil         | Gaiman          |          2016 |             43 |   304 |
|       3 | American Gods                                       | Neil         | Gaiman          |          2001 |             12 |   465 |
|       4 | Interpreter of Maladies                             | Jhumpa       | Lahiri          |          1996 |             97 |   198 |
|       5 | A Hologram for the King: A Novel                    | Dave         | Eggers          |          2012 |            154 |   352 |
|       6 | The Circle                                          | Dave         | Eggers          |          2013 |             26 |   504 |
|       7 | The Amazing Adventures of Kavalier & Clay           | Michael      | Chabon          |          2000 |             68 |   634 |
|       8 | Just Kids                                           | Patti        | Smith           |          2010 |             55 |   304 |
|       9 | A Heartbreaking Work of Staggering Genius           | Dave         | Eggers          |          2001 |            104 |   437 |
|      10 | Coraline                                            | Neil         | Gaiman          |          2003 |            100 |   208 |
|      11 | What We Talk About When We Talk About Love: Stories | Raymond      | Carver          |          1981 |             23 |   176 |
|      12 | Where I'm Calling From: Selected Stories            | Raymond      | Carver          |          1989 |             12 |   526 |
|      13 | White Noise                                         | Don          | DeLillo         |          1985 |             49 |   320 |
|      14 | Cannery Row                                         | John         | Steinbeck       |          1945 |             95 |   181 |
|      15 | Oblivion: Stories                                   | David        | Foster Wallace  |          2004 |            172 |   329 |
|      16 | Consider the Lobster                                | David        | Foster Wallace  |          2005 |             92 |   343 |
+---------+-----------------------------------------------------+--------------+-----------------+---------------+----------------+-------+
```

จาก Table นี้ ต้องการที่จะเชื่อมชื่อและนามสกุลคนแต่ง

```
SELECT CONCAT(author_fname, author_lname) FROM BOOKS;
```

```
mysql> SELECT CONCAT(author_fname, author_lname) FROM books;
+-------------------------------------+
| CONCAT(author_fname, author_lname) |
+-------------------------------------+
| JhumpaLahiri                        |
| NeilGaiman                          |
| NeilGaiman                          |
| JhumpaLahiri                        |
| DaveEggers                          |
| DaveEggers                          |
| MichaelChabon                       |
| PattiSmith                          |
| DaveEggers                          |
| NeilGaiman                          |
| RaymondCarver                       |
| RaymondCarver                       |
| DonDeLillo                          |
| JohnSteinbeck                       |
| DavidFoster Wallace                 |
| DavidFoster Wallace                 |
+-------------------------------------+
```

จะเห็นว่าชื่อและนามสกุลติดกัน

และ title หัวตารางมันไม่สวย

```
SELECT CONCAT(author_fname,' ', author_lname) AS Author_Name FROM BOOKS;
```

```
mysql> SELECT CONCAT(author_fname,' ', author_lname) AS Author_Name FROM BOOKS;
+----------------------+
| Author_Name          |
+----------------------+
| Jhumpa Lahiri        |
| Neil Gaiman          |
| Neil Gaiman          |
| Jhumpa Lahiri        |
| Dave Eggers          |
| Dave Eggers          |
| Michael Chabon       |
| Patti Smith          |
| Dave Eggers          |
| Neil Gaiman          |
| Raymond Carver       |
| Raymond Carver       |
| Don DeLillo          |
| John Steinbeck       |
| David Foster Wallace |
| David Foster Wallace |
+----------------------+
```

เปลี่ยนชื่อหัวตารางเป็น Author_Name

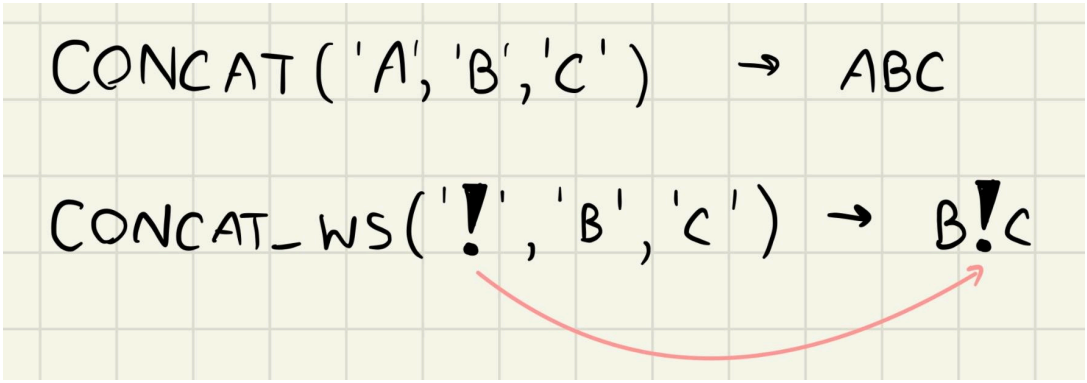
เพิ่ม Space ระหว่างชื่อและนามสกุล

Example: CONCAT_WS

```
SELECT CONCAT_WS(' ',author_fname, author_lname) AS Author_name_ws FROM
books;
```

```
mysql> SELECT CONCAT_WS(' ',author_fname, author_lname) AS Author_name_ws FROM books;
+----------------------+
| Author_name_ws       |
+----------------------+
| Jhumpa Lahiri        |
| Neil Gaiman          |
| Neil Gaiman          |
| Jhumpa Lahiri        |
| Dave Eggers          |
| Dave Eggers          |
| Michael Chabon       |
| Patti Smith          |
| Dave Eggers          |
| Neil Gaiman          |
| Raymond Carver       |
| Raymond Carver       |
| Don DeLillo          |
| John Steinbeck       |
| David Foster Wallace |
| David Foster Wallace |
+----------------------+
```

ฟังชั่นนี้ก็สามารถทำงานได้เหมือนกัน



Exercise: ต่อชื่อคนคุย, คณะ, สถานะเข้าด้วยกัน ให้มีขีดตรงกลาง (e.g Bird - EBA - Not Talking)

SUBSTRING - String Slicing

Slicing string just like in Python, but it is different

```
SELECT SUBSTRING(<string>, <starting_point>, <slicing_length>) FROM
<table_name>;
SELECT SUBSTR(<string>, <starting_point>, <slicing_length>) FROM
<table_name>;
```

SUBSTR is a synonym for SUBSTRING, both functions the same.
**NOTE : Unlike Python, SQL count the first character as 1

```
mysql> SELECT SUBSTRING('Hello World',1);
+----------------------------+
| SUBSTRING('Hello World',1) |
+----------------------------+
| Hello World                |
+----------------------------+
```

```
mysql> SELECT SUBSTRING('Hello World',4);
+----------------------------+
| SUBSTRING('Hello World',4) |
+----------------------------+
| lo World                   |
+----------------------------+
```

Exercise: Write down result for these command (ห้ามใช้คอม คิดเองนะจ๊ะ)

```
SELECT SUBSTR('Now you''re in New York', 5);
```

Answer:

```
SELECT SUBSTRING('Let It Gooooooooooooo',4);
```

Answer:

```
SELECT SUBSTR('ABCDEFGHIJKLMNOPQRSTUVWXYZ', 8,5);
```

Answer:

```
SELECT SUBSTRING('0 1 2 3 4 5 6 7 8 9',2,3);
```

Answer:

```
SELECT SUBSTR('Do you believe love at first sight? Or, do I have to walk by
again?',4,99);
```

Answer:

REPLACE - Replace part of the string

IMPORTANT - The string is case sensitive

```
SELECT REPLACE(<string>, <from>, <to>);
SELECT REPLACE(<string>, <from>, <to>) FROM <table>;
```

Example 1: All repeated string are going to be replaced

```
mysql> SELECT REPLACE('Jiggle Jiggle', 'Jiggle', 'Hello') as Replace_Example;
+-----------------+
| Replace_Example |
+-----------------+
| Hello Hello     |
+-----------------+
```

Example 2: Specified string are case sensitive

```
mysql> SELECT REPLACE('Jan and Jeen', 'jeen', 'Jan') as Replace_Example_2;
+-------------------+
| Replace_Example_2 |
+-------------------+
| Jan and Jeen      |
+-------------------+
```

REVERSE - Reverse the string

It can be use with both string and integer

Example 1: Reversing Number

```
mysql> SELECT REVERSE(78);
+-------------+
| REVERSE(78) |
+-------------+
| 87          |
+-------------+
```

Example 2: Reversing String

```
mysql> SELECT REVERSE('OLLEH');
+-------------------+
| REVERSE('OLLEH') |
+-------------------+
| HELLO             |
+-------------------+
```

CHAR_LENGTH - Count character length

Works like lovely len function from python

```
SELECT CHAR_LENGTH(<string>);
SELECT CHAR_LENGTH(<string>) FROM <table>;
```

Example: SQL and Python len function

```
mysql> SELECT CHAR_LENGTH('You got that big big, bubble butt');
+-------------------------------------------------+
| CHAR_LENGTH('You got that big big, bubble butt') |
+-------------------------------------------------+
|                                              33 |
+-------------------------------------------------+
```

```
print(len("You got that big big, bubble butt"))
✓ 0.0s

33
```

UPPER & LOWER

Just like in Python

UPPER - Change all character to be upper-case

```
SELECT UPPER(<string>);
SELECT UCASE(<string>);
SELECT UPPER(<string>) FROM <table>;
SELECT UCASE(<string>) FROM <table>;
```

```
mysql> SELECT UPPER('hello');
+----------------+
| UPPER('hello') |
+----------------+
| HELLO          |
+----------------+
```

LOWER - Change all character to be lower-case

```
SELECT LOWER(<string>);
SELECT LCASE(<string>);
SELECT LOWER(<string>) FROM <table>;
SELECT LCASE(<string>) FROM <table>;
```

```
mysql> SELECT LOWER('ITS ME');
+-----------------+
| LOWER('ITS ME') |
+-----------------+
| its me          |
+-----------------+
```

INSERT - Insert string into another string

Insert given string into a position on another string

```
SELECT INSERT(<string> , <position>, <replace_length>, <new_string>);
SELECT INSERT(<string> , <position>, <replace_length>, <new_string>) FROM
<table>;
```

`<replace_length>` is the count of how many character of new string going to replace the old string

Example 1: Insert & Replace some character in a string

```
mysql> SELECT INSERT('Pink Champagne', 3, 2, 'g');
+-------------------------------------+
| INSERT('Pink Champagne', 3, 2, 'g') |
+-------------------------------------+
| Pig Champagne                       |
+-------------------------------------+
```

Example 2: Insert without replacing character

```
mysql> SELECT INSERT('Pink Champagne', 3, 0, 'g');
+-------------------------------------+
| INSERT('Pink Champagne', 3, 0, 'g') |
+-------------------------------------+
| Pignk Champagne                     |
+-------------------------------------+
```

Example 3:

```
mysql> SELECT INSERT('I gave a second chance to ', 26, 0, 'Cupid');
+-----------------------------------------------------+
| INSERT('I gave a second chance to ', 26, 0, 'Cupid') |
+-----------------------------------------------------+
| I gave a second chance toCupid                      |
+-----------------------------------------------------+
```

Example 4:

```
mysql> SELECT INSERT('I gave a second chance to ', 26, 0, ' Cupid');
+------------------------------------------------------+
| INSERT('I gave a second chance to ', 26, 0, ' Cupid') |
+------------------------------------------------------+
| I gave a second chance to Cupid                      |
+------------------------------------------------------+
```

LEFT & RIGHT - Select string from left or right side

```
SELECT LEFT(<string>, <length>);
SELECT RIGHT(<string>, <length>);
```

Example 1: Print from right side

```
mysql> SELECT RIGHT('Memories follow me left and right', 5);
+---------------------------------------------+
| RIGHT('Memories follow me left and right', 5) |
+---------------------------------------------+
| right                                       |
+---------------------------------------------+
```

Example 2: Print from left side

```
mysql> SELECT LEFT('Memories follow me left and right', 5);
+---------------------------------------------+
| LEFT('Memories follow me left and right', 5) |
+---------------------------------------------+
| Memor                                       |
+---------------------------------------------+
```

REPEAT - Repeat the string

```
SELECT REPEAT(<string>, <times>);
```

Example:

```
mysql> SELECT REPEAT('Ha', 3);
+-----------------+
| REPEAT('Ha', 3) |
+-----------------+
| HaHaHa          |
+-----------------+
```

TRIM - Trim out spaces on each end in string

```
SELECT TRIM(<string>);
```

Example:

```
mysql> SELECT TRIM('      YO         ');
+--------------------------+
| TRIM('      YO         ') |
+--------------------------+
| YO                       |
+--------------------------+
```

Exercise 2: Combining String Functions

String functions can be mix and match to achieve a more complex result. EASY NOT HARD NAJA

We have already know about CONCAT and SUBSTRING, lets try to combine these two together.

book_id	title	author_fname	author_lname	released_year	stock_quantity	pages
1	The Namesake	Jhumpa	Lahiri	2003	32	291
2	Norse Mythology	Neil	Gaiman	2016	43	304
3	American Gods	Neil	Gaiman	2001	12	465
4	Interpreter of Maladies	Jhumpa	Lahiri	1996	97	198
5	A Hologram for the King: A Novel	Dave	Eggers	2012	154	352
6	The Circle	Dave	Eggers	2013	26	504
7	The Amazing Adventures of Kavalier & Clay	Michael	Chabon	2000	68	634
8	Just Kids	Patti	Smith	2010	55	304
9	A Heartbreaking Work of Staggering Genius	Dave	Eggers	2001	104	437
10	Coraline	Neil	Gaiman	2003	100	208
11	What We Talk About When We Talk About Love: Stories	Raymond	Carver	1981	23	176
12	Where I'm Calling From: Selected Stories	Raymond	Carver	1989	12	526
13	White Noise	Don	DeLillo	1985	49	320
14	Cannery Row	John	Steinbeck	1945	95	181
15	Oblivion: Stories	David	Foster Wallace	2004	172	329
16	Consider the Lobster	David	Foster Wallace	2005	92	343

The imported table should look like this

Type your answer code under the picture

HINT: Check out the given table in great detail

Exercise 1: Print out first 10 character of book names + …

concat(substr(title,1,10),'...')
The Namesa...
Norse Myth...
American G...
Interprete...
A Hologram...
The Circle...
The Amazin...
Just Kids...
A Heartbre...
Coraline...
What We Ta...
Where I'm ...
White Nois...
Cannery Ro...
Oblivion: ...
Consider t...

Exercise 2: Print out table to look like this

```
+-----------------------------------------+
| Book_Release                            |
+-----------------------------------------+
| Title :The Namesa... | Release Year: 2003 |
| Title :Norse Myth... | Release Year: 2016 |
| Title :American G... | Release Year: 2001 |
| Title :Interprete... | Release Year: 1996 |
| Title :A Hologram... | Release Year: 2012 |
| Title :The Circle... | Release Year: 2013 |
| Title :The Amazin... | Release Year: 2000 |
| Title :Just Kids... | Release Year: 2010  |
| Title :A Heartbre... | Release Year: 2001 |
| Title :Coraline... | Release Year: 2003   |
| Title :What We Ta... | Release Year: 1981 |
| Title :Where I'm ... | Release Year: 1989 |
| Title :White Nois... | Release Year: 1985 |
| Title :Cannery Ro... | Release Year: 1945 |
| Title :Oblivion: ... | Release Year: 2004 |
| Title :Consider t... | Release Year: 2005 |
+-----------------------------------------+
```

Exercise 3: Print out table to look like this

```
+-----------------------------------------+
| Book_Stock                              |
+-----------------------------------------+
| Book ID: 1| Title: The Na| Stock Count: 32   |
| Book ID: 2| Title: Norse | Stock Count: 43   |
| Book ID: 3| Title: Americ| Stock Count: 12   |
| Book ID: 4| Title: Interp| Stock Count: 97   |
| Book ID: 5| Title: A Holo| Stock Count: 154  |
| Book ID: 6| Title: The Ci| Stock Count: 26   |
| Book ID: 7| Title: The Am| Stock Count: 68   |
| Book ID: 8| Title: Just K| Stock Count: 55   |
| Book ID: 9| Title: A Hear| Stock Count: 104  |
| Book ID: 10| Title: Corali| Stock Count: 100 |
| Book ID: 11| Title: What W| Stock Count: 23  |
| Book ID: 12| Title: Where | Stock Count: 12  |
| Book ID: 13| Title: White | Stock Count: 49  |
| Book ID: 14| Title: Canner| Stock Count: 95  |
| Book ID: 15| Title: Oblivi| Stock Count: 172 |
| Book ID: 16| Title: Consid| Stock Count: 92  |
+-----------------------------------------+
```

Exercise 4: Print out first 10 character of the book with author first and last name plus last two digit of the year released (ex. ด้วยรักและหักหลัง - J.Kotchakorn - 22)

Exercise 5: Change space in the title name to - and change all character to be uppercase

Exercise 6: Print out the first character of author lastname and add a dot and combine it with firstname (ex. J.Kotchakorn) (ALL UPPER CASE)

Day 3 (Part 1) - Refining Selection

Select function can be modified to achieve more accurate result.
Add these books into the database

```
INSERT INTO books
(title, author_fname, author_lname, released_year, stock_quantity, pages)
VALUES ('10% Happier', 'Dan', 'Harris', 2014, 29, 256),
('fake_book', 'Freida', 'Harris', 2001, 287, 428),
('Lincoln In The Bardo', 'George', 'Saunders', 2017, 1000, 367);
```

DISTINCT - Select unique value from the data

```
SELECT DISTINCT <col> FROM <table_name>;
SELECT DISTINCT <value> FROM <table_name>;
```

Using distinct function helps selecting value that has a column that duplicated but the other column is not the same

```
mysql> SELECT DISTINCT released_year FROM books;
+---------------+
| released_year |
+---------------+
|          2003 |
|          2016 |
|          2001 |
|          1996 |
|          2012 |
|          2013 |
|          2000 |
|          2010 |
|          1981 |
|          1989 |
|          1985 |
|          1945 |
|          2004 |
|          2005 |
|          2014 |
|          2017 |
+---------------+
```

Exercise 1: Show a list of author first name that is unique

Exercise 2: Show a list of author fullname (first + last) that is unique

ORDER BY - Ordering the selected value

```
SELECT <col> FROM <table_name> ORDER BY <col> DESC;
SELECT <col1>, <col2> FROM <table_name> ORDER BY <col3>, <col4>;
SELECT <col1>, <col2>, <col3> AS <name> FROM <table> ORDER BY <name>;
```

This function can be used with both alphabets or number. By default, without specifying DESC in the back, it would sort the value in Ascending order. However, adding DESC will sort the data in Desending order.

You can also use order by with multiple column as it has a duplicated value (e.g Author name) but another column has unique value (e.g Book Title, Released Year)

```
mysql> SELECT author_lname FROM books ORDER BY author_lname;
+----------------+
| author_lname   |
+----------------+
| Carver         |
| Carver         |
| Chabon         |
| DeLillo        |
| Eggers         |
| Eggers         |
| Eggers         |
| Foster Wallace |
| Foster Wallace |
| Gaiman         |
| Gaiman         |
| Gaiman         |
| Harris         |
| Harris         |
| Lahiri         |
| Lahiri         |
| Saunders       |
| Smith          |
| Steinbeck      |
+----------------+
```

Exercise 1: Sort the list of books by page count (both ascending and descending)

Exercise 2: Sort the list of books by released year (both ascending and descending)

Exercise 3: Sort the list of books by the author firstname (both ascending and descending)

LIMIT - Limit number of selected value to a specific count

```
SELECT <col> FROM <table> LIMIT <number>;
SELECT <col> FROM <table> LIMIT <start from> <data_count>;
```

Limit can be added to the back of SELECT function

```
mysql> SELECT author_lname FROM books ORDER BY author_lname LIMIT 5;
+--------------+
| author_lname |
+--------------+
| Carver       |
| Carver       |
| Chabon       |
| DeLillo      |
| Eggers       |
+--------------+
```

Exercise 1: Print out the book table. Then without input LIMIT command into the SQL terminal, guess the outcome of these commands

```
SELECT title, released_year FROM books ORDER BY released_year DESC LIMIT 1;
```

Ans:

```
SELECT title, released_year FROM books ORDER BY released_year DESC LIMIT
10,1
```

Ans:

```
SELECT title FROM books LIMIT 20, 123219476457;
```

Ans:

Exercise 2: Print out names of the top 5 oldest books in the database
(Format: 2000 - How to Jeeb - J.Kotchakorn)

Exercise 3: Print out names of the top 5 newest books in the database
(Format: 2000 - How to Jeeb - J.Kotchakorn)

Exercise 4: Print out names of the rank 7 - 15 in the database
(Format: 2000 - How to Jeeb - J.Kotchakorn)

LIKE - Additional Arguments for WHERE

```
SELECT <col> FROM <table> WHERE <col> LIKE <arguments>;
```

Arguments on the back has several use:

```
'%<text>%' - use for searching some character that is in a string
```

```
'<____>' - each underscore represent a character
```

It can be modified to have more complexity of the result.

```
mysql> SELECT author_lname FROM books WHERE author_lname LIKE '%m%';
+--------------+
| author_lname |
+--------------+
| Gaiman       |
| Gaiman       |
| Smith        |
| Gaiman       |
+--------------+
```

```
mysql> SELECT author_lname FROM books WHERE author_lname LIKE '_____';
+--------------+
| author_lname |
+--------------+
| Smith        |
+--------------+
```

Exercise 1: Print out a list of books that the author have a in the name
(Formation: Jan Kotchakorn - How to be spicy)

Exercise 2: Print out a list of books that the author name is 4 character long
(Formation: Jan Kotchakorn - How to be spicy)

Exercise 3: Print out a list of books that has the author name is not NULL
(Formation: Jan - How to be spicy)

Exercise 4: Print out a list of books that the title ends with “y”
(Formation: How to be spicy - Jan Kotchakorn)

Exercise 3: Advanced SELECTION

Question 1: Print out all of the books that the title contain the word “stories”

Question 2: Print the book that has the greatest page count (Formation: Book - Page)

Question 3: Print out 3 of the most recent books (Formation: Book - Released Year)

Question 4: Find all the book that the author last name has a space

Question 5: Find the top 3 books that has the lowest quantity of stock

Question 6: Print the title and author name (sorted by author name first then follow by title)

Question 7: Print out author name that is sorted by their last name
(Formation: Tai Verdes - My fav)

Day 3 (Part 2) - Aggregate Function

A function that can operate on multiple roles and pieces of data at once

COUNT - Counting items in the database

COUNT will count the items in the database that is not NULL

```
SELECT COUNT(<col>) FROM <table>;
```

```
mysql> SELECT COUNT(*) FROM books;
+----------+
| COUNT(*) |
+----------+
|       19 |
+----------+
```

Count everything in the list

Exercise 1: Count different every different author full name in the database

Exercise 2: Find how many book title contain the word ‘the’

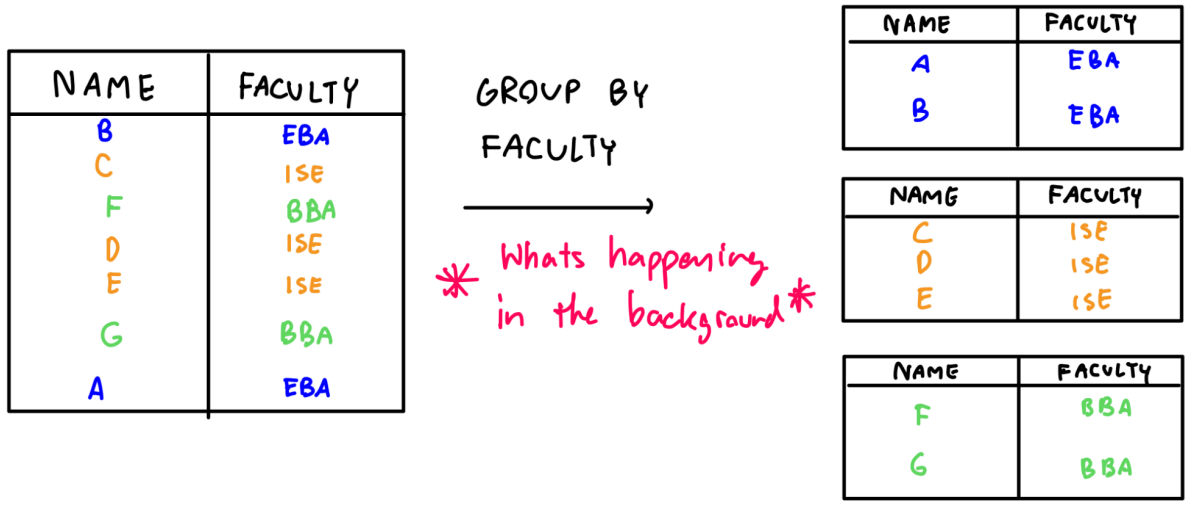
Exercise 3: Find how many person has the lastname Harris

Exercise 4: Count how many books has the stock under 100

GROUP BY - Summarizing or aggregates identical data into a single role

```
SELECT <col1> FROM <table> GROUP BY <col1>;
```

Grouping the data into a sub-table that only contains a specific parameter categorized by a column



THE DATA DOESNT SHOW LIKE THIS

Example 1: Grouping data from the author last name

```
mysql> SELECT author_lname FROM books GROUP BY author_lname;
+-----------------+
| author_lname    |
+-----------------+
| Lahiri          |
| Gaiman          |
| Eggers          |
| Chabon          |
| Smith           |
| Carver          |
| DeLillo         |
| Steinbeck       |
| Foster Wallace  |
| Harris          |
| Saunders        |
+-----------------+
```

Example 2: Counting how many rows is in a group

```
mysql> SELECT author_lname, COUNT(*) FROM books GROUP BY author_lname;
+-----------------+----------+
| author_lname    | COUNT(*) |
+-----------------+----------+
| Lahiri          |        2 |
| Gaiman          |        3 |
| Eggers          |        3 |
| Chabon          |        1 |
| Smith           |        1 |
| Carver          |        2 |
| DeLillo         |        1 |
| Steinbeck       |        1 |
| Foster Wallace  |        2 |
| Harris          |        2 |
| Saunders        |        1 |
+-----------------+----------+
```

Example 3: Using order by and limit to shorten the response

```
mysql> SELECT author_lname, COUNT(*) FROM books GROUP BY author_lname ORDER BY COUNT(*) DESC LIMIT 5;
+-----------------+----------+
| author_lname    | COUNT(*) |
+-----------------+----------+
| Gaiman          |        3 |
| Eggers          |        3 |
| Lahiri          |        2 |
| Foster Wallace  |        2 |
| Carver          |        2 |
+-----------------+----------+
```

Example 4: Using AS to make the command more clear

```
mysql> SELECT author_lname AS Lastname, COUNT(*) AS Books_Written FROM books GROUP BY author_lname ORDER BY Books_Written DESC LIMIT 5;
+---------------+---------------+
| Lastname      | Books_Written |
+---------------+---------------+
| Gaiman        |             3 |
| Eggers        |             3 |
| Lahiri        |             2 |
| Foster Wallace |             2 |
| Carver        |             2 |
+---------------+---------------+
```

Now you have seen the ability to add various arguments to the command to make it shows more complex result. TRY IT YOURSELF

Exercise 1: Group up the book by their released year and show how many book have released in that year

Exercise 2: Group up the book by their author firstname, show how many book they have released. Show only top 5.

MIN & MAX - Maximum and Minimum Value

```
SELECT MIN(<col>) FROM <table>;
SELECT MAX(<col>) FROM <table>;
```

Can be use with both String and Integer

Exercise 1: Find the newest book

Exercise 2: Find the maximum number of pages

Exercise 3: Find the lowest page count

Subqueries - Additional argument for WHERE that gives a more precise result

It is a query within a larger query.

```
SELECT <col> FROM <table> WHERE <subqueries>;
```

Subqueries is another select function with the ability to