



Data Engineering MySQL

Taught by Pichaya Tandayya

18 June 2025

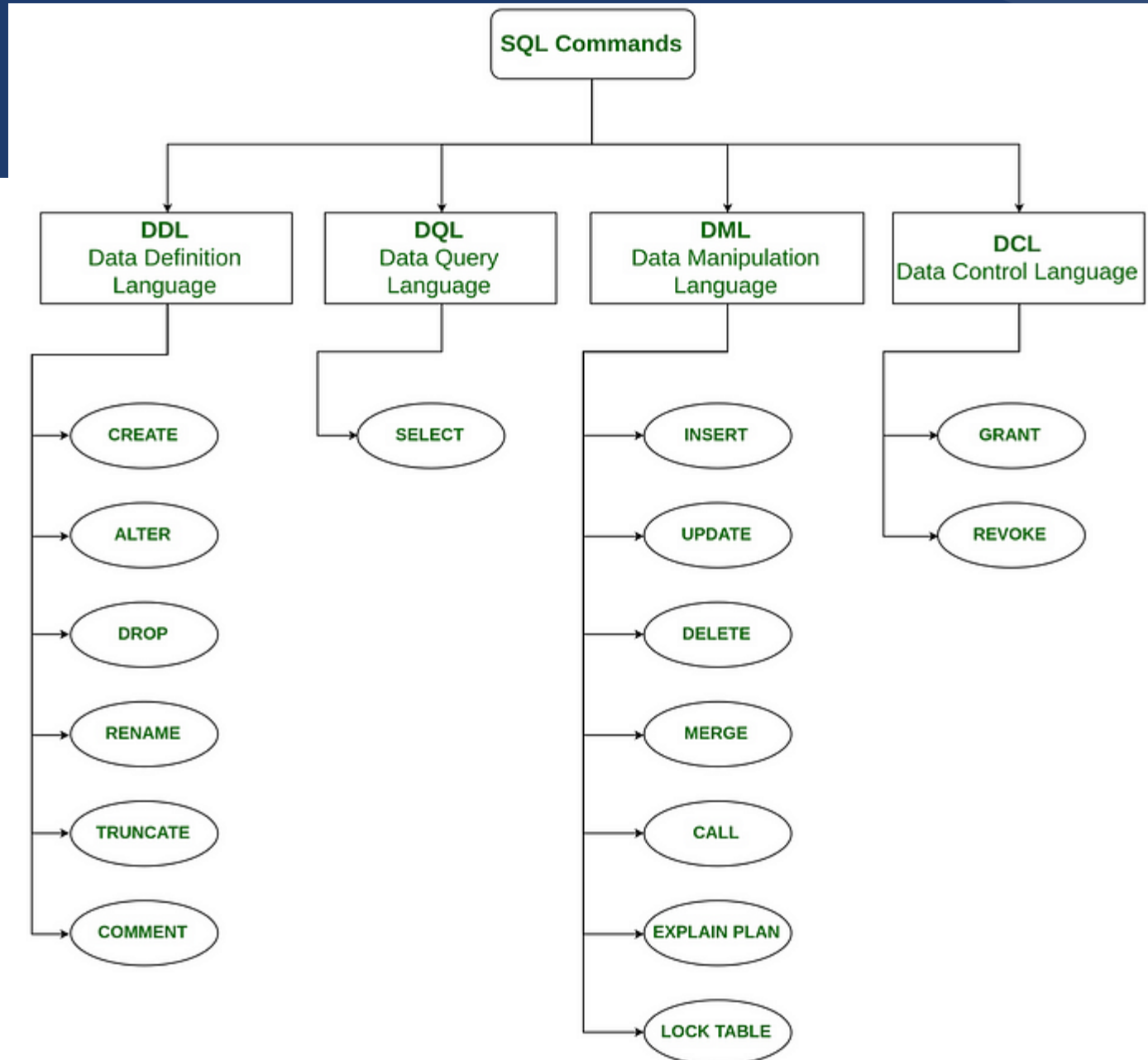
Structure Query Language (SQL)

- The standard for relational database management systems (RDBMS)
- RDBMS: a database management system that manages data as a collection of tables in which all relationships are represented by common values in related tables.

Structure Query Language (SQL)

- SQL Environment
 - **Catalog**: a set of schemas
 - **Schema**: the structure that contains descriptions of objects created by a user (base tables, view, constraints)
 - **Data Definition Language** (DDL): Commands which define a database, including creating, altering, and dropping tables, and establishing constraints
 - **Data Manipulation Language** (DML): Commands which maintain and query a database
 - **Data Control Language** (DCL): Commands which control a database, including administrating privileges and committing data

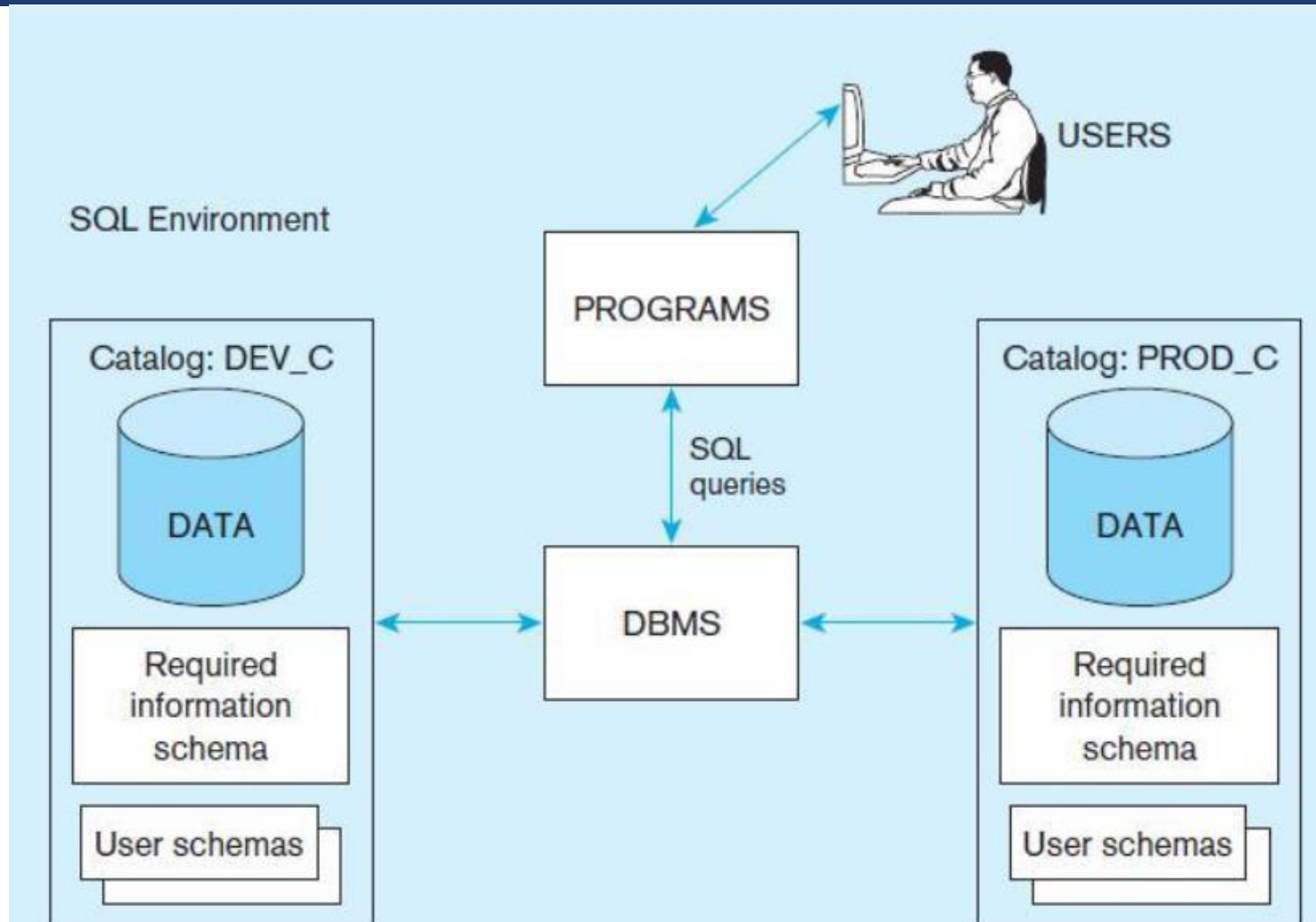
SQL Commands



Source:

<https://prosbeginner.medium.com/%E0%B8%A0%E0%B8%B2%E0%B8%A9%E0%B8%B2-sql-%E0%B8%81%E0%B8%B1%E0%B8%9A-mariadb-part-1-18d28016c616>

Structure Query Language (SQL)



MySQL

- MySQL เป็นระบบจัดการฐานข้อมูลแบบโอเพนซอร์ส (Open Source Database Management system) สำหรับจัดการระบบฐานข้อมูล(Database System) ผ่านทางภาษาเอสคิวแอล SQL (language) โปรแกรมนี้ถูกพัฒนาโดย บริษัท MySQL AB ใน ประเทศสวีเดนสำหรับปัจจุบันผู้เป็นเจ้าของคือบริษัท Oracle มีทั้งแบบใช้ฟรี และเชิงธุรกิจ
- ในการจัดการฐานข้อมูลMySQL จะสามารถใช้โปรแกรมแบบ command-line เพื่อจัดการฐานข้อมูล (โดยใช้คำสั่ง: mysql และ mysqladmin เป็นต้น) หรือจะดาวน์โหลดโปรแกรมจัดการฐานข้อมูลแบบGUI จากเว็บไซต์ของ MySQL ได้แก่ โปรแกรม MySQL Administrator และ MySQL Query Browser เป็นต้น

MySQL

Option	Description
AUTO_INCREMENT	The next auto increment value you want to set for your table (MyISAM)
AVG_ROW_LENGTH	An approximation of the average row length for your table. You only need to set this for tables with variable size records.
CHECKSUM	Set this to 1 if you want MySQL to maintain a checksum for all rows (makes the table a little slower to update but makes it easier to find corrupted tables) (MyISAM)
COMMENT	A 60 character comment for your table
MAX_ROWS	Max number of rows you plan to store in the table
MIN_ROWS	Minimum number of rows you plan to store in the table

MySQL

Option	Description
PACK_KEYS	Set this to 1 if you want to have smaller index. This usually makes updates slower and reads faster (MyISAM, ISAM).
DELAY_KEY_WRITE	Set this to 1 if want to delay key table updates until the table is closed (MyISAM).
ROW_FORMAT	Defines how the rows should be stored (for the future
PRIMARY KEY	Difference Record must not equals
NOT NULL	Must have any thing in record not nul

รูปแบบคำสั่งการสร้าง table

CREATE TABLE data1 (field1, field2, field3,..., fieldn) หรือ
create_definition:
col_name type [NOT NULL | NULL] [DEFAULT default_value]
[AUTO_INCREMENT][PRIMARY KEY]
[reference_definition]
or PRIMARY KEY (index_col_name,...)
or KEY [index_name] (index_col_name,...)
or INDEX [index_name] (index_col_name,...)
or UNIQUE [INDEX] [index_name] (index_col_name,...)

รูปแบบคำสั่งการสร้าง table

CREATE TABLE data1 (field1, field2, field3,..., fieldn) หรือ
create_definition:
col_name type [NOT NULL | NULL] [DEFAULT default_value]
[AUTO_INCREMENT][PRIMARY KEY]
[reference_definition]
or PRIMARY KEY (index_col_name,...)
or KEY [index_name] (index_col_name,...)
or INDEX [index_name] (index_col_name,...)
or UNIQUE [INDEX] [index_name] (index_col_name,...)

คำอธิบาย

CREATE TABLE table1

CREATE TABLE เป็นคำสั่งสร้าง table
table1 เป็นชื่อ table ที่ต้องการสร้าง
field1 เป็นชนิดของcolumn

ชนิดข้อมูลที่ MySQL สนับสนุน

- ชนิดข้อมูลที่เป็นตัวเลข
 - BIT (มีใช้ได้กับ MyISAM, InnoDB, Memory), TINYINT, SMALLINT, MEDIUMINT, INT, BIGINT, Float
- ชนิดข้อมูลที่เกี่ยวข้องกับวันที่และเวลา
 - DATETIME, DATE, TIMESTAMP, TIME, YEAR
- ชนิดข้อมูลที่เกี่ยวข้องกับตัวอักษร
 - CHAR, VARCHAR, BINARY, VARBINARY, BLOB, TEXT, ENUM, SET

SELECT Statement

✖ Used for queries on single or multiple tables

✖ Clauses of the SELECT statement:

+SELECT

✖ List the columns (and expressions) to be returned from the query

+FROM

✖ Indicate the table(s) or view(s) from which data will be obtained

+WHERE

✖ Indicate the conditions under which a row will be included in the result

+GROUP BY

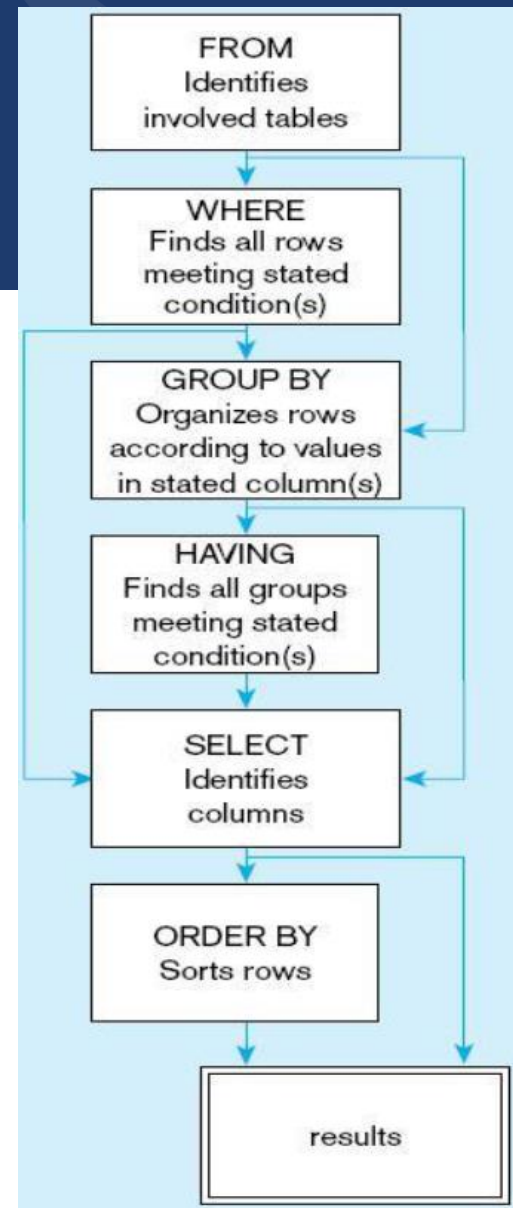
✖ Indicate categorization of results

+HAVING

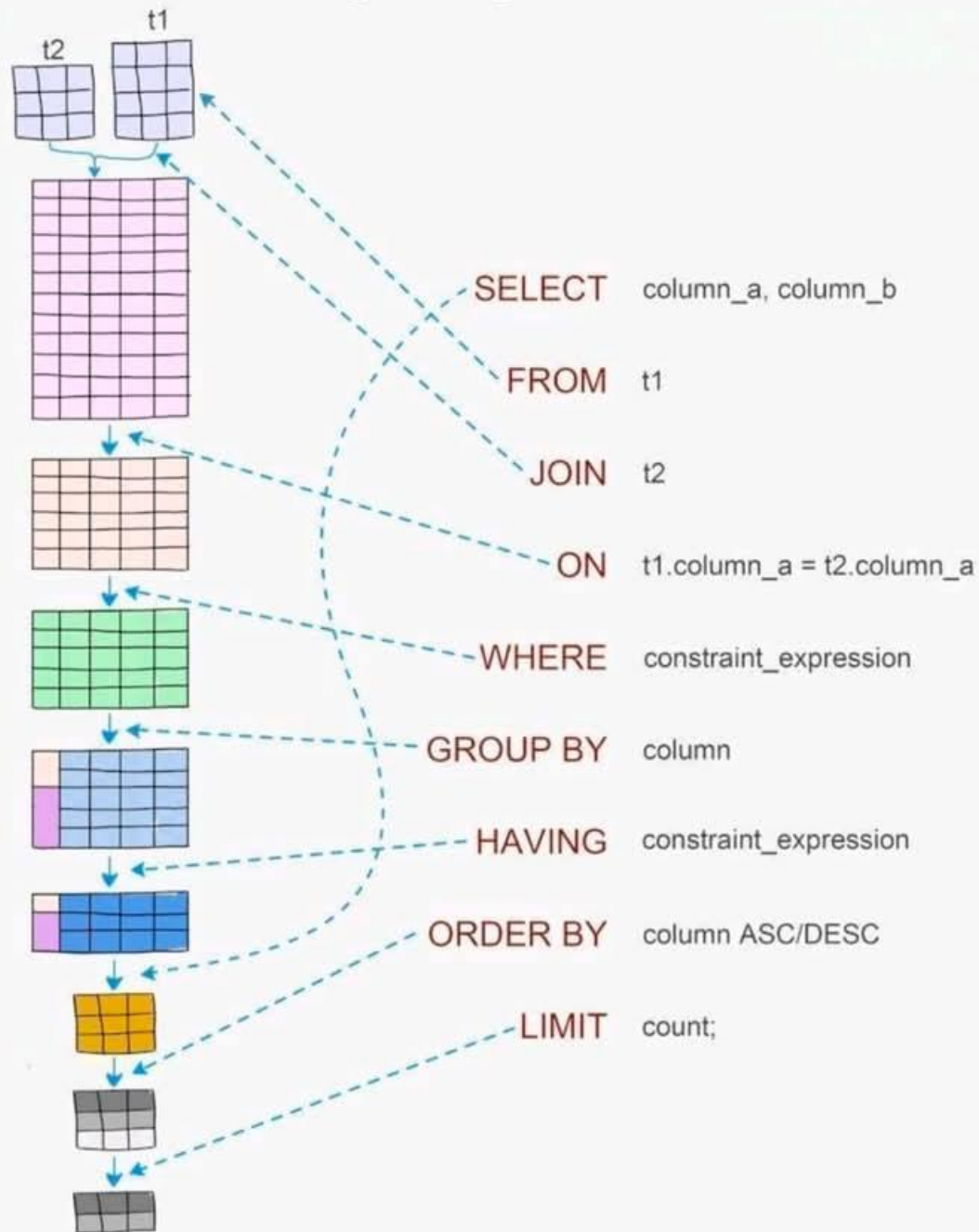
✖ Indicate the conditions under which a category (group) will be included

+ORDER BY

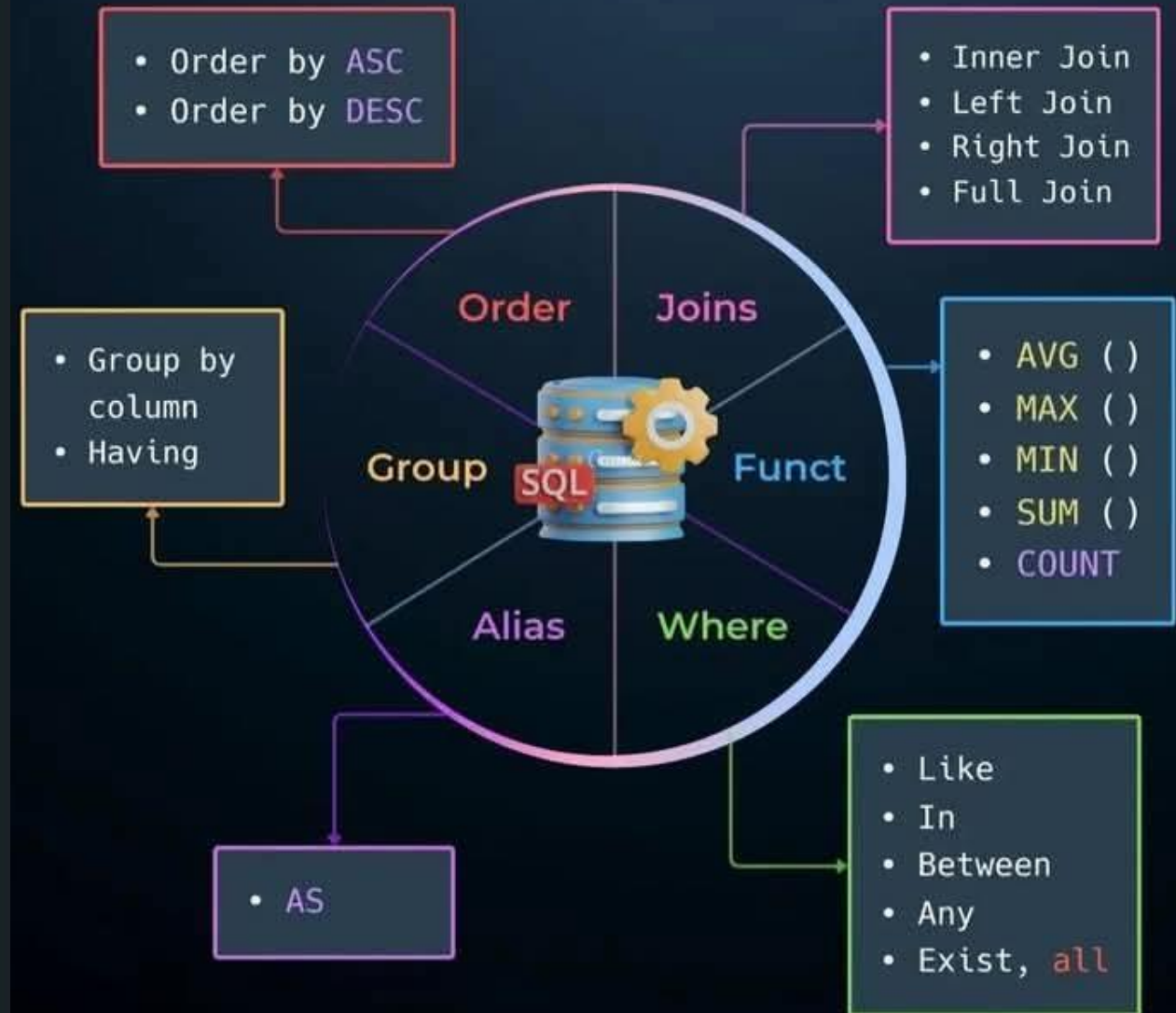
✖ Sorts the result according to specified criteria



SQL Query Logical Order



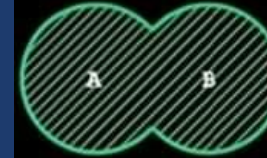
The SQL Circle



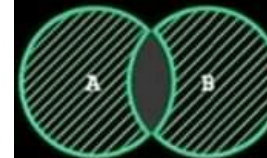
SQL Set Commands



```
1 SELECT *
2 FROM A
3 INNER JOIN B ON A.key = B.key
```



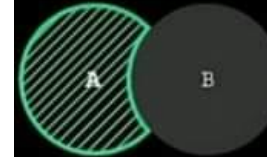
```
1 SELECT *
2 FROM A
3 FULL JOIN B ON A.key = B.key
```



```
1 SELECT *
2 FROM A
3 FULL JOIN B ON A.key = B.key
4 WHERE A.key IS NULL OR
5 B.key IS NULL
```



```
1 SELECT *
2 FROM A
3 LEFT JOIN B ON A.key = B.key
```



```
1 SELECT *
2 FROM A
3 LEFT JOIN B ON A.key = B.key
4 WHERE B.key IS NULL
```



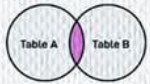
```
1 SELECT *
2 FROM A
3 RIGHT JOIN B ON A.key = B.key
```



```
1 SELECT *
2 FROM A
3 RIGHT JOIN B ON A.key = B.key
4 WHERE B.key IS NULL
```

The 4 Types Of SQL Joins

INNER JOIN

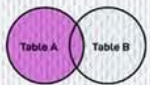


```
SELECT <list_of_columns> FROM
TABLE_A A INNER JOIN TABLE_B B
ON A.KEY = B.KEY
```

Table A	Table B	
A1 1	1 B1	
A2 2	2 B2	
A3 3	4 B3	

Key	Val_A	Val_B
1	A1	B1
2	A2	B2

LEFT JOIN

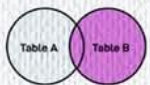


```
SELECT <list_of_columns> FROM
TABLE_A A LEFT JOIN TABLE_B B ON
A.KEY = B.KEY
```

Table A	Table B	
A1 1	1 B1	
A2 2	2 B2	
A3 3	4 B3	
		NULL

Key	Val_A	Val_B
1	A1	B1
2	A2	B2
3	A3	NULL

RIGHT JOIN

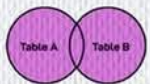


```
SELECT <list_of_columns> FROM
TABLE_A A RIGHT JOIN TABLE_B B
ON A.KEY = B.KEY
```

Table A	Table B	
A1 1	1 B1	
A2 2	2 B2	
A3 3	4 B3	
NULL		

Key	Val_A	Val_B
1	A1	B1
2	A2	B2
3	NULL	B3

FULL OUTER JOIN



```
SELECT <list_of_columns> FROM
TABLE_A A FULL OUTER JOIN
TABLE_B B ON A.KEY = B.KEY
```

Table A	Table B	
A1 1	1 B1	
A2 2	2 B2	
A3 3	4 B3	
NULL		NULL

Key	Val_A	Val_B
1	A1	B1
2	A2	B2
3	A3	NULL
4	NULL	B3

BASIC SQL COMMANDS

SELECT	Select Data From Database
FROM	Specify Table We're Pulling From
WHERE	Filter Query To Match A Condition
AS	Rename Column Or Table With Alias
JOIN	Combine Rows From 2 Or More Tables
AND	Combine Query Conditions. All Must Be Met
OR	Combine Query Conditions. One Must Be Met
LIMIT	Limit Rows Returned. See Also FETCH & TOP
IN	Specify Multiple Values When Using WHERE
CASE	Return Value On A Specified Condition
IS NULL	Return Only Rows With A NULL Value
LIKE	Search For Patterns In Column
COMMIT	Write Transaction To Database
ROLLBACK	Undo A Transaction Block
ALTER TABLE	Add/Remove Columns From Table
UPDATE	Update Table Data
CREATE	Create TABLE, DATABASE, INDEX Or VIEW
DELETE	Delete Rows From Table
INSERT	Add Single Row To Table
DROP	Delete TABLE, DATABASE Or INDEX
GROUP BY	Group Data Into Logical Sets
ORDER BY	Set Order Of Result. Use DESC To Reverse Order
HAVING	Same As WHERE But Filters Groups
COUNT	Count Number Of Rows
SUM	Return Sum Of Column
AVG	Return Average Of Column
MIN	Return Min Value Of Column
MAX	Return Max Value Of Column



SQL

CHEATSHEET

- ✓ Genuine
- ✓ Authentic
- ✓ Quality

Categories

DDL : Data Definition Language
DQL : Data Query Language
DML : Data Manipulation Language
DCL : Data Control Language
TCL : Transaction Control Language

Commands

DDL

CREATE | DROP | ALTER | TRUNCATE
RENAME | COMMENT

DQL

SELECT

DML

INSERT | UPDATE | DELETE | LOCK
CALL | EXPLAIN PLAN

DCL

GRANT | REVOKE

TCL

COMMIT | ROLLBACK
SAVEPOINT | SET TRANSACTION

Operators

Arithmetic

+ - * / %

Bitwise

& | ^

Comparison

= < > <= >= != <> !=

Compound

+= -= *= /= %= &= |= ^=

Logical

AND | OR | NOT | ANY
SOME | ALL | BETWEEN
IN | EXISTS | LIKE
IS NULL | UNIQUE

Important Keywords

WHERE | DISTINCT | LIMIT
ORDER BY | DESC | ASC
AS | FROM | SET | VALUES
CASE | DEFAULT

Database Objects

TABLE | VIEW | SYNONYM
SEQUENCE | INDEX | TRIGGER

Constraints

NOT NULL | UNIQUE
PRIMARY KEY | FOREIGN KEY
CHECK | DEFAULT

Aggregation Functions

AVG | COUNT
MAX | MIN | SUM

Aggregation Keywords

GROUP BY | HAVING

Joins

INNER JOIN



LEFT [OUTER] JOIN



FULL [OUTER] JOIN



RIGHT [OUTER] JOIN



Set Operations

UNION UNION ALL



INTERSECT



EXCEPT MINUS



DDL Examples

Create a Table

```
CREATE TABLE Students(  
  rollno int PRIMARY KEY,  
  fname varchar(255) NOT NULL,  
  lname varchar(255)  
);
```

Adding a new column to the Table

```
ALTER TABLE Students  
ADD email varchar(255);
```

Modifying the data type of existing column

```
ALTER TABLE Students  
ALTER COLUMN lname varchar(512);
```

Removing an existing column from the Table

```
ALTER TABLE Students  
DROP COLUMN email;
```

Truncate (remove all data) a Table

```
TRUNCATE TABLE Students;
```

Drop a Table

```
DROP TABLE Students;
```

DQL Examples

Fetch all data from a Table

```
SELECT * FROM Students;
```

Filter data from a Table

```
SELECT * FROM Students  
WHERE rollno=1234;  
  
SELECT * FROM Students  
WHERE rollno>1234  
AND age<15;
```

Fetch selected columns

```
SELECT fname, lname  
FROM Students  
WHERE rollno>1234  
AND age<15;
```

Fetch maximum 10 rows

```
SELECT fname, lname  
FROM Students  
WHERE rollno>1234  
AND age<15  
LIMIT 10;
```

Fetch count of records

```
SELECT count(*)  
FROM Students;
```

Fetch Maximum Age

```
SELECT max(age)  
FROM Students;
```

Fetch Minimum Age

```
SELECT min(age)  
FROM Students;
```

Fetch Sum of Age

```
SELECT sum(age)  
FROM Students;
```

Fetch Average Age

```
SELECT avg(age)  
FROM Students;
```

Fetch Average Age for each gender

```
SELECT avg(age)  
FROM Students  
GROUP BY gender;
```

Sort (order) fetched records

```
SELECT fname, lname  
FROM Students  
WHERE rollno>1234  
AND age<15  
ORDER BY gender;
```

Sort in descending order

```
SELECT fname, lname  
FROM Students  
WHERE rollno>1234  
AND age<15  
ORDER BY gender DESC;
```

Fetch from 2 Tables

```
SELECT fname, clsteacher  
FROM Students  
INNER JOIN Section  
ON Students.section  
= Section.id;
```

DML Examples

Insert data (rows) into a Table

```
INSERT INTO Students(rollno, fname, lname)  
VALUES (1234, 'Christiano', 'Ronaldo');
```

Update data (value of column) of a Table

```
UPDATE Students SET lname = 'Messi'  
WHERE rollno=1234;
```

Delete data (rows) from a Table

```
DELETE FROM Students WHERE rollno=1234;
```

Aggregate and, Filter

```
SELECT section, count(*) AS studentcount  
FROM Students  
GROUP BY section  
HAVING count(*) > 20;
```

Full Outer Join

```
SELECT fname, clsteacher  
FROM Students  
FULL JOIN Section  
ON Students.section = Section.id;
```


เริ่มต้นใช้งาน MySQL

1. ใช้งาน MySQL ผ่าน command prompt เบื้องต้น

- 1.1 ฝึกการเข้าและออกจาก MySQL
- 1.2 แสดงรายชื่อ database
- 1.3 เรียกใช้หรือเข้าไปในฐานข้อมูล
- 1.4 แสดงรายชื่อ table ในระบบฐานข้อมูล
- 1.5 สร้างฐานข้อมูลใหม่
- 1.6 เรียกใช้ฐานข้อมูลและสร้างตารางใหม่
- 1.7 เพิ่มระเบียนใหม่ในตาราง
- 1.8 เรียกข้อมูลบาง field หรือ ทุก field มาแสดง
- 1.9 แสดง field ในตาราง
- 1.10 หาผลรวมของบาง field
- 1.11 ลบข้อมูลบางระเบียน
- 1.12 เปลี่ยนแปลงข้อมูลในบางระเบียน

2. การปรับปรุงข้อมูลใน MySQL โดย command prompt

- 2.1 ดูสถานะของ MySQL
- 2.2 สร้างตารางใหม่
- 2.3 ลบตารางที่มีอยู่

การใช้งาน MySQL ผ่าน command prompt เบื้องต้น

Click Start, Run and type powershell

`c:\>` cd path ที่ติดตั้ง MySQL เช่น

`c:\mysql\bin>`mysql -u root

ถ้าหากเชื่อมต่อกับ MySQL server ได้สำเร็จจะมีเครื่องหมายรับคำสั่ง `mysql>` หรือ `mariadb>`

หากต้องการออกจาก MySQL server ให้พิมพ์ exit หรือ quit เพื่อออกจาก MySQL

`mysql>` exit

การใช้งาน MySQL ผ่าน command prompt เบื้องต้น

1.2 แสดงรายชื่อ database

```
mysql> show databases;
```

Note คำสั่งSQL จะปิดท้ายด้วยเครื่องหมาย ; เสมอ เมื่อเริ่มต้นติดตั้ง จะมีฐานข้อมูลชื่อ MySQL และ test มาให้

1.3 เรียกใช้ หรือเข้าไปในฐานข้อมูลชื่อ MySQL

```
mysql> use mysql;
```

1.4 แสดงรายชื่อ table ในระบบฐานข้อมูล MySQL

```
mysql> show tables;
```

1.5 สร้างฐานข้อมูลใหม่ ชื่อ lab

```
mysql> create database lab;
```

```
mysql> show databases;
```

การใช้งาน MySQL ผ่าน command prompt เบื้องต้น

1.6 เรียกใช้ฐานข้อมูลและสร้างตารางใหม่

```
mysql> use lab;
```

```
mysql> create table customer(id int primary key , name varchar(80) , credit double);
```

```
mysql> show tables;
```

ส่วนประกอบโครงสร้าง ของตาราง (option)

การสร้าง table มีรูปแบบคำสั่งดังนี้

- CREATE TABLE data1 (field1, field2, field3,..., fieldn)
- หรือ create_definition:
- col_name type [NOT NULL | NULL]
[DEFAULT default_value]
[AUTO_INCREMENT][PRIMARY KEY]
[reference_definition]
- or PRIMARY KEY (index_col_name,...)
- or KEY [index_name]
(index_col_name,...) or INDEX
[index_name] (index_col_name,...) or
UNIQUE [INDEX] [index_name]
(index_col_name,...)

Option	Description
AUTO_INCREMENT	The next auto increment value you want to set for your table (MyISAM)
AVG_ROW_LENGTH	An approximation of the average row length for your table. You only need to set this for tables with variable size records.
CHECKSUM	Set this to 1 if you want MySQL to maintain a checksum for all rows (makes the table a little slower to update but makes it easier to find corrupted tables) (MyISAM)
COMMENT	A 60 character comment for your table
MAX_ROWS	Max number of rows you plan to store in the table
MIN_ROWS	Minimum number of rows you plan to store in the table
PACK_KEYS	Set this to 1 if you want to have smaller index. This usually makes updates slower and reads faster (MyISAM, ISAM).
PASSWORD	Encrypt the .frm file with a password. This option doesn't do anything in the standard MySQL version.
DELAY_KEY_WRITE	Set this to 1 if want to delay key table updates until the table is closed (MyISAM).
ROW_FORMAT	Defines how the rows should be stored (for the future).
PRIMARY KEY	Difference Record must not equals
NOT NULL	Must have any thing in record not null

ตัวอย่าง

```
CREATE TABLE table1 (field1, field2, field3,..., fieldn)
```

CREATE TABLE เป็นคำสั่งสร้างตาราง

table1 เป็นชื่อ table ที่ต้องการสร้าง

field1-n เป็นชนิดของcolumn

Data Types

- ชนิดข้อมูลที่เป็นตัวเลข

- BIT (มีใช้ได้กับ MyISAM, InnoDB, Memory)
- TINYINT
- SMALLINT
- MEDIUMINT
- INT
- BIGINT
- Float

- ชนิดข้อมูลที่เกี่ยวข้องกับวันที่และเวลา

- DATETIME
- DATE
- TIMESTAMP
- TIME
- YEAR

- ชนิดข้อมูลที่เกี่ยวข้องกับตัวอักษร

- CHAR
- VARCHAR
- BINARY
- VARBINARY
- BLOB
- TEXT
- ENUM
- SET

การใช้งาน MySQL ผ่าน command prompt เบื้องต้น

1.7 เพิ่มระเบียบใหม่ในตาราง

```
mysql> insert into customer values(1, 'First One', 10000.15);
```

```
mysql> insert into customer values(2, 'Second Two', 20000.25);
```

```
mysql> insert into customer values(3, 'Third Three', 30000.35);
```

- รูปแบบคำสั่ง insert

INSERT [INTO]

{ table_name WITH (<table_hint_limited> [..n]) | view_name | rowset_function_limited } { [(column_list)] {
VALUES ({ DEFAULT | NULL | expression } [..n]) | derived_table | execute_statement } }

- หมายเหตุ ข้อมูลตัวอักษรจะอยู่ในเครื่องหมายคำพูด เช่น ' ' (single quotes) หรือ " " (double quotes)

การใช้งาน MySQL ผ่าน command prompt เบื้องต้น

1.8 เรียกข้อมูลบาง field หรือ ทุก field มาแสดง

```
mysql> select credit from customer;
```

```
mysql> select * from customer;
```

- รูปแบบคำสั่ง select

```
SELECT [ALL | DISTINCT] select_list [INTO new_table]
```

```
FROM table_source
```

```
[where search_condition]
```

```
[GROUP BY group_by_expression] [HAVING search_condition]
```

```
[ORDER BY order_expression [ASC | DESC]]
```

การใช้งาน MySQL ผ่าน command prompt เบื้องต้น

1.9 แสดง field ในตาราง

หลังใช้คำสั่งสร้างตาราง หากดูผลการสร้างก็ใช้คำสั่ง show columns หรือคำสั่ง desc

```
mysql> show columns from customer;
```

```
mysql> desc customer;
```

1.10 หาผลรวมของบาง field

```
mysql> select sum(credit) from customer;
```

การใช้งาน MySQL ผ่าน command prompt เบื้องต้น

1.11 ลบข้อมูลบางระเบียน

```
mysql> delete from customer where id = 2;
```

```
mysql> select * from customer;
```

- รูปแบบคำสั่ง delete

DELETE

{ table_name WITH (<table_hint_limited> [..n]) | view_name | rowset_function_limited } [FROM { <table_source> } [..n]]

[where <search_condition>]

การใช้งาน MySQL ผ่าน command prompt เบื้องต้น

1.12 เปลี่ยนแปลงข้อมูลในบางระเบียน

```
mysql> update customer set name='First Count' where id = 1;
```

```
mysql> select * from customer;
```

- รูปแบบคำสั่ง update

UPDATE

{ table_name WITH (<table_hint_limited> [..n]) | view_name | rowset_function_limited } SET

column_name = { expression | default | null } { [FROM { <table_source> } [..n]] [where <search_condition>] }

การปรับปรุงฐานข้อมูลใน MySQL โดย command prompt

2.1 ดูสถานะของ MySQL

```
mysql> status
```

```
mysql> show tables;
```

2.2 สร้างตารางใหม่

```
mysql> create table new(id int primary key);
```

```
mysql> show tables;
```

2.3 ลบตารางที่มีอยู่

```
mysql> drop table new;
```

```
mysql> show tables;
```

2.4 สร้างฐานข้อมูลใหม่ชื่อ labnew

```
mysql> show databases;
```

```
mysql> create database labnew;
```

```
mysql> show databases;
```

2.5 ลบฐานข้อมูลชื่อ labnew

```
mysql> show databases;
```

```
mysql> drop database labnew;
```

```
mysql> show databases;
```

Assignment

1. จงเขียนคำสั่งเพื่อสร้างฐานข้อมูลใหม่ชื่อ topic
2. จงเขียนคำสั่งเพื่อเรียกใช้ฐานข้อมูล topic แล้วสร้างตารางใหม่ชื่อ student ที่เก็บข้อมูล id ชนิด int และ name ชนิด varchar(50)
3. จงเขียนคำสั่งเพื่อเพิ่มข้อมูลจำนวนหนึ่งระเบียนลงในตาราง customer ของฐานข้อมูล lab ด้วยข้อมูล id = 9 ตามด้วย name เป็นชื่อภาษาอังกฤษของผู้ทดลอง (เช่น Kamon) และ credit เป็น 9999.9
4. จงเขียนคำสั่งเพื่อแสดงข้อมูลทั้งหมดของตาราง customer ของฐานข้อมูล lab
5. จงเขียนคำสั่งเพื่อเปลี่ยนแปลงข้อมูลหนึ่งระเบียนของตาราง customer ของฐานข้อมูล lab ที่มีค่า id = 2 โดยเปลี่ยนแปลงให้ค่าของ name เป็นชื่อภาษาอังกฤษของผู้ทดลอง (เช่น Kamon)

เอกสารอ้างอิง

- Vikram Wasvani, How to do Everything with PHP & MySQL, McGraw Hill, **2005**.
- <https://www.geeksforgeeks.org/extract-data-from-database-using-mysql-connector-and-xampp-in-python/>
- <https://mariadb.com/resources/datasheets/mariadb-standard-developer-cheat-sheet/>

