

11/11

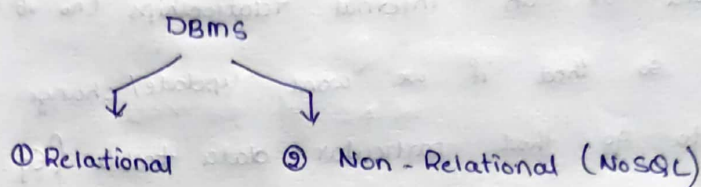
SQL

→ Structured Query Language

Data Base:

collection of data stored in a format that can be easily accessed.

→ To manage our data bases, we use a software application called Data Base Management System (DBMS).



① Relational Data Bases Management Systems:

we store data in tables that are linked to each other using relationships.

Each table stores data about a specific type of object like customers, orders, products.

→ we use SQL to query & modify our data.

Most popular RDBMS:

1. MySQL

2. SQL Server (by MS)

3. Microsoft Oracle

→ MySQL

It is the most popular open source data base in the world.

② Non-Relational DBMS (NoSQL)

→ These systems don't understand SQL. They have their own query language.

# SQL with MySQL

## \* Creating the Data Bases

In every data base we have tables (we store data) and we have views (virtual tables) so that we can combine different tables, and put them in a view.

→ There will be an internal relationships (no. of) between the tables, so that if we want update/change data, we have to go that particular data table (eg: customers, orders)

The Select Statement [How to retrieve data from a single panel]

→ For 1<sup>st</sup> step to write query: (to get data ← database)

1. Select a data base. [Query written will be executed against that data base]

1. USE sql\_store; → database

To retrieve data from this data base

2. SELECT customer\_id, first-name

columns we want to retrieve  
or we can use \* to retrieve all columns.

3. FROM customers

→ table we want to query.



4) WHERE customer\_id = 1



to retrieve data of only a specific customer

Use " -- " → hyphens to make a comment.

5) ORDER BY first\_name



sorted in order by first name.

Select Clause

SELECT

FROM

WHERE

ORDER BY

order is important

otherwise it will be a syntax error

→ Line breaks are not necessary.

→ follows basic arithmetic rules (BODMAS)

→ We can use " AS " to give name.

Es: (points + 10) \* 100 AS discount\_factor

alias

Use ' ( ) ' " "

to get space in middle.

→ SELECT DISTINCT state

FROM customers

for unique list with no duplicates.

13/11

## The WHERE clause

we can use  $>$ ,  $>=$ ,  $<$ ,  $<=$  operations

like these in the where clause.

→ WHERE state = 'VA'

→ dates should be enclosed in single quotes

birth\_date > '1990-01-01'



standard format

→ eg: Get Orders placed this year ;

SELECT \*

FROM 'orders'

WHERE date\_d >= '2022-01-01'

→ AND, OR, ~~AND~~ and NOT operators

we can add another condition by

~~AND~~ points > 1000 (but both conditions should be true).

(OR) → any one condition can be true.

AND has more priority over OR while using multiple operations.

order is very important like BODMAS

AND > OR

To get opposite result;

WHERE NOT birthdate > '1990-01-01'.

↓  
like using negation.

OR  $\rightarrow$  AND

AND  $\rightarrow$  OR.

$\rightarrow$  we can use arithmetic expression in WHERE clause. (not only on select)

IN operator

$\rightarrow$  For replacing multiple OR operation.

WHERE state = 'VA' OR state = 'FL' OR state = 'MA'

↓  
WHERE state ~~IN state~~ IN ('VA', 'FL', 'MA')

Exercise Return products with quantity in stock equal to 49, 38, 72

SELECT \*

FROM stocks

WHERE quantity IN (49, 38, 72)

Between operator

WHERE points >= 1000 AND points <= 3000

↓  
WHERE points BETWEEN 1000 AND 3000



## Like operator

```
SELECT *  
FROM customers  
WHERE last_name LIKE 'b%'
```

⇒ gives all ~~data~~ customers whose last name starts with b

To indicate, there can be any number of characters after b.

⇒ % b %

↳ b is somewhere in the last name.

% y

↳ ends with y

⇒ '\_y'

↳ ends with y & should have only one character before

⇒ Use NOT LIKE to get opposite result.

14/11

## The REGEXP operator

Reg. expression → powerful expression for searching strings.

WHERE last\_name REGEXP 'field' → search for any where

should start with ← '^field'

should end with ← 'field\$'

for searching multiple strings

'field | mac' → any one

'field | mac | str'

→ '[gim]e' → ~~ends~~ with ge (s) ie (s) me

'[a-h]e'

## Exercise

SELECT \* FROM customers

WHERE first\_name REGEXP 'ELKA | AMBUR'

WHERE last\_name REGEXP 'EY\$ | ON\$'

WHERE last\_name REGEXP '^MY | SE'

WHERE last\_name REGEXP 'B[RU]'

## Null operator

WHERE phone IS NULL.

## The ORDER BY clause

by using DESC (sorted in descending order)

### \* LIMIT CLAUSE

To obtain only some limited no. of details of particular data.

SELECT \*

FROM customers

LIMIT 3

→ only 3 details.

**\*\*** LIMIT

6, 3

↓  
skipping the starting  
6 data

↓  
Printing the next 3 data.

### Exercise

Top 3 loyal customers: (most points)

SQL

SELECT \*

FROM customers

ORDER BY points DESC

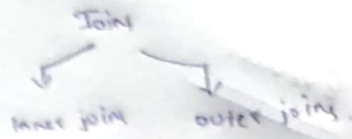
LIMIT 3.

IMP  
**\*\***

LIMIT clause should always come at end.



Inner join :



```
SELECT *  
FROM orders  
JOIN customers  
ON order.customer_id = customers.customer_id
```

output

all tables of orders

1				
2				
3				

→ We can also join the table to itself

[like in organization finding employees & their managers]

USE sql\_hr;  
→ respective database

SELECT

e.employee\_id,  
e.first\_name,  
m.first\_name AS manager

FROM employees e

JOIN employees m

ON e.reports\_to = m.employee\_id

\* Joining more than two tables :

what ever we require

USE sql\_store; → data base

SELECT \*

FROM orders o

JOIN customers c ON o.customer\_id = c.customer\_id

JOIN order\_statuses os  
ON o.status = os.order\_status\_id

o.order\_id  
o.order\_date  
c.first\_name  
c.last\_name  
os.name AS status

## Exercise

USE sql\_invoicing;

SELECT

name\_of\_the\_client  
payment\_method.

\*

From payments p

JOIN clients c

ON p.client\_id = c.client\_id