02 - Retrieving Data From a Single Table

Thursday, March 21, 2024 12:51 AM

select which Database to work with

```
USE
USE student;
```

Comments

```
--
-- SELECT * FROM STUDENT;
```

Tips & Best Practice

- **Recommend** to keep saved words in sql as Captial --> SELECT FROM WHERE …etc
- **Order** of syntax of query is matter like that [SELECT FROM WHERE ORDER BY]
- **All Query Part except SELECT Word** -> SELECT 1 , 2; are valid Query
- **Keep Each Clause** in one line as query go longer it's become readable
- Date values in SQL Must be in Quote --> '2024-03-21' even date not string value

YY- MM-DD

SELECT Clause and Rename The Result

```
SELECT * FROM X;
SELECT a , b FROM X;
SELECT a * 10  as 'My Calculation' FROM X;
```

SELECT Unique Result

```
SELECT DISTINCT department FROM X;
```

WHERE Clause & Multi Conditions AND OR NOT

Type of Filter [> , >= , < ,<= , = , != or <>]

- ```
  SELECT * FROM CUSTOMER
  WHERE id > 5 ;
  ```
- ```
  SELECT * FROM CUSTOMER
  WHERE birthdate  > '2024-03-21' ;
  ```
- ```
  SELECT * FROM customer
  WHERE id > 5 AND points >= 1000;
  ```
  ```

IN Operator

- When You Have a lot of things need to search if item [x] in this set if items
  Example : Get all info for customer if his state is VA , CA , UN

```
SELECT  *
FROM customer
WHERE state IN ('VN' , 'CA' ,'UN ') ;
```

```
SELECT  *
FROM customer
WHERE state NOT IN ('VN' , 'CA' ,'UN ') ;
```

#BETWEEN Operator

Like When you need to check in range

```
SELECT *
FROM customer
WHERE points BETWEEN 1000 AND 3000
```

```
SELECT  *
FROM customers
WHERE birth_date BETWEEN '1990-01-01' AND '2000-01-01';
```

LIKE Operator

For String Matching Used a lot in Search and Filter in real Applications

- % Any chars

'%d'	--> Mean string has last char = 'd' and have any numbers of chars before
'%d%'	--> Mean string has in it char = 'd' and has before and after it any numbers of chars

- _ Only one Char

'_d' -->	Mean String has #Length = 2 where first char is any thing but the second must be 'd'
'd_'-->	Mean String has #Length = 2 where first char must be 'd' and second can be anything

```
SELECT *
FROM customers
```

```
WHERE last_name LIKE '%m';

SELECT *
FROM customers
WHERE last_name LIKE '__X';
```

NULL Values

Null : simply mean the absence of the value

- You can Query on data to know if there are **null** values using **IS NULL** Keyword

```
SELECT *
FROM customer
WHERE email IS NULL ;
```

- You can also query by reverse the meaning by getting all that have values **NOT NULL**

```
SELECT *
FROM customer
WHERE email IS NOT NULL ;
```

Sort Your Query Result

You Can Sort Your Query Result Based On what you want like using Keyword **ORDER BY [ASC - DESC]**

- Any table column
- You can Make an **Allies**
- You can Even make it by not selected column
- You can Make it even with data generated like say sort based on value of (X * Y)

```
SELECT  *
FROM   X
ORDER BY last_name DESC

SELECT  *
FROM   X
ORDER BY ( quantity * price_unit ) DESC
```

Limit Your Returned Data Count

You can Control the number of data that returned from the query set using **LIMIT** keyword

```
SELECT  *
FROM orders
LIMIT 5                 // Telling Just Get the First 5 Record
```

You Can Also use offset with LIMIT to Skip Some Record like that

```
SELECT  *
FROM orders
LIMIT 6 , 3             // Skip the first 6 records and Start Selecting from the 7th
```