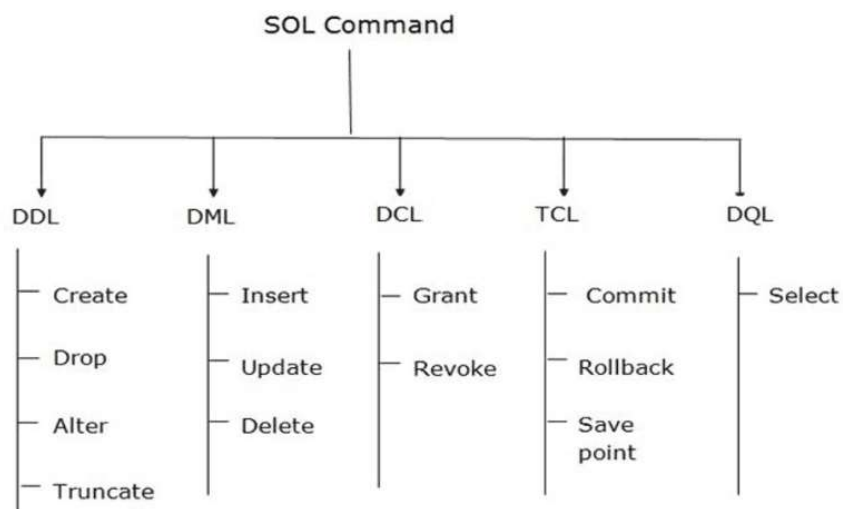


## BASIS DATA DASAR

SQL adalah singkatan dari Structured Query Language yang merupakan suatu bahasa pemrograman untuk menjalankan serta membuat perintah menyimpan dan mengelola Data pada Relational Database Management System (RDBMS). Relational Database Management System (RDBMS) merupakan suatu perangkat lunak yang memiliki fungsi untuk melakukan pengeditan, perubahan, menambahkan, serta memperbaharui suatu query. Selain itu dibidang Big Data Analytics Relational Database Management System (RDBMS) juga banyak digunakan dalam melakukan Data Visualization. SQL adalah Bahasa standar yang digunakan dalam Relational Database Management System (RDBMS). Jenis jenis Relational Database Management System (RDBMS) yang paling populer digunakan saat ini seperti Oracle, Informix, MySQL dan SQL Server menggunakan SQL sebagai bahasa standar mereka.

Perintah Dasar SQL adalah suatu query yang digunakan untuk melakukan komunikasi dengan Database. Komunikasi ini berfungsi untuk merubah, menambahkan, mengubah sutau data dalam database. Beberapa perintah yang dapat dijalankan antara lain seperti membuat tabel, membuat kolom, membuat baris, menghapus baris, memodifikasi table dan lainnya. Ada beberapa Tipe dalam Perintah Dasar SQL antara lain : DDL, DML, DCL, TCL, and DQL.



### Fundamental Perintah SQL

Fundamental pada SQL ada beberapa macam berikut pembahasannya

#### CREATE

CREATE pada SQL berfungsi untuk membuat tabel

Syntax:

```
CREATE TABLE "TABLE_NAME"(  
"COLUMN1" "DATA_TYPE" CONSTRAINTS,  
"COLUMN2" "DATA_TYPE" CONSTRAINTS,  
"COLUMN3" "DATA_TYPE" CONSTRAINTS,  
.....  
"COLUMN N" "DATA_TYPE" CONSTRAINTS  
);
```

## INSERT

INSERT pada SQL berfungsi untuk memasukkan atau menambahkan data baru pada tabel

Syntax:

```
INSERT INTO  
"TABLE_NAME" (COL1, COL2, .....COL_N)  
VALUES (Col_val_1,Col_val_2, ..... Col_val_N);
```

## SELECT

SELECT pada SQL berfungsi untuk mengambil data dari tabel

Syntax

```
SELECT * FROM "TABLE_NAME";
```

Select Satu Kolom

```
SELECT "COLUMN_NAME" FROM "TABLE_NAME";
```

Select multiple columns

```
SELECT "COLUMN1,COLUMN2,..."  
FROM "TABLE_NAME";
```

Select all columns

```
SELECT * FROM "TABLE_NAME";
```

## DISTINCT

DISTINCT berfungsi menghilangkan seluruh duplikat data dan menyisakan data yang unik saja

Syntax:

```
SELECT DISTINCT(*) FROM "TABLE_NAME";
```

## WHERE

WHERE berfungsi untuk melakukan filter dalam pengambilan data dengan suatu kondisi tertentu

Syntax:

```
SELECT "COLUMN_NAME(S)"  
FROM "TABLE_NAME "  
WHERE CONDITION;
```

## AND/OR

AND/OR berfungsi untuk mengcombine multiple conditions

Syntax:

```
SELECT "COLUMN_NAMES(s)"  
FROM "TABLE_NAME"  
WHERE CONDITION AND/OR CONDITION;
```

## UPDATE

UPDATE berfungsi untuk melakukan modifikasi suatu data sebelumnya (existing data) table

Syntax:

```
UPDATE "TABLE_NAME"  
SET COL_1=VAL_1,COL_2=VAL_2,...  
WHERE CONDITION;
```

## DELETE

DELETE berfungsi untuk menghapus existing data pada tabel

Syntax:

For delete all rows

```
DELETE FROM "TABLE_NAME";
```

For delete single/multiple row(s)

```
DELETE FROM "TABLE_NAME "  
WHERE CONDITION;
```

## ALTER

ALTER berfungsi untuk merubah struktur dari suatu tabel

Syntax:

ADD COLUMN

```
ALTER TABLE "TABLE_NAME"  
ADD "COLUMN_NAME" "DATA_TYPE";
```

DROP COLUMN

```
ALTER TABLE "TABLE_NAME"  
DROP "COLUMN_NAME";
```

MODIFY DATA TYPE

```
ALTER TABLE "TABLE_NAME"  
ALTER COLUMN "COL_NAME" TYPE NEW_DATA_TYPE;
```

RENAME COLUMN

```
ALTER TABLE "TABLE_NAME"  
RENAME COLUMN "COL_NAME" TO "NEW_NAME";
```

ADD CONSTRAINTS

```
ALTER TABLE "TABLE_NAME"  
ADD CONSTRAINT COL_NAME CHECK CONDITION;
```

## Perintah Filtering Pada SQL

Filtering pada SQL berfungsi untuk mengambil data dari suatu database dengan suatu kondisi tertentu. Berikut merupakan beberapa perintah Filtering pada SQL

### IN

IN dalam SQL Digunakan untuk mengurangi beberapa perintah dalam memilih, menghapus atau menyisipkan suatu pernyataan tertentu.

Syntax:

```
SELECT "COL_NAME" FROM "TABLE_NAME"  
WHERE "COL_NAME" IN ('VAL1', 'VAL2',...);
```

## **BETWEEN**

Between dalam SQL Berfungsi untuk mengambil suatu data dalam range atau rentang tertentu

Syntax:

```
SELECT "COL_NAME(S)" FROM "TABLE_NAME"  
WHERE "COL_NAME" BETWEEN "VAL1" AND "VAL2";
```

## **LIKE**

Like dalam SQL Digunakan untuk melakukan pencocokan pola suatu data tertentu

Syntax:

```
SELECT "COL_NAME" FROM "TABLE_NAME"  
WHERE "COL_NAME" LIKE 'PATTERN';
```

## **Perintah Ordering Pada SQL**

Ordering pada SQL berfungsi untuk menyeleksi urutan atau mengurutkan suatu data dari database dengan suatu kondisi tertentu. Berikut merupakan beberapa perintah Ordering pada SQL

### **ORDER BY**

Order By Digunakan untuk mengurutkan data & hanya digunakan di SELECT satatement

Syntax:

```
SELECT "COL_NAME(S)" FROM "TABLE_NAME"  
ORDER BY "COL_NAME" ASC/DESC;
```

### **LIMIT**

Limit dalam SQL Digunakan untuk membatasi jumlah data berdasarkan suatu limit tertentu yang telah ditentukan

Syntax:

```
SELECT "COL_NAME(S)" FROM "TABLE_NAME"  
[WHERE & ORDER BY – Optional]  
LIMIT "LIMIT_VALUE";
```

## **Perintah Grouping Pada SQL**

Grouping pada SQL berfungsi untuk mengelompokkan suatu data berdasarkan satu atau beberapa kolom sekaligus. Berikut merupakan beberapa perintah Grouping pada SQL

### **GROUP BY**

GROUP BY dalam SQL digunakan untuk mengelompokkan suatu hasil berdasarkan satu atau beberapa kolom sekaligus

Syntax:

```
SELECT "COL_1", "COL_2",..... FROM "TABLE_NAME"  
GROUP BY "COL_NAME";
```

### **HAVING**

HAVING ditambahkan ke SQL karena Kata kunci WHERE tidak dapat digunakan dengan agregat Fungsi

Syntax:

```
SELECT "COL_1", "COL_2",..... FROM "TABLE_NAME"  
GROUP BY "COL_NAME"  
HAVING 'CONDITION';
```

### Soal Latihan

1. Select all records from the Customers table, sort the result alphabetically, first by the column Country, then, by the column City.

```
SELECT * FROM Customers  
ORDER BY Country, City;
```

2. Select all records where the City column has the value 'Berlin' and the PostalCode column has the value '12209'.

```
SELECT * FROM Customers  
WHERE City = 'Berlin'  
AND PostalCode = '12209';
```

3. Select all records where the City column has the value 'Berlin' OR 'London'.

```
SELECT * FROM Customers  
WHERE City = 'Berlin'  
OR City = 'London';
```

4. Use the NOT keyword to select all records where City is NOT "Berlin".

```
SELECT * FROM Customers  
WHERE NOT City = 'Berlin';
```

5. Select all records where the value of the City column starts with the letter "a".

```
SELECT * FROM Customers  
WHERE City LIKE 'a%';
```

6. Select all records where the value of the City column starts with letter "a" and ends with the letter "b".

```
SELECT * FROM Customers  
WHERE City LIKE 'a%b';
```

7. Select all records where the second letter of the City is an "a".

```
SELECT * FROM Customers  
WHERE City LIKE '_a%';
```

8. Select all records where the first letter of the City starts with anything from an "a" to an "f".

```
SELECT * FROM Customers  
WHERE City LIKE '[a-f]%' ;
```

9. Insert the missing parts in the JOIN clause to join the two tables Orders and Customers, using the CustomerID field in both tables as the relationship between the two tables.

```
SELECT *  
FROM Orders  
LEFT JOIN Customers  
ON Orders.CustomerID=Customers.CustomerID;
```

10. Tampilkan data dari table KARYAWAN dengan jumlah record empat (4) dan urutkan secara ascending berdasarkan Nama karyawan.

```
SELECT * FROM karyawan  
ORDER BY nama  
LIMIT 4;
```

11. Tampilkan data products mulai dari data ke 4 dan hanya 6 data yang ditampilkan.

```
SELECT * FROM products LIMIT 4, 6
```