BASIC SQL

- **What is SQL?**

✓ SQL stands for <u>Structured Query Language</u>. It is used for storing and managing data in relational database management system (RDBMS)

- **<u>SQL Commands</u>:**

● <u>SELECT</u> - <u>Extract data from database.</u>

✓ The data returned is stored in a result table, called the result - set.

<u>SELECT Syntax</u>

```
SELECT Column 1, Column 2, ....
FROM   table_name,
```

Here, column 1, column 2, ... are the field names of the table you want to select data from.

IF you want to select all the fields available in the table use the following syntax :

```
SELECT *FROM  table_name
```

SELECT columns Example

The following SQL statement selects the "Customer Name", "City", and "Country" columns from the "Customers" table:

Example

```
SELECT Customer Name, City,
Country FROM Customers:
```

UPDATE

Updates data in a database.

The UPDATE statement is used to modify the existing records in a table.

UPDATE Syntax

```
UPDATE table_name
SET   Column 1 =   value 1,
      column 2 =   value 2,....

WHERE Condition;
```

Example

```
UPDATE   Customer

SET Contact Name = 'Kanth'
              City = Frank furt
```

• DELETE

Deletes data from a database.

The DELETE statement is used to delete existing records in a table.

DELETE Syntax

```
DELETE FROM table_name
WHERE condition;
```

Example

```
DELETE FROM Customers
WHERE customer
           name = 'Kanth';
```

• INSERT INTO

Inserts new data into a database.

The INSERT INTO statement is used to insert new records in a table.

INSERT INTO Syntax

It is possible to write the INSERT INTO statement in two ways:

① Specify the both the column names and the values to be inserted:

```
INSERT INTO table_name
(column1, column2, column3,.....)
values (value1, value2, value3,......);
```

→

②
```
INSERT INTO table_name
VALUES ( value1, value2, value3,.... );
```

Example

```
INSERT INTO Customers ( customer Name,
Contact Name, Adress, city,
postal code, country )

VALUES ('Kanth', 'Tom B.', 'skagen 21',
'stavanger', '4006', 'Norway') ;
```

● CREATE DATABASE

Creates a new Database.

The CREATE DATABASE statement is used to create a new SQL Database.

Syntax
```
CREATE DATABASE database name;
```

Example
```
CREATE DATABASE text DB;
```

● ALTER TABLE

Modifies a table

The ALTER TABLE statement is used to add, delete, or modify columns in an existing table. And also used to add & drop various constraints on existing table.

ALTER TABLE - ADD column.

world_of_data_and_science

```
ALTER TABLE    table_name
ADD Column_name data type
```

Example

```
ALTER TABLE    Customer
ADD    Email varchar (255 )
```

DROP column

```
ALTER TABLE    table_name
DROP Column   column_name ;
```

Example

```
ALTER TABLE    Customers
DROP COLUMN    Email ;
```

NOTES GALLERY (TELEGRAM)

● **CREATE TABLE**

Creates a new table.

The CREATE TABLE statement is used to create a new table in a database.

Syntax

```
CREATE TABLE table_name (
    Column 1  datatype ,
    column 2  datatype ,
    column 3  datatype ,
    . . . . .
);
```

Scanned with CamScanner

```
CREATE TABLE   Person (
  Person ID    int
  Last Name   varchar ( 255 ),
  First Name   varchar ( 255 ),
  Address  varchar ( 255 ),
  City  varchar ( 255 )
) ;
```

● DROP TABLE

Deletes a table.

The DROP TABLE statement is used to drop an existing table in a data base.

Syntax

```
DROP TABLE  table_ name;
```

Example

```
DROP TABLE  Shippers ;
```

● CREATE INDEX

Creates an Index (Search key)

The CREATE INDEX statement is used to create Indexes in table.

Indexes are used to retrive data from the data base more quickly than otherwise.

The users cannot see the indexes, they are just used to speed up searches / queries.

Syntax

```
CREATE INDEX  Index_name
ON table_name
  ( column 1 , column 2 , ..... );
```

Example

```
CREATE INDEX idx_ lastname
ON person (LastName, FirstName);
```

● **DROP INDEX**

Deletes on Index

The **DROP INDEX** statement is used to delete an index in a table.

```
ALTER TABLE  table_name
DROP INDEX  index_Name;
```