STRUCTURED QUERY LANGUAGE

STRUCTURED QUERY LANGUAGE SQL is a database language used for storing and retrieving data from the database.

SQL was invented by IBM in early 1970's.

SQL supports the following categories of commands to communicate with the database Languages.

DDL,DML,DCL,TCL

Commands

DDL (Data Definition Language)	: CREATE, ALTER, DROP, RENAME, TRUNCATE
DML (Data Manipulation Language)	: INSERT, DELETE, UPDATE
DCL (Data Control Language)	: GRANT, REVOKE
TCL(Transaction Control Language)	: COMMIT, ROLLBACK, SAVEPOINT

Oracle Data types

Char (length) – Fixed length character, ex. char(10)

Varchar2(length) – Variable length character, ex. varchar2(10)

Number – Integer of any,
ex. number(3) – only 3 digits
ex. number(4,1) – float of max 1 decimal place

Date – ex. 01-jan-24

DDL commands

DDL commands are used for table definition. They are used to create, remove and alter the structure of database objects.

CREATE
ALTER
TRUNCATE
DROP
RENAME

CREATE

– Used to create the table

Syntax

```
CREATE TABLE < tablename > (
< column name 1> < datatype>,
< column name 2> < datatype>,
< column name 3> < datatype>
…..
< column name 50> <
datatype>);
```

Ex.

```
CREATE TABLE emp(
emp_id NUMBER(6),
ename VARCHAR2(20),
ph_no VARCHAR2(20),
job_id VARCHAR2(10),
salary NUMBER(8,2)) ;
```

ALTER – Altering the table

Add - Adding new columns

```
ALTER TABLE <tablename> add ( <column name> < datatype>);
```

ex. `ALTER TABLE emp add( dob date);`

Modify - Modify the data type or increase / decrease the column width

```
ALTER TABLE <tablename> modify ( <column name> < newdatatype> );
```

ex. `ALTER TABLE emp modify(job_id varchar2(20));`

ALTER cont..

To Drop a column

```
ALTER TABLE <tablename> drop column < column name>;
```

ex. `alter table emp drop column job_id`

To Rename a column

```
ALTER TABLE <tablename> rename column <oldcolumnname> to <newcolumnname>;
```

ex. `alter table emp rename column dob to dateofbirth;`

Rename – change the table name change the table name

```
Rename <old tablename> to <new tablename >;
```

ex. `rename emp to employee;`

Drop – drop the table definition.

```
Drop Table <table name>;
```

ex. `drop table employee;`

Truncate – Removing the rows not definition

```
Truncate table <table name>;
```

ex. `Truncate table employee;`

Data Manipulation Language DML

DML commands are used to insert, update, retrieval and delete information in the database.

INSERT
UPDATE
DELETE

Insert Command

Inserting values

```
INSERT INTO <tablename> VALUES( val1,val2 …);
```

ex. `insert into emp values(10,'anu','0416-2265767','sales',4000);`

Inserting interactively

```
INSERT INTO <tablename> VALUES( &<column name1> , &<column name2> …);
```

ex. `insert into emp values(&emp_id,'&ename','&ph_no','&job_id',&salary);`

Inserting null values

```
INSERT INTO <tablename> VALUES( val1,' `,' `,val4);
ex. insert into emp values(10,'anu',' ',NULL,4000);

INSERT INTO <tablename> (column name1,column name2) values (val1,val2);
ex. insert into emp(emp_id, ename, salary) values (10,'banu',5000);
```

UPDATE

Simple update

```
UPDATE < tablename > SET <col> = < new value>;
```

ex. `update emp set salary = salary*10;`

Using where clause

```
UPDATE < tablename > SET <col1> = < new value> , <col2> = < new value>
WHERE <conditions>;
```

ex. `update emp set salary = salary*10 where emp_id =10;`

DELETE

Deleting all rows

```
DELETE FROM <tablename>;
```

ex. `delete from emp;`

Deleting specific rows

```
DELETE FROM <tablename> where <condition>;
```

ex. `delete from emp where emp_id=10;`

SELECT

Selecting all the rows from a table

```
SELECT * FROM < tablename>;
```

ex. `select * from emp;`

Selecting specific rows

```
SELECT * FROM < tablename> where <condition>;
```

ex. `select * from emp where salary>5000;`

Selecting specific column

```
SELECT <col1>, <col2> FROM < tablename>;
```

ex. `select emp_id, ename from emp;`

Alias name

```
SELECT <col1> <alias name 1> , <col2> < alias name 2> FROM
<tablename>;
```

ex. `select emp_id “employee no” from emp;`

ex. `select emp_id as employee_id from emp;`

Selecting distinct values for a column

```
SELECT DISTINCT <col2> FROM < tab1>;
```

ex. `Select distinct emp_id from emp;`

Selecting columns satisfying a condition

ex. `SELECT <col1>, <col2> FROM < table name> WHERE <conditions>;`

ex. `select emp_id, ename from emp where salary>5000;`