Datatypes:-

→ Character datatype
→ Number datatype
→ Date/time datatype

Character datatype:-

(i) Char(size)
(ii) Varchar(size)
(iii) Varchar2(size)
(iv) Nchar(size)
(v) Nvarchar2(size)
(vi) Raw(size)
(vii) Long
(viii) Longraw

Number datatype:-

(i) Number
(ii) Number(size)
(iii) Number(P, S)
(iv) Integer
(v) Float
(vi) Decimal

Date datatype/Time datatype:-

(i) DATE
(ii) TIMESTAMP

→ The character and date types data are inserted within single quotes.
→ The number datatypes are inserted directly without single quotes.

NOTE:-
→ SQL quories are case insensitive.

CREATE TABLE COMMAND :-

Syntax:-

```
CREATE TABLE TABLENAME
(col1   datatype(size),
 col2   datatype(size),
  :
  :
 coln   datatype(size));
```

→ Tablename must be a single word, otherwise use underscore.

COMMAND TO VIEW THE TABLE DESCRIPTION:-

Syntax:-

```
Desc tablename;
```

COMMAND TO KNOW THE EXISTING TABLES IN SQL:-

Syntax:-

```
select * from tab;
```

INSERTING DATA INTO TABLE / POPULATING DATA INTO TABLE :-

NOTE:-

The data will be inserted in the order in which the attributes are created in the table.

Syntax:-

```
insert into tablename values(data for col1,
  data for col2, ..., data for coln);
```

ANOTHER WAY TO INSERT DATA IN THE TABLE:-

Syntax:-

```
insert into tablename values(&col1, &col2, ..., &coln);
```

→ To create another row, use one symbol /(slash), then click enter. Automatically, it says you to input the values for another row.

DATA RETRIVAL:-

(i) For retriving whole data from the table:-

Syntax:-

```
select * from tablename;
```

(ii) For retriving selected attributes:-

Syntax:-

```
select col1, col2, ..., colk from tablename;
```

(iii) For retriving selected rows:-

Syntax:-

```
select * from tablename where condition;
```

→ The condition here is of the form:-

attribute op value

or attribute op attribute

→ Here op is one of the operators among =, !=, >, >=, <, <=, LIKE, IN, BETWEEN... AND, AND, IS, NULL, etc.

→ More than one condition can be combined by AND, OR, NOT operators.

(iv) For retriving selected rows and columns:-

Syntax:-

```
select col1, col2, ..., coln
from tablename
where condition;
```

BETWEEN... AND operator:-

In 'Between', a rows of values are selected where upper & lower limit are checked.

IN OPERATOR:-

→ It uses instead of multiple OR operator.

LIKE OPERATOR:-

(i) % :- It matches zero or more characters.

(ii) _(underscore):- It matches any one character.

Assignment :- 1

Create the following tables.

(a) STUDENT (Rollno, Name, dob, course, CGPA, subcode)

(b) SUBJECT (Subcode, subname, credit, LH, PH)

→(a) CREATE TABLE STUDENT
(Rollno Number(5),
Name varchar2(20),
dob date,
Course varchar2(5),
CGPA Number(3,2),
Subcode varchar2(10));

(b) CREATE TABLE SUBJECT
(Subcode varchar2(10),
subname varchar2(15),
credit Number(2),
LH Number(2),
PH Number(2));

Q) Insert atleast 5 records in each table.

(I) (a) insert into student values(101,'Sonali','25-NOV-2000','MCA',8.57,'10A23');
(b) insert into student values(102,'Priya','30-JAN-1999','MBA',9.76,'10B25');
(c) insert into student values(103,'Nitu','17-OCT-1996','MCA',7.38,'10A22');
(d) insert into student values(104,'Gudu','07-DEC-1998','MBA',6.38,'10B26');
(e) insert into student values(105,'Pragyan','23-JUN-1995','MCA',9.00,'10A30');

(II) insert into subject values('10A23','DBMS',4,4,3);
insert into subject values('10B25','DS',4,3,4);
insert into subject values('10A22','Marketing',3,3,2);
insert into subject values('10B26','ESA',2,3,0);
insert into subject values('10A30','OS',4,3,2);

Q→ Retrive the information of all students.

SQL> select * from STUDENT;

Q→ Retrive information about subjects.

SQL> select * from SUBJECT;

Q) Retrive the information of students who are studying MCA.

SQL) select * from student where course='MCA';

Q) Retrive the information of students who are born not earlier than 01-01-1985.

SQL) select * from student
where dob>='01-JAN-1985';

Q) Retrive rollno, name, CGPA from student table.

SQL) select rollno, name, CGPA from student
~~where CGPA>=8~~

Q) Retrive rollno, name, CGPA of all students who secure 8.5 or above CGPA.

SQL) select rollno, name, CGPA
from STUDENT
where CGPA>=8.5;

Q→ Retrive the subject code & subject name along with credit for the subjects having credit <=3.

SQL) select subcode, subname, credit
from SUBJECT
where credit<=3;

Q→ Retrive the information of subjects for which lecture hour is 4 per weak and lab hour is 3 hours per weak.

SQL) select * from SUBJECT
where LH=4 and PH=3;

Q→ Retrive the information of students who belongs to MCA dept and securing CGPA 8.5 or above.

SQL) select *
from STUDENT
where course='MCA' and CGPA>=8.5;

Q→ Retrive the information of students who are not belonging to MBA dept.

SQL) select * from STUDENT
where course!='MBA';

Q→ Retrive the information of subjects where lecture hour is equal to practical hours.

```
SQL> select * from SUBJECT
     where LH=PH;
```

Q→ Retrive rollno, name, dob of students who are not enrolled for MCA and secured CGPA 8.0 or above.

```
SQL> select Rollno, name, dob
     from STUDENT
     where course!='MCA' and CGPA>=8.0;
```

DISTINCT:-

This keyword is used to supress the duplicate values in a column.

Syntax:- select distinct colname
from tablename
where condition;

SORTING:-

Sorting in a table means specific ordering of rows based on a column/columns.

Sort order:-

asc → for ascending order
desc → for descending order

Note:- By default, the sort order is ascending.

Syntax:- select columnlist
from tablename
where condition
order by column/expression asc/desc;

Column alias:- (Alternative name to a column)

Syntax:- select colname "new colname"
from tablename;

or, select colname as newcolname
from tablename;

CONCATENATION:-

→ Concatenation operator is || or ||.

Ex→ Display ename, dno and salary of all employees in the format xxx having dno 1 word having salary 25000 per month.

```
SQL> select ename||'having dno'||dno||'having salary'
     ||sal||'per month' from EMPLOYEE;
```

DML (Data Manipulation Language):-

→ The DML statements are: select, insert, update, delete, merge, call, explain plan, lock table etc..

UPDATE statement:-

→ It is used to update the information in a table.

Syntax:-

```
update tablename
set col1=newvalue, col2=newvalue, ...
coln = new value
where condition;
```

DELETE STATEMENT:- Delete data from table.

Syntax:-

```
delete from tablename
where condition;
```

COMMIT:-

After a DML statement is executed, commit is used to save the changes by the DML statement in the database.

```
SQL> commit; ↵
```

ROLLBACK:-

After a DML statement is executed, the rollback undo the change in the database. i.e. the rollback is taking the database to its previous state after the DML statement is executed

Syntax:- rollback;↵

NOTE:-

After commit, the rollback can't be applied.

→ Commit and rollback statements are only applicable for DML statements, but not for DDL statements.

DDL (Data Definition Language) :-

→ The DDL statements are :- create, alter, drop, tuncate, comment, rename.

(a) DROP STATEMENT :-

Drop table command is used to remove a table from the database permanently and delete the data and description (or structure) permanently from the database.

Syntax :- drop table tablename;

(b) TUNCATE STATEMENT :-

Tuncate is used to delete the data from a table. But the structure remains intact.

Syntax :- tuncate table tablename;

(c) ALTER TABLE STATEMENT :-

→ It is used to update the structure of a table.
→ ALTER table statement can be used to
(i) add columns to a table
(ii) delete columns to a table
(iii) add or delete a constraint
(iv) create or destroy indexes
(v) change the datatype and size of existing columns
(vi) Renaming columns or constraints

Adding a new column to an existing table :-

Syntax :-
```
alter table tablename
add(newcol1 datatype(size),
    newcol2 datatype(size),
    :
    newcoln datatype(size));
```

DATA UPDATE in that column:-

Syntax:- update tablename
set colname=&colname
value value
where colnamey=&colnamey; ↵

→ Here colnamey should be a key attribute.

DROP a column from the table:-

Syntax:- alter table tablename
drop column colname; ↵

RENAME STATEMENT:-

→ For renaming a table:-

Syntax:- rename oldtablename to newtablename;

→ For renaming a column:-

Syntax:-
alter table tablename
rename column oldname to newname;

→ As rename is a DDL statement, it affects the table in the database.

MODIFYING DATATYPE OF A COLUMN:-

Condition:- If data is present in a column, it is not advised to
(i) change the data i.e. from one datatype to another datatype.
(ii) Decrease the size of the column.

Syntax:- alter table tablename
modify (colname newdatatype(size));

→ If new column add in the table, then that data of that column is now empty. So we can change the datatype & also size.

→ If the data is not present in the column, then we can change the datatype or decrease the size of that column.

CREATING A TABLE FROM ANOTHER TABLE/EXISTING TABLE:-

Creating with all columns of existing table:-

Syntax:- create table tablename
as select * from existing_tablename;

NOTE:- The destination table isn't going to ~~with data~~ be populated with data when a condition will be false i.e. to copy structure of a table to a new table.

→ For copying only structure of a table to another/new table;-

Syntax:-

```
create table newtablename
as select * from existing_tablename
where false_condition;
```

→ The false condition may be 1=2; etc..

CREATING/COPYING A TABLE WITH SPECIFIC COLUMN FROM AN EXISTING TABLE:-

Syntax:-

```
create table newtablename(col1, col2, ..., coln)
as select col1, col2, ..., coln
from existing_tablename;
```