UNIT-2

DDL - Data Definition language.
1) create —
2) Alter —
3) Truncate —
4) Rename —

DDL - create
Alter
Truncate
Rename

DML - Insert
update
delete
selelt

1] create :- to create a new database object

2] Alter :- to chang aspects
delect, add, modify

3] DROP :- to remove database object
DROP object <object name>

4] TRUNCATE :- remove all records from table without changing its attributes.

5] Rename :- rename an database object
Rename <old name> to < new name>

DML → Data Manipulation Language
1) Insert
2) Update
3) delete
3) select

1] Insert → insert data in table
Insert values for attributes.

2] Update → to modify column values of existing record.
UPDate STUDENT SET CGPA = 9.0 WHERE CGPA = 8.0

3] Delect :- Used to delect one or more records from table.
Delect FROM student where PRN = '33'.

4] ~~Rename~~ -

4] select :- To retrieve all rows & all columns.
Select PRN, SNAME FROM STUDENT

data type:-

char - A fixed-length char. string
varchar - A variable-length char string
varchar2

int - An integer
number - A fix point numbers with user specified precision
float(n):- A floating point number.
real - } double preciosion floating point No.
double - }
precision - }

date - year/month/day
time - time of day - hour, minute, second format
timestamp - combinate of date & time

constraints:- (7)

Not null - column cannot have null value
Unique - all value in column are different
primary key - not null & unique uniquely identify each row
foreign key - uniqueely identify row in another table
check - all values ~~specified~~ satisfy specific cond?
default - set default value
Index - used to creat & retrieve data from database quickly.

(1) Not Null :- (1) It's not possible to insert null values in column.
(2) it can implemented by create & Alter comn.

(2) Unique:- doesn't allow duplicate value.

(3) primary key:- must contain unique value & not null value
table have only one primary key

(4) Foreign key:- (1) joins two table together
(2) is a field where table refers primary key of another table
(3) Foreign - child table
Candidate - parent table.

Default:- (1) insert default value in column.

Altee :-

① add, delect, modify the column

Add new column

ALTER TABLE table name (Add) column name datatype;

delect

ALTER TABLE table name (DROP column) column name;

Modify Chang the data type.

ALTER TABL table name MODIFY ~~column~~ column name datatype;

ex:- ALTER TABLE STUDENT ADD SNAME NOT NULL snamc;

Give three variations of Insert command with example.

→ ① Insert values for all attributes.

INSERT ~~&~~ INTO table name VALUES (attribut1 value [attribut2 value--])

INSERT ~~&~~ INTO PERSON VALUES (111, 'ABC', 'DD')

② Insert values for some attributes.

INSERT INTO table name (attribut1, attribute2) VALUES (attribut1 value, [attribut2 value--])

③ select data from one table & insert into another table.

① INSERT ~~INTO~~ INTO name of table SELECT attribut nane FROM name of table.

② INSERT INTO name of table (attributes) SELECT attribut name FROM name of table.

SQL joins — cartesian product
Left, right, Full outer join
Inner join
Natural join.

Cartesian product / cross join :-

Cartesian product :-
every row of one table joins every row of another table.

It returns ⇒ all rows from first table combined with all rows from second table.

cross join ⇒ every row of one table matched with every row of another table.

cartesian product & cross join are same.

T1 & T2 are two sets :-

T1 X T2

(No. of record in T1 (explicit)) X (No. of records in T2 (implicit))

It does not check for common attribute.

T1

A	B	C
a_1	b_1	c_1
a_2	b_2	c_2

X	Y
x_1	y_1
x_2	y_2

m x n →

A	B	C	X	Y
a_1	b_1	c_1	x_1	y_1
a_1	b_1	c_1	x_2	y_2
a_2	b_2	c_2	x_1	y_1
a_2	b_2	c_2	x_2	y_2

natural join :-

1. same as inner join
2. retrieve data have same values for same attribute from two tables.
3. take condition implicitly where as inner take explicitly
4. column name in both table must be same.

SELECT FROM NATURAL JOINS

SELECT * FROM FACULTY NATURAL JOINS STUDENT.

Set operations :- union, union all, minus, Intersect.

1) Union all :-
   combine all result of two SELECT statements into one set & retaine all dublicates.

2) Union :-
   1) like OR operation.
   2) combine all results of two SELECT statments into one set & eliminate dublicates.

3) Minus :-
   1) It takes result set is ~~that~~, one select statment and removes rows that are also present in second select statment.
   2) It retrieve rows that are present 1 but not in 2

4) intersect :-
   1) AND operation
   2) it retrives those rows which are present in both two statements.

Aggregation :-

Aggregation is fun that takes collection of values as input & return single value.

MAX, MIN, Average, SUM, COUNT

(1) MIN :-

1) Return smallest value from specified column.
2) column need to be numeric type.
3) Min ignores any null value.
4) SELECT MIN column name/expression From table name;

Select MIN(Balance) as minimumbalance From account;

(2) MAX :-

1) Return MAX value from specified column.
2) column need to be numeric
3) MAX ignores any null value.

(3) Average :-

1) Return average of all values from specified column.
2) SQL AVG() ignores null value
3) column need to be numeric

AVG ([DISTINCT] column name/expression) From table name;

select AVG(balance) as averagebal FROM account;

(4) SUM :-

1) returns sum of numeric column.
2) SQL SUM() ignores null value.

SUM ([DISTINCT] Column name/expression) From tablename;

SELECT SUM(loan) as Total amount from account;

(5) COUNT :-

1) Return the no. of tuples returned by query as number.
2) COUNT(*) = no. of rows

COUNT [DISTINCT] (column name/expression)

SELECT count(*) From customer;

Q.14] explain sql statment – like, in, between

① like, in, between are data retrival ~~words~~ statment in SQL.

1] Like:-

1) It is useful when you want to search rows to match specific pattern, or when you do not know entire value.

2) For this purpose we use '%' character.

ex.

To select student name starting with 'S'.

SELECT First name From STUDENT WHERE FIRST NAM LIKE 'S%.';

2] IN:-

IN operator is used when you want to compare a column with more than one value.

It is similar to OR condition.

For ex.:- You want to find student name who are studying either maths or science.

SELECT first_Name, subject From STUDENT WHERE SUBJECT IN ('MATHS', 'Science');

3] Between:-

It is used to compare data for a range of values.

For ex:- find the name of student bet' age 10 to 15.

SELECT first name, age FROM STUDENT WHERE age BETWEEN 10 AND 15;