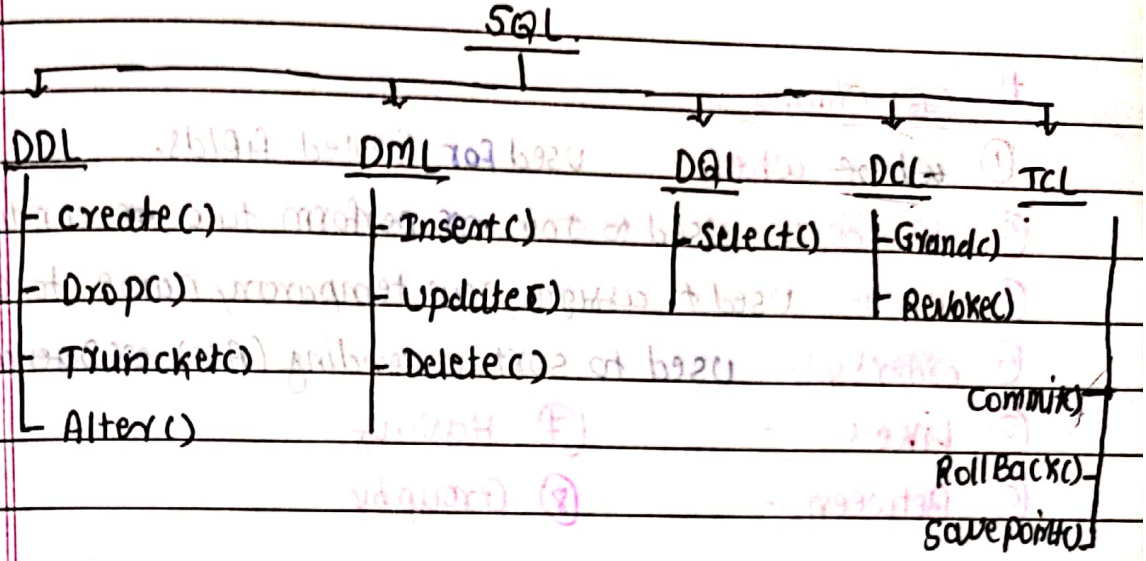
SQL

PAGE No.
DATE / /

Q.1.



① create() :- create table tbl_name( );

② Drop() :- Drop table tbl_name;

③ Truncket :- Truncate table tbl_name;

④ Alter :- Alter table tbl_name add (clm^n name);
Alter table tbl_name modify(clm^n name);

⑤ Insert() :- Insert into tbl_name values( );
Insert into tbl_name (clm^n name) values(clm^n name);

⑥ update() :- Update table_name set address = 'B' where sid = 1;
Update tbl_name set col^m = value where col^m = value;

⑦ Delete() :- Delete from tbl_name where conditions ;

⑧ Select() :- Select "expressions" from tbl_name where conditions;
select * from tbl_name where condition ;

⑨ Grand() :- grand select, update on my_table to some user,
Another user. ;

⑩ Revoke() :- Revoke, select, update on my_table from user1,
User2 ;

⑪ Commit() :- Delete * from tbl_name where condition ;

⑫ Rollback() :- Any quere ; Rollback;

⑬ save point() :- save point savepoint_name;

PAGE No.
DATE / /

SQL Clauses :-

① ~~Whose~~ Where :- Used for limited fields.
② AND / OR :- Used to join or perform two conditions.
③ As :- Used to assign new temparary name to col^mn or tbl.
④ orderby :- Used to sort cussending (Asc) or Desending (Desc) order.
⑤ Like :-
⑥ Between :-
⑦ Having.
⑧ Groupby.

Constants :-

① Not Null :- Insure that col^mn have some value.
② UNIQUE :- Insure that all values are different in specific col^mn.
③ primary key :- A combination of not null or uniquely identify eitch rows in table.
④ Foreign key :- It is a col^mn i.e. used to establish link bet^n 2 tables. A foreign key in one table used to point primary key in another table.
⑤ check :- Insure that the values in col^mn satisfy a specific cond^n.
⑥ Default :- set default value for column if no value is specified.
⑦ Auto Increment :- It is allows a unique no. to be generated autom atically, when new record inserted in table.

Joins :- used to combine two or more tables in database.

5 Types of Join :-

① Inner join ② outer join :- i) Left join ii) Right join.
③ self join ④ cross join

① Inner Join :- Used to written only thows result from tbl that data matches both tables.
② Left Join :- Return all rows from left table & matches record on right table or return null if no record found.

③ right Join :- Return all rows on right side table & match with left side.

④ Cross Join :- used to combine two diff^nt tables we will get Cartation product of set of rows from join tables. when eitch row of 1st table is combine with eitch row of 2nd table. Also known as cartation join or cross Join.

⑤ self Join :- Used to join a table with it self we can perform self join using table alices. That is allow us to use same name of table.

Join Syntax & Queries

① Inner Join :-

select * from tbl1 Inner Join tbl2 ON tbl1.id = tbl2.id;

=> select * from student Inner Join clg ON student.id = clg.id;

② Left Join :-

select * from tbl1 Left Join tb2 ON tbl1.id = tbl2.id.

=> select * from stud Left Join clg. ON stud.id = clg.id;

③ Right Join :-

select * from tbl1 Right Join tbl2 ON tbl1.id = tbl2.id

=> select * from stud Right Join clg ON stud.id = clg.id;

④ Cross Join :-

select * from tbl1 cross Join tbl2 ON tbl1.id = tbl2.id

=> select * from stud cross Join clg ON stud.id = clg.id;

⑤ self Join :-

select * from tbl1 self Join tbl2 ON tbl1.id = tbl2.id

=> select * from stud self Join clg ON stud.id = clg.id;

PAGE No.
DATE / /

SQL fun^{n} :-

① MIN fun^{n} :- MIN() / min() :- used to find min. value.
Syntax :- select MIN (clm_name) from tbl_name;

② MAX fun^{n} :- MAX() / max() :- used to find max. value.
Syntax :- select MAX (clm_name) from tbl_name;

③ Avg fun^{n} :- AVG() / avg() :- used to find Avg of num^{rc} clm.
Syntax :- select AVG (clm_name) from tbl_name;

④ sum fun^{n} :- SUM / sum() :- used to find sum of num^{rc} clm.
Syntax :- select SUM (clm_name) from tbl_name;

⑤ count fun^{n} :- COUNT / count() :-
Syntax :- select COUNT (clm_name) from tbl_name;

View :- ① View is a virtual table based on result-set of an SQL statement. ② A view contains rows & columns, just like a real table.

Syntax :-
```
CREATE VIEW view_name AS
    SELECT clm1, clm2, ----
    FROM tbl_name
    Where condition;
```

```
CREATE VIEW ( V1 ) AS
SELECT * FROM stud where condition;
```

Stored procedure :- A stored procedure is prepared SQL code that we can save, so that code can be reused over & again & again.

Syntax :-
```
CREATE PROCEDURE procedure_name AS
    SQL statements GO;
```

Cursor :- A SQL cursor is a database object that is used to retrive data from result set one row at a time. A SQL cursor is used when the data needs to be updated row by row.

SQL Cursor Lifecycle :-

1. Declaring cursor
2. opening cursor.
3. Featching cursor
4. closing cursor.
5. Deallocating cursor.

Why use Cursor :-

The relational databases, operations are made on a set of rows

Syntax :-

```
DECLARE cursor_name CURSOR [Local / Global] for
Select * from tbl_name order by clm_no;
open cursor_name
close cursor_name.
```

Trigger :- Trigger is a set of actions that are run automatically when a specified change operation is performed on a specified table.

Syntax :-

```
CREATE TRIGGER trigger_name
trigger_time (BEFORE) trigger_event (Insert) ON tbl_name
FOR EACH ROW
trigger_body
trigger_time :{ BEFORE / AFTER }
trigger_event :- { INSERT / UPDATE / DELETE }
```