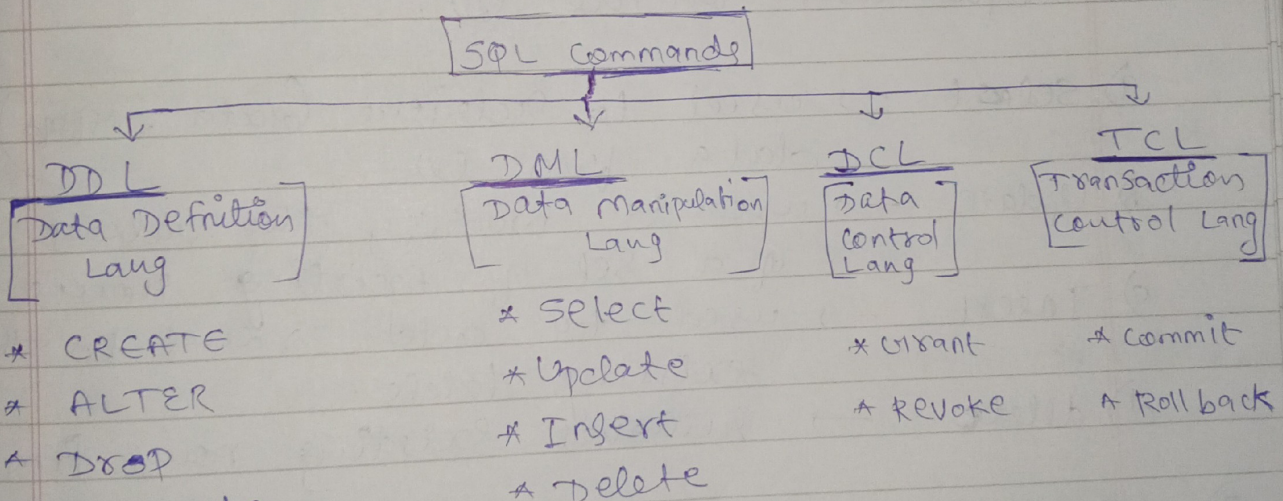
Module 4

(1) → database.

SQL.

* Structured Query Language (SQL)
* wrk with database → RDBMS (type)
* Relational → collection of tables.
* SQL → Q → query ⇒ commands

* SQL =
  - It is a prgming lang for storing & processing info in a relational database.
  - A relational database stores info in a tabular form with rows & colmns representing diffrnt data attributes & various relationship b/s the data values.



1) DDL =
used to create & destroy database objects & database objects.
This cmnts will be used will (1) administrators during setup & removal phases of the (1) project.

a) Create :- create causes an obj to be

[Drop → Entire table dlt.
Truncate → table rows dlted]

Created within a (D)

b) Alter = permits the user to modify an existing obj in various ways.

(entire table drop) c) Drop = causes an existing obj within a (D) to be deleted.

(table rows dlt) d) Truncate = dlts all ~~table~~ data from a table.

2) DML =

used to retrieve ins & modify ~~all~~ database info.

This cmnts will be used by all (D) users during the routine operation on the (D).

a) Select → used to retrieve (data [illegible]) data from (D)

b) Update → used to modify the values of a set of existing rows

c) Insert → used to add rows to the existing table.

d) delete → removes existing rows from table.

3) DCL =

Handles the authorisation aspect of data & permits the user to control who has to access to see / manipulate data within the (D)

a) Grant → Authorize 1 / more user to perform an operation / set of operation on an obj.

b) Revoke → removes / restrict the capability of an user to perform an opration / set of oprations.

4) TCL =

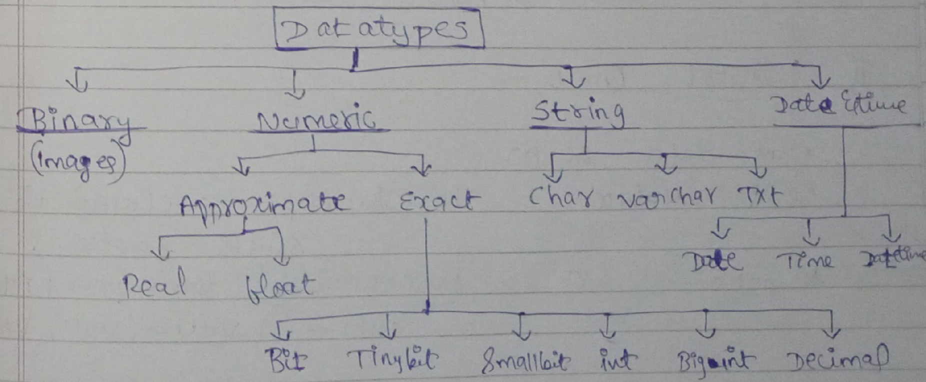
used to handle (D) transactions.
used to keep track of the modifiction that DML statemnts make.

a) Commit = (Change [illegible])
saves the transaction to the (D)

b) Roll back = (Change [illegible])
It will undo the transactions in the (D)

⇒ SQL Datatypes =

* It is an attribute that specifies the type of data of any obj.
* Each colmn, variable & expression has a related datatype in SQL.
* You can use datatypes while creating ur tables.



Stp 1 → (D) create.
Stp 2 → table create
Stp 3 → enter required attribute name & their datatypes.

I CREATE Cmnt =

1) Sytx to create (D) —
SQL > Create database database_name ;

2) Display DB in MySQL —

SQL > Show database;

3) Entering into (D) — (കുറെ (D) ഉണ്ടാകും, അതിൽ ആവശ്യമുള്ള (D) -ലേക്ക് കേറാൻ)
SQL > Use database_name;

4) Creating table —
SQL > create table table_name;

SQL > create table tablename (Att 1 datatype,
Att 2 datatype , ...) .
eg → Create table Studnt (Stid integer,
Sname varchar (20) , percentage float);

5) Display tables —
SQL > Show tables;

II Insert Cmnt =

* Ins data in (D)
* If datatype is str should be enclosed in single quotes.
* Sytx → SQL > INSERT into table_name (Att1,
Att2 ...) values (val1, val2 ...)

III Alter Cmnt =

* DDL Cmnt
* To change the schema / str of Db.
* Add colmn
* Dlt colmn
* modify datatype of existing column.

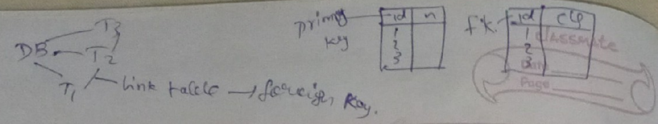
1) Add Colmn =
SQL > ALTER table table_name
add Colmn_name datatype;
eg → SQL > Alter table studnt
add ~~stage~~ age integer;

2) Drop column =
SQL > Alter table table_name
drop column colmn_name;
eg → SQL > Alter table Student
drop column age ;

IV Constraints = Set of rules ~~used for~~ -

1) NOT NULL - values in colmn should not be empty
SQL > CREATE Table tablename (sid
integer NOT NULL,
Student name varchar);

2) Primer key — A table can hve 1 p.key bt can hve many unique constraints
SQL > CREATE Table table_name (Sid integer,
Sname varchar,
PRIMARY KEY (S Id));



3) Foreign key —
used to maintain the integrity of a data.

SQL > CREATE table Employee (D_id integer NOT NULL,
D_name varchar(20), D_salary integer,
PRIMERY key (D_id), FOREIGN KEY (D_id)
REFERENCES Department (D_id));

(last [illegible] table name [illegible] D_id [illegible] foreign key)

4) check Constraint —
SQL > CREATE table Student (S_id int
NOT NULL, S_age int,
CHECK (S_age >= 18));
(for condition purpose)

=> logical operators =

1) AND (0) =
SELECT * from Student
WHERE age = 18 AND place = Cochin;

(capital letters keywords)

(2 attri — T [illegible] we get details) → | Ansar | 18 | Cochin |

AND (0) allows the existing of multiple conditions in an SQL Statement where clause

2) OR (0) =
used to compare multiple condition in an SQL Statements.

eg → Select * from Student;
where place = "Irimbalassery" OR age = 20;

3) BETWEEN (0) =
used to combare a value to an applicable value in the list as per the condi.
eg →
Select * from Student;
where age BETWEEN 25 AND 30;

4) Exist (0) =
used to search for the presence of row in a specified table.
eg →
~~Select * from Student;~~
where Exist (select age from Student where age > 30);

5) IN =
used to filter data based on a set of specified values. returns True if a specified value matches any values in a list.
eg → select * from Student;
where age IN (20, 30);

6) LIKE =
used to search for a specified pattern in a column. It is used with wild card characters '%' & "_".
% → represents 0 or more characters.
_ → represents a single characters.
eg → select * from Student
where Studentname LIKE 'A%';
(A-[illegible] start [illegible] values list [illegible])
'%apple%' → contains str apple anywhere within it
% → used to match any no. of char before & after the str 'apple'

bfre transaction — A = 300
A = A-100
⇒ 200

B = 400
B = B+100
⇒ 500

200+500 = 700 ✓ (after transaction)

300+400 = 700

⇒ Transaction Managmnt =

([illegible] value transfer [illegible])

* collection of operations that forms a single logical unit of work → Transaction (T)
* A (T) is a unit of prgm exicution that access & possibility updates various data items.

T.M (4 operations).

Read | write | Commit | Rollback.

→ ACID property = (loss of data [illegible] 4 (pro))

a) Atomicity =

* Entire (T) takes place at once | doesn't hpn.
* must be atomic, all (T) is completed ~~[illegible]~~ None.

b) Consistency = (Bfre (T) = after (T).)

* prgmr must ensure that whn ~~[illegible]~~ run to completion by itself against a consistance (D) instent, the (T) will leave the (D) in a consistnce state.

c) Isolation = (Should be independent)

* multiple (T) can occur concurrently without leading to the inconsistncy of (D).
* (T) should occur independently.

d) Durability = A → 300 [illegible] 100 [illegible] 200 (data change)

* All updates done by a (T) must be perminent. until you change the value
* (D) will not be erased.

→ States of (T) =