DBMS

1. Differentiate DBMS vs RDBMS :-

- Database management system (DBMS) is a software that is used to define, create and maintain a database and provides controlled access to the data.
- RDBMS is used to store or manage only the data that are in the form of tables.
- MySQL, PostgreSQL, IBM DB2, SQL server, Oracle, Microsoft Access, Amazon Redshift etc. are examples of RDBMS

2. What is Primary key!

- A primary key is a column or a group of columns in a table that uniquely identifies the rows of data in that table
- Pk = Unique + Not Null

3. What is foreign key:

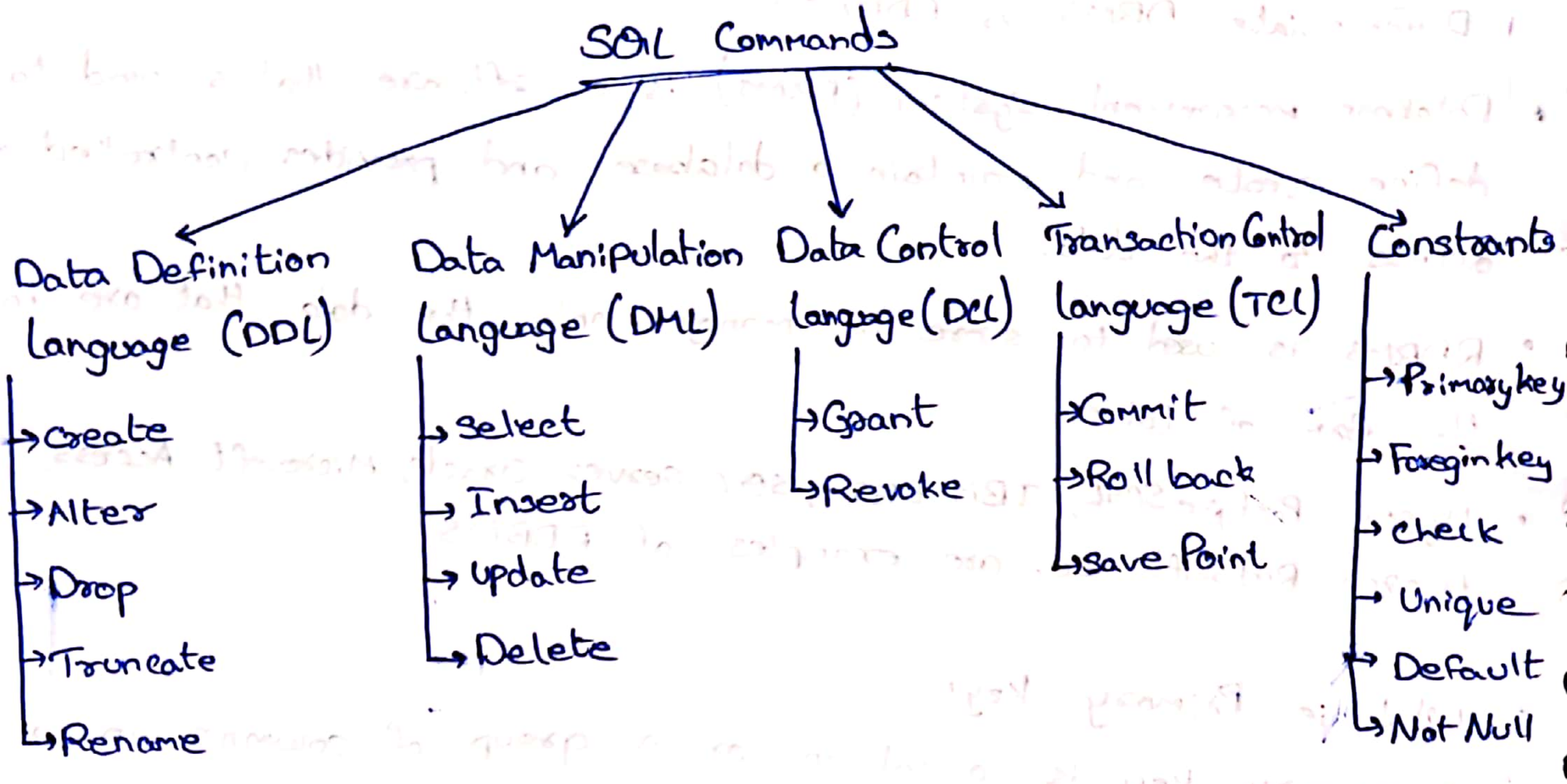
- A FOREIGN KEY is a column (or group of columns) in one table that refers to the PRIMARY KEY in another table
- The table with the foreign key is called as child table, the table with the primary key is called referenced or parent table
- It maintains referential integrity in database

4. What are Constraints & their types:

- SQL constraints are used to specify rules for the data in a table

They are: NOT NULL, UNIQUE, PRIMARY KEY, FOREIGN KEY,
CHECK, DEFAULT
↘ set a default value for column if no value is specified
↓
(ensure that the values in a column satisfies a specific condition)

5. Explain different types of SQL Commands



6. Differentiate Delete, Drop & Truncate

DELETE	DROP	TRUNCATE
Removes rows from a table (at a time)	Removes a table from the database / data dictionary (even removes the table structure from db)	Removes all rows from a table
DML	DDL	DDL
Can be rolled back	Cannot be rolled back	Cannot be rolled back

7. Differentiate group by & order by

- GROUP BY clause is applicable when we want to use aggregate functions to more than one set of rows.
- The ORDER BY Clause is applicable when we want to get the data obtained by a query in the sorting order.
  (just display in order, doesn't change in db)

8. Types of Joins :-

- Cross join : A cross join returns all possible combinations of rows of two tables (also called a Cartesian product
- Inner join : An inner join, also known as a simple join, returns rows from joined tables that have matching rows.
- Left outer join / Left join.
- Right outer join / right join.
- Full outer join.

9. Pattern Matching :-

WHERE Name LIKE 'a%' (here a% means the word starts with a and can have anything after it.

WHERE Name LIKE '_a%' (first letter can be any thing, next a & next can be anything can come)

10. Find 2nd Highest salary :-

- SELECT name, Max(salary) As salary FROM employee
  WHERE Salary <> (SELECT MAX(salary) From employee);
  (not in)

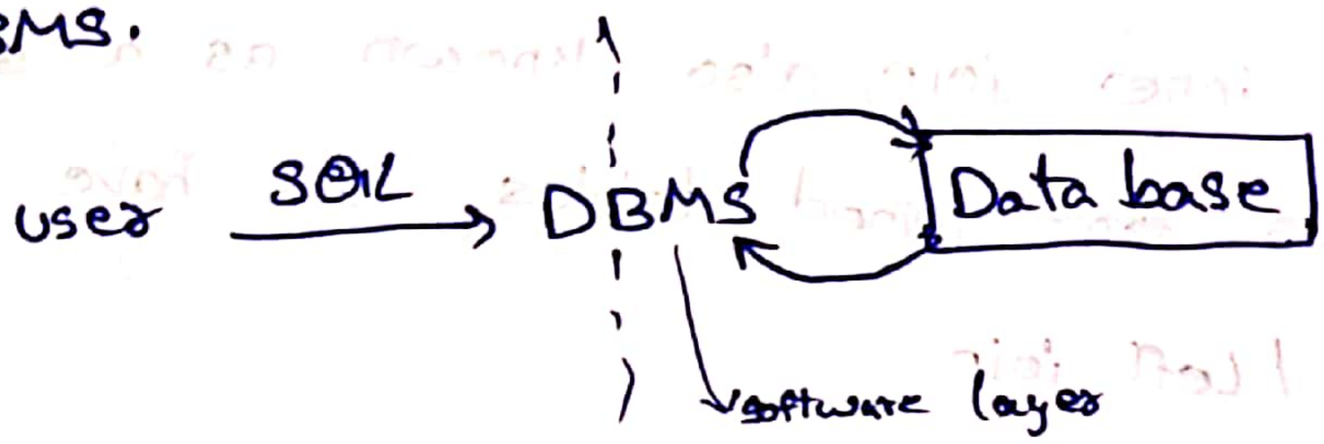
11. ACID Properties :-

- Atomicity : Either all changes are performed, or none of them
- Consistency : Data is in a consistent state when a transaction starts and when it ends.
- Isolation : Transactions (that run concurrently appear to be serialized.
- Durability : After a transaction successfully completes, changes to data Persist and not undone, even in the event of a systemfailure.

SQL

28/08/2023

-> Database is collection of data in a format that can be easily accessed (Digital) & software application used to manage our DB is called DBMS.



-> Types of Database :-

Relational - RDBMS (data stored in tables)
Eg :- MYSQL, PostgreSQL, ORACLE, SQL server, etc...

Non-Relational - NoSQL (data not stored in tables)
Eg :- Mongo DB

-> SQL - (Structured Query language) is a programming language used to interact with RDBMS. (relational dB). & it is used to perform CRUD operations : Create, Read, Update, Delete.

-> In a Table, Columns → structure / schema (design)
rows → individual data.

```
CREATE TABLE table_name(
    column_name1 datatype constraint,
    column_name2 datatype Constraint
);
```

Primary key : it is a column (or set of colums) in a table that uniquely identifies each row. (a unique id). There is only 1 PK & it should Not be Null

Foreign key : is a column (or set of columns) in a table that refers to the primary key in another table. There can be multiple Foreign keys (FK) FKs can have duplicate & null values.

→ WHERE:- Clause

Arithmetic Operators : +(add), -(sub), *(mul), /(div), %(modulus) → remindr

Comparison Operators : = (equal to), != (Not equal to), >, <, >=, <=

logical Operators : AND, OR, NOT, IN, BETWEEN, ALL, LIKE, ANY

Bitwise operators: & (Bitwise AND)
| (Bitwise OR)

IN → matches any value in the list

BETWEEN → select for a given range

→ LIMIT Clause :

SELECT * FROM student LIMIT 3;

sets an upperlimit or numbers of (tuples) rows to be returned.

→ ORDER BY Clause:

To sort in ascending (ASC) or descending (DESC)

→ Aggregate Function :

These perform a calculation on a set of values, and return a single value

- COUNT()
- MAX()
- MIN()
- SUM()
- AVG()

To get max marks:

```
SELECT max(marks)
FROM student;
```

To get AVG marks:

```
SELECT AVG(marks)
FROM student;
```

→ GROUP BY Clause:

-) groups rows that have same values into summary rows.
-) it collect data from multiple records and groups the result by one or more columns.
* generally we use GROUP BY with some aggregation function.

Eg:- Count no/of students in each city :

```
SELECT city, count(name)
FROM student
GROUP BY city;
```

→ having Clause: (applies condition on group by, so it will come after Group by)

Similar to where i.e applies some condition on rows.

used when we want to apply any condition after grouping.

Eg:- Count numbers of students in each city where max marks cross 90.

```
SELECT count(name), city
FROM student
GROUP BY city
HAVING max(marks) >90;
```

General Order of a SQL Query:-

```
SELECT column(s)
FROM table_name
WHERE condition
GROUP BY column(s)
HAVING condition
ORDER BY column(s) ASC;
```

Table related Queries:- update (to update existing rows)

→ UPDATE table_name
SET col1 = val1, col2 = val2
WHERE Condition;

Eg:-
```
UPDATE student
SET grade = "O"
WHERE grade = "A";
```

→ default the safe mode will on in SQL, so to off it:

SET SQL_SAFE_UPDATES = 0; → off (or) 1 = on

→ DELETE (to delete from existing rows)

DELETE FROM table_name
WHERE Condition;

Eg:- DELETE FROM student
WHERE marks<33;

→ Linking tables using foreign key:

FOREIGN KEY (dept_id) REFERENCES dept (id)

Cascading for FK :-

on Delete cascade : when we create a foreign key using this option, it deletes the referencing rows in the child table when the reference row is deleted in the parent table which has a primary key.

on update cascade : When we create a foreign key using UPDATE CASCADE the referencing rows are updated in the child table when the referenced row is updated in the parent table which has a primary key.

eg:-
```
CREATE TABLE student (
    id INT PRIMARY KEY,
    courseID INT,
    FOREIGN KEY (courseID) REFERENCE couse(id)
    ON DELETE Cascade
    ON UPDATE Cascade
);
```

→ ALTER (to change the schema)

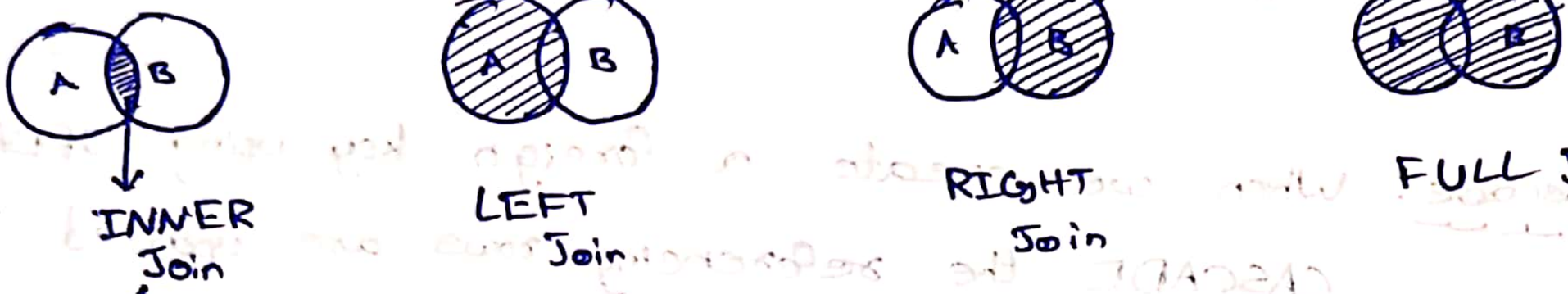
- ADD Column : `ALTER TABLE table_name ADD COLUMN column_name datatype constraint;`
- DROP Column : `ALTER TABLE table_name DROP COLUMN column_name;`
- RENAME table : `ALTER TABLE table_name RENAME TO new_table_name;`

→ TRUNCATE (to delete table's data) where DROP delete's even structure of data

Joins in SQL:

Join is used to combine rows from two or more tables, based on a related column between them.

Types of Joins:-



INNER Join

LEFT Join

RIGHT Join

FULL Join

↓

Returns records that have matching values in both tables

```
SELECT Column(s)
FROM table_A
INNER JOIN table_B
ON tableA.col_name = tableB.col_name;
```

LEFT JOIN :- Returns all the records from the left table, and the matched records from the right table.

Right JOIN :- Returns all the records from the right table, and the matched records from the left table.

FULL Join : Returns all records when there is a match in either left or right table.

↓

Present in postgreSQL & ORACLE but not in MYSQL.

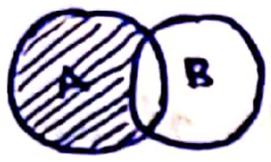
So, in MYSQL :-

```
SELECT * FROM student as a
LEFT JOIN Course as b
ON a.id = b.id
UNION
SELECT * FROM student as a
RIGHT JOIN Course as b
ON a.id = b.id;
```

LEFT JOIN

UNION

RIGHT JOIN



Left Exclusive Join

```
SELECT *
FROM student as a
LEFT JOIN Course as b
ON a.id = b.id
WHERE b.id IS NULL;
```

Right Exclusive Join

```
SELECT *
FROM student as a
RIGHT JOIN Course as b
ON a.id = b.id
WHERE a.id IS NULL;
```

→ SELF JOIN :-
it is a regular join but the table is joined with itself.

```
SELECT Column(s)
FROM table as a
JOIN table as b
ON a.col_name = b.col_name;
```

→ UNION :- it is used to combine the result-set of two or more SELECT statements

To use it -
- every SELECT should have same no. of columns
- columns must have similar data types
- columns in every SELECT should be in same order.

Eg:-
```
SELECT Column(s) FROM tableA
UNION
SELECT Column(s) FROM tableB
```

→ SQL Sub Queries :-

-) A sub query or Inner query or a Nested query is a query within another SQL query.

-) Involves 2 select statements, where it can be written inside: SELECT (or) FROM (or) WHERE ← most common

```
SELECT Column(s)
FROM table_name
WHERE col_name operator (subquery);
```