SQL

```
CREATE  DATABASE  temp1;
drop    DATABASE  temp1;
```

Student

id	name	age
1	AMAN	26

creating our first table

```
USE db_name;
CREATE TABLE table_name(
  column_name1  datatype  constraint,
     ..           2          ..          ..
);
```

```
CREATE TABLE student(
  id INT PRIMARY KEY,
  name VARCHAR(50),
  age INT NOT NULL );
```
(columns)

putting data { INSERT INTO student VALUES (1, "AMAN", 26);

to print { SELECT * FROM student;

Fixed length : CHAR(50)

upto given length : VARCHAR(50)

Signed : +ve and -ve

unsigned : +ve

SQL commands

1) DDL (DATA defination language) : create, alter, rename, truncate & drop
2) DQL (Query) Select
3) DML (Manipulation) insert, update & delete
4) DCL (control) Grant & revoke permission
5) TCL (Transaction control language) Start transaction, commit rollback E.

Keys : 1) Primary key → unique, not null.

2) Foreign Key → First मे रहेगा value normal है पर dusare table मे Primary key है।

↓ can be multiple, null value and duplicate

Constraints : NOT NULL → columns cannot have a null value

Eg: col1 int NOT NULL

Unique → all values in column are different

Eg: col2 int UNIQUE.

PRIMARY KEY → make a column unique & not null but used only for one.

Eg: id int PRIMARY KEY | For declaring : PRIMARY KEY (id)

FOREIGN KEY → Prevent actions that would destroy link b/w tables.

Eg: CREATE TABLE temp(
cust_id int;
FOREIGN KEY (cust_id) references customer (id));

means which column want to make FK.

DEFAULT → set default value in column

fx: salary INT DEFAULT 25000);

[When we declare default then we write: INSERT INTO info VALUES (1, 'ansh', 300);
INSERT INTO info (id, name) values (2, 'Vansh');]

CHECK → It can limit the values allowed in a column

```
CREATE TABLE city(
  id int primary key,
  city varchar(50),
  age int,
  CONSTRAINT age_check CHECK (age >= 18 AND city = 'Delhi')
);
```

Kuch bhi likh sakte hain

SELECT : Select * from table

* → all (OR we can use distinct or any)

where clause
→ To define some conditions

Eg: SELECT * FROM Student WHERE marks > 80;

1) IN, NOT IN

Limit clause Eg: SELECT * FROM student LIMIT 3;

3 student ka data aayega

→ ORDER BY CLAUSE ⟶ ASC / DESC

ORDER BY City ASC;

→ AGGREGATE FUNCTION

→ COUNT()
→ MAX()
→ MIN()
→ SUM()
→ AVG()

Eg:
Select MAX(marks)
FROM student;

→ GROUP BY CLAUSE

SELECT CITY, count(rollno)
FROM student
GROUP BY city;

SEQUENCE

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

→ HAVING CLAUSE

Similar to where i.e applies some condition on rows.
Used when we want to apply any condition after grouping.

→ count number of students in each city where max marks cross 90.

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

→ Table related Queries

→ update: update table_name
set col1 = val1, col2 = val2
where condition;

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

↔ Delete

```
DELETE FROM table_name
WHERE condition;
```

Eg: DELETE FROM student
WHERE marks < 33;

CASCADING FOR FK → एक जगह से update हुआ तो दूसरे जगह automatic changes reflect होगा through FK.

ON UPDATE CASCADE
ON DELETE CASCADE

→ ALTER - change something (ALTER TABLE table_name) (mandatory to run command)

- ADD column (ADD column columnname datatype constraint;)
- DROP column (DROP column columname;)
- RENAME ~~table column~~ table (RENAME TO newtablename;)
- RENAME column (RENAME COLUMN name TO fullname;) (name ↑ old, fullname ↑ new)
- CHANGE (column (rename)
  - ALTER TABLE table_name
    CHANGE COLUMN oldname newname new_datatype newconstraint;
- MODIFY column (modify datatype / constraint)
  - ALTER TABLE table_name
    MODIFY col_name new_datatype new_constraint;

→ TRUNCATE → to delete table's data.

JOINS IN SQL

↓

Used to combine rows from two or more tables based on related column b/w them.