SQL

Date / /
Page No.

Database

Relational (RDBMS)

- Data stored in forms of tables
- MySQL, SQL Server, ORACLE

Non Relational (NoSQL)

- Data not stored in tables
- Mongo DB

We use SQL to work on these DBMS's

What is SQL → Structured Query Language

↓

Prog. lang used to interact with relational databases

Used to Perform CRUD operations

Create, Read, Update, Delete

SEQUEL

Structured English Query Language

↓

SQL Structured Query Language

Date / /
Page No.

Structure of Data Base :

Data Base

Table 1 Data | Table 2 Data

(x → Student

	Roll no.	Name	Class	DOB	Gender	City
Row1 ←	1	Ram	X	1-4-06	M	Agra
R_2 ←	2	Sham	XI	2-5-07	M	Delhi
R_3 ←	3	Gagan	XII	4-1-01	M	Mumbai
R_4 ←	4	Monu	XI	5-1-02	M	Chd.
	↓	↓	↓	↓	↓	↓
	Column 1	C_2	C_3	C_4	C_5	C_6

Columns & Rows

↓

Structure / Schema (design)

↓

individual data

Run first SQL query / Create first DB

Date / /
Page No.

• Create database db_name ;

• Drop database db_name ;

→ CREATE DATABASE db_name ;

→ DROP DATABASE db_name ;

↳ to delete a database

or create database college ;
drop database college ;

// not case sensitive

→ USE college ; (to use a database)
↳ name of database

Creating tables

USE db_name ;

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

define schema or design of table

Student

id	name	age

SQL DATATYPES

Date / /

when we already know the value is +ve

Signed and Unsigned

TINYINT UNSIGNED (0 to 255)

TINYINT (-128 to 127) } range increases

Types of SQL commands

- DDL (Data Definition Language):

  create, alter, rename, truncate, drop

- DQL (Data Query Language): select

- DML (Data Manipulate Language): insert, update & delete

- DCL (Data control Lang): grant & revoke permission to users

- TCL (Transaction Control Lang): start transaction, commit, roll back

- Database related queries

  CREATE DATABASE IF NOT EXISTS db_name;

  DROP DATABASE IF EXIST db_name;

  ↳ less chances of errors in code

- BLOB — String (0 - 65,535) → Store long strings — BLOB(1000)
- INT — Stores integers — INT
- TINYINT — integers (-128 to 127) — TINY INT
- BIG INT — big integers — BIGINT
- BIT — x bit (1 - 64) — BIT(2) → store only 2 bit values

  BIT(1) → 0, 1

  BIT(2) → 00, 10, 01, 11 etc.
- FLOAT — decimal values — FLOAT
- DOUBLE — more precise decimal — DOUBLE
- BOOLEAN — 0, 1 — BOOLEAN
- DATE — YYYY-MM-DD — DATE
- YEAR — YYYY in four digits — YEAR

Constraints -> use to set limits

Date / /
Page No.

ex

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

```
INSERT INTO student VALUES (1, "arnav", 19);
"  "  "  (2, "vikas", 10);
```

output =>

id	name	age
1	arnav	19
2	vikas	10

to run => `SELECT * FROM student;`

SQL Datatypes

Define	Purpose	Usage
• CHAR	-> stores in a fixed length string (0-255)	CHAR(50)
• VARCHAR	string (0-255) -> stores up to given length	VARCHAR(50)

- CHAR -> takes whole memory it will occupy all 50 bytes even though string is small.

To view the data

Select & view all columns

Date / /
Page No.

```
SELECT * FROM table_name ;

SELECT * FROM teachers ;
```

Insert

```
INSERT INTO student VALUES ( 101, "Kunal" )
                           ( 102, "Abhi" ) ;
```

KEYS

- Primary keys
  - It is a columns (or set of columns) in a table that uniquely identify each row.
  - It can't be NULL
  - There is only 1 PK & it should be NOT NULL

ex → serial no. in student table.
- All rows are identified by their serial no.

ex → roll no. can be made a primary key as it is unique for all students

SHOW DATABASES;

SHOW TABLES ;

Date / /
Page No.

To use a database which is created already → USE db_name ;

TABLE Related queries

```
CREATE TABLE table_name (
    column_name1  Datatype constraint
    column_name2  Datatype constraint
);
```

ex →

```
CREATE TABLE teachers (
    name    VARCHAR(20),
    Salary  INT PRIMARY KEY
);
```

To insert rows

```
INSERT INTO teachers VALUES("ram", 500);
INSERT INTO teachers VALUES("sita", 1000);
```

Foreign Key

This key exist in a table but it is primary key for other table.

There can be multiple FK's in a table

Ex:- Student

Roll no.	Marks	City id
PK		FK

City

City id	Name
PK	

FK's can have NULL values

Constraints

Used to specify rules for data in a table

NOT NULL : Columns can not have a NULL value.

Col1 INT NOT NULL

UNIQUE : all values in a column are unique.

Col 2 INT UNIQUE

PRIMARY KEY : makes a column UNIQUE + NOT NULL

Col 3 INT : PRIMARY KEY

Where clause

Date / /
Page No.

SELECT * from students WHERE
marks > 50 AND roll no. > 10 ;

WHERE clause is used to check the conditions

Operators → +, −, /, *
AND, OR, NOT

- BETWEEN :

SELECT * FROM students WHERE
marks BETWEEN 80 AND 90

- IN ⇒ checks condition from a list

SELECT * FROM students WHERE
~~marks~~ city IN ("Delhi", "Mumbai");

- NOT IN (negates the condition)

SELECT * from students WHERE
rollno NOT BETWEEN 10 AND 20;

INSERT INTO employee
values
(default, "ram"), (5000, "sita");

SELECT * FROM employee ;

Date / /
Page No.

- CHECK

```
Create table city (
    id INT PRIMARY KEY,
    city VARCHAR(50),
    age INT,
CONSTRAINT age_check CHECK (age >= 18 AND
                            city = "Delhi")
);
```

Or

```
CREATE TABLE new (
    age INT CHECK(age >= 18)
);
```

Select Command in detail

Basic syntax

```
SELECT col1, col2 FROM table_name;
```

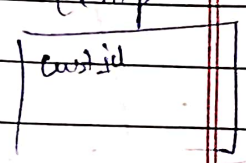
To select all

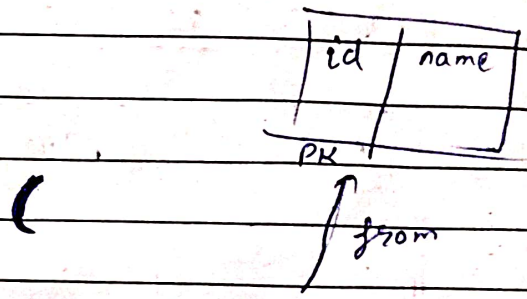
```
SELECT * FROM table_name ;
```

Date / /
Page No.

```
CREATE TABLE temp (
    Name VARCHAR(50) UNIQUE,
    rollno. INT PRIMARY KEY
);
```

Using foreign key






```
CREATE TABLE temp (
    cust_id int,
    FOREIGN KEY (cust_id) references customer(id
);
```

DEFAULT Sets the default value of a column.

```
Salary INT DEFAUTT 25000
```

, if no value is given then defaut value is set

ex →

```
USE college ;
CREATE TABLE employee (
    salary INT Default 25000,
    name VARCHAR(20) NOT NULL
);
```

LIMIT Clause

Date / /
Page No.

Sets an upper limit on number of rows to be returned

SELECT * FROM students LIMIT 3 ;

ORDER BY → ASC, DESC → to sort in ascending or descending order

SELECT * FROM students ORDER BY marks DESC LIMIT 3 ;

→ gives top 3 students with highest marks

~~OR~~

Aggregate functions

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

ex→ • SELECT MAX(marks) FROM students ;
→ gives the maximum marks from data.

• SELECT COUNT(rollno) FROM students ;
→ gives the no. of rows

GROUP BY Clause

Date / /
Page No.

- divides the data into groups having same data

ex → SELECT city, count (roll no)
FROM students
GROUP By city ;

- Makes the groups of diff. city and gives count of students from each city

(it converges the data)

Q) Write a query to find avg. marks in each city in ascending order

Ans ⇒ SELECT ~~FROM~~ city, avg (marks)
FROM students
GROUP By city
ORDER By city;

Q) for a given table find the total payment according to each payment method :

Ans ⇒ SELECT mode, count (customer_id)
or customers
FROM table
GROUP By mode ;

DELETE Command

Date / /
Page No.

- To delete existing rows

DELETE from table_name
WHERE condition ;

ex → Delete all students with "f" grade

⇒ DELETE from students
WHERE grade = "f" ;

ex → Delete the student with name "Rita" and marks = 10 ;

⇒ DELETE from students
where name = "Rita" AND
marks = 10 ;

Foreign Keys

→ interlinks primary key of one table to non primary key of other.

ex → dept

id	name
101	English
102	Hindi
103	Maths

teacher

id	name	dept_id
1	Suman	101
2	Raghav	103
3	Ramesh	102

```
SELECT city
FROM student
WHERE GRADE = "A"
GROUP BY city
HAVING max(marks) > 90
ORDER BY desc;
```

UPDATE clause

Used to change values in existing rows

=>
```
UPDATE table_name
SET col1 = val1, col2 = val2
WHERE condition ;
```

ex → To change marks of student having name "Arnav"

=>
```
UPDATE students
SET marks = 89
WHERE name = "Arnav";
```

ex → To replace "A" grade with "O"

```
UPDATE students
SET grade = "O"
WHERE grade = "A";
```

To remove SAFE mode in My SQL

=> SET SQL_SAFE_UPDATES = 0/1; off on

Q) from a table of students count the number of students getting each Grade (A, B, C, f)

Ans ⇒
```
SELECT grade , count(rollno)
FROM students
~~ORDER~~
GROUP BY Grades ;
```

HAVING CLAUSE — used in groups (Similar to WHERE used in rows)

used to apply conditions after Grouping.

```
SELECT city , count (rollno)
FROM students
GROUP BY city
HAVING max (marks) > 80 ;
```

↳ gives the no. of students from each city having max marks > 80

⇒ General Order of Commands

SELECT (values) → FROM (table name) → WHERE (condition) → GROUP BY (Columns) → HAVING (condition) → ORDER BY (Columns) Asc , desc.

(iv) CHANGE Column

Date / /
Page No.

ALTER TABLE table_name
CHANGE COLUMN old_name new_name new_datatype Constra

(v) MODIFY Column (modify datatype / constraint)

ALTER TABLE
MODIFY col_name new_datatype new_constraint ;

ex → add a column to teacher table
name salary default = 25000 ;

⇒ ALTER TABLE teacher
ADD column salary INT DEFAULT 25000;

now change the column name to teacher_salary

⇒ ALTER TABLE teacher
CHANGE COLUMN salary teacher_salary
INT DEFAULT 25000;

⇒ now change the salary of teacher having dept_id to 34000
103

here, use UPDATE not ALTER

UPDATE teacher
SET teacher_salary = 34000
WHERE dept_id = 103 ;

=> Making changes in Parent table also reflects in child table

Date / /
Page No.

=> Here dept is Parent table and teacher is child table.

ex =>
```
UPDATE dept
SET id = 110
WHERE id = 101 ;
```

↳ this will also change values in teacher table if Cascading is used.

Table related Queries

=> Alter (to change the schema)

(i) ADD column

```
ALTER TABLE table_name
ADD COLUMN column_name datatype constraint;
```

(ii) DROP Column

```
ALTER TABLE table_name
DROP COLUMN column_name;
```

(iii) RENAME TABLE

```
ALTER TABLE table_name
RENAME TO new_table_name ;
```

Date / /
Page No.

How to connect foreign key

```
CREATE TABLE teacher (
    id INT PRIMARY KEY,
    name VARCHAR(20),
    dept_id INT,
    FOREIGN KEY (dept_id)
        REFERENCES dept (id)
);

CREATE TABLE dept (
    id PRIMARY KEY,
    name VARCHAR(20)
);
```

// firstly create this table

CASCADING for FK

↳ When one change in a table also appears in other.

ex) ON UPDATE CASCADE
ON DELETE CASCADE

↳ add these in the table having foreign key

Date / /
Page No.

Q) To change table name to teachers

```
ALTER TABLE teacher
  RENAME TO teachers;
```

TRUNCATE (to delete table's data)

```
TRUNCATE TABLE table_name;
```

It is different from DROP TABLE as it does not delete the table, but the data in it.

Q) In the student table :

(i) Change the name of column "name" to "full_name".

⇒

```
ALTER TABLE student
CHANGE COLUMN name full_name INT;
```

(ii) Delete all students who scored marks less than 80.

```
DELETE FROM student
  WHERE marks < 80;
```

(iii) Delete the column for grades

```
ALTER TABLE student
  DROP Column grades;
```