**MySQL Reference Notes**

**Author: Arindam Das**

**Database & Table Creation**

CREATE DATABASE college;

CREATE TABLE student(

rollno INT PRIMARY KEY,

name VARCHAR(50),

marks INT NOT NULL,

grade VARCHAR(1),

city VARCHAR(20)

);

**Insert Data**

INSERT INTO student (rollno, name, marks, grade, city)

VALUES

(101,'Arindam',68,'B','Kolkata'),

(102,'Sujata',93,'O','Pune'),

(103,'Ankita',80,'A','Kolkata'),

(104,'Subham',76,'B','Delhi'),

(105,'Shahwaiz',92,'O','Kolkata'),

(106,'Chotuu',55,'D','Mumbai');

**SELECT Queries**

SELECT name, marks FROM student;

SELECT city FROM student;

SELECT DISTINCT city FROM student;

**WHERE Clause**

SELECT \* FROM student WHERE marks >= 80;

SELECT \* FROM student WHERE city = 'Kolkata';

SELECT \* FROM student WHERE name = 'Sujata';

SELECT \* FROM student WHERE marks = 55;

**Logical Operators**

-- AND

SELECT \* FROM student WHERE city='Kolkata' AND marks >= 90;

-- OR

SELECT \* FROM student WHERE city='Kolkata' OR marks = 90;

-- BETWEEN

SELECT \* FROM student WHERE marks BETWEEN 80 AND 95;

-- IN

SELECT \* FROM student WHERE city IN ('Kolkata','Delhi');

SELECT \* FROM student WHERE city NOT IN ('Kolkata');

SELECT \* FROM student WHERE name NOT IN ('Sujata');

**LIMIT and ORDER BY**

SELECT \* FROM student LIMIT 3;

SELECT \* FROM student WHERE marks > 75 LIMIT 3;

SELECT \* FROM student ORDER BY city ASC;

SELECT \* FROM student ORDER BY name ASC;

**Aggregate Functions**

SELECT MAX(marks) FROM student;

SELECT COUNT(name) FROM student;

SELECT MIN(marks) FROM student;

SELECT SUM(marks) FROM student;

SELECT AVG(marks) FROM student;

**GROUP BY Clause**

SELECT city, COUNT(rollno) FROM student GROUP BY city;

SELECT grade, COUNT(rollno) FROM student GROUP BY grade ORDER BY grade;

SELECT city, AVG(marks) FROM student GROUP BY city ORDER BY AVG(marks) ASC;

**HAVING Clause**

SELECT city, COUNT(rollno)

FROM student

GROUP BY city

HAVING MAX(marks) > 90;

SELECT city

FROM student

WHERE grade='O'

GROUP BY city

HAVING MAX(marks) >= 92

ORDER BY city ASC;

**UPDATE Queries**

UPDATE student SET grade='O' WHERE grade='A';

UPDATE student SET marks=80 WHERE rollno=101;

UPDATE student SET marks=marks+1;

UPDATE student SET grade='A' WHERE marks BETWEEN 80 AND 85;

UPDATE student SET marks=12 WHERE marks=56;

**DELETE Query**

DELETE FROM student WHERE marks <= 12;

**FOREIGN KEY Example**

CREATE TABLE Dept(

id INT PRIMARY KEY,

name VARCHAR(50)

);

CREATE TABLE Teacher(

id INT PRIMARY KEY,

name VARCHAR(50),

dept\_id INT,

FOREIGN KEY(dept\_id) REFERENCES dept(id)

ON UPDATE CASCADE

ON DELETE CASCADE

);

**ALTER TABLE**

ALTER TABLE student ADD COLUMN age INT NOT NULL DEFAULT 19;

ALTER TABLE student RENAME COLUMN age TO stu\_age;

ALTER TABLE student DROP COLUMN stu\_age;

ALTER TABLE student RENAME TO stu;

**TRUNCATE**

TRUNCATE TABLE student;

**Practice Exercise**

ALTER TABLE stu RENAME COLUMN name TO fullname;

DELETE FROM stu WHERE marks < 80;

ALTER TABLE stu DROP COLUMN grade;

**JOINs**

CREATE TABLE student(student\_id INT, name VARCHAR(50));

CREATE TABLE course(student\_id INT, course VARCHAR(50));

**JOIN Examples:**

-- INNER JOIN

SELECT \* FROM student INNER JOIN course ON student.student\_id = course.student\_id;

-- LEFT JOIN

SELECT \* FROM student LEFT JOIN course ON student.student\_id = course.student\_id;

-- RIGHT JOIN

SELECT \* FROM student RIGHT JOIN course ON student.student\_id = course.student\_id;

-- FULL JOIN (supported in some DBs)

SELECT \* FROM student FULL JOIN course ON student.student\_id = course.student\_id;

-- LEFT EXCLUSIVE JOIN

SELECT \* FROM student LEFT JOIN course ON student.student\_id = course.student\_id WHERE course.student\_id IS NULL;

SELF JOIN

CREATE TABLE employee(

id INT PRIMARY KEY,

name VARCHAR(50),

manager\_id INT

);

SELECT a.name AS manager\_name, b.name

FROM employee a

JOIN employee b ON a.id = b.manager\_id;

**UNION**

SELECT name FROM employee

UNION

SELECT name FROM employee;

-- UNION

SELECT name FROM employee

UNION ALL

SELECT name FROM employee;

**SUBQUERIES**

SELECT name, marks FROM studentnew

WHERE marks > (SELECT AVG(marks) FROM studentnew);

**Even Roll Numbers**

SELECT name, rollno

FROM studentnew

WHERE rollno % 2 = 0;

**Subquery for MAX marks in Delhi**

SELECT MAX(marks)

FROM (SELECT \* FROM studentnew WHERE city='Delhi') AS temp;

**VIEWS**

CREATE VIEW view1 AS

SELECT rollno, name, marks FROM studentnew;

SELECT \* FROM view1 WHERE marks > 90;

DROP VIEW view1;