

NATIONAL INSTITUTE OF TECHNOLOGY KARNATAKA

SURATHKAL, MANGALORE - 575 025

Course Code – CS254

Course Name – Database Systems Lab

Lab - 04

Date – February 23, 2022

Submitted To

Dr. Annappa B

Department of Computer Science and Engineering

National Institute of Technology Karnataka, Surathkal

Submitted By

Md Rakib Hasan

Roll – 201CS132

Department of Computer Science and Engineering

**1. Create a db of lecturers with 10 tuples which consist of first name, last name, age, city, state, pin code, subject, salary and years of experience.**

CREATE DATABASE IF NOT EXISTS lecturers;

USE lecturers;

CREATE TABLE info (

    first\_name VARCHAR(255),

    last\_name VARCHAR(255),

    age INT,

    city VARCHAR(255),

    state VARCHAR(50),

    pin\_code INT,

    subject VARCHAR(50),

    salary INT,

    experience INT);

INSERT INTO info

    VALUES ('Rakib', 'Hasan', 25, 'Dhaka', 'NA', 2000, 'C', 20000, 2),

    ('Tanzimul Ayaan', 'Tanaf', 24, 'Mangalore', 'Karnataka', 575025, 'python', 25000, 3),

    ('Fabiha', 'Smrity', 26, 'Tangail', 'NA', 1950, 'dbms', 30000, 4),

    ('Abdullah', 'Al Mamun', 27, 'Kolkata', 'West Bengal', 202524, 'cpp', 22000, 1),

    ('Abdur', 'Rahim', 23, 'Bangalore', 'Karnataka', 552025, 'java', 19000, 2),

    ('Tanvir', 'Rahman', 28, 'Nagpur', 'Moharasto', 202478, 'javascript', 21000, 4),

    ('Attada', 'Ramprashad', 29, 'AA', 'UP', 275000, 'C', 19000, 2),

    ('Mansur', 'Ali', 35, 'Bangalore', 'Karnataka', 220020, 'Cpp', 20500, 7),

    ('ABC', 'DEF', 20, 'Mangalore', 'Karnataka', 210120, 'C', 23000, 2),

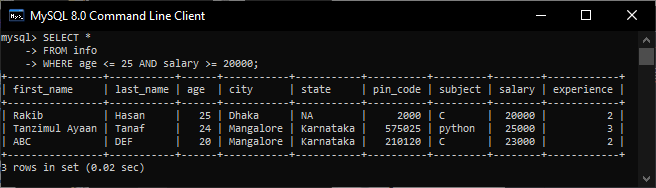
    ('IJK', 'LMN', 36, 'Mangalore', 'Karnataka', 225700, 'C', 27000, 2)

**Write a query to find the salary where age <= 25 and salary >= 20000**

SELECT \*

FROM info

WHERE age <= 25 AND salary >= 20000

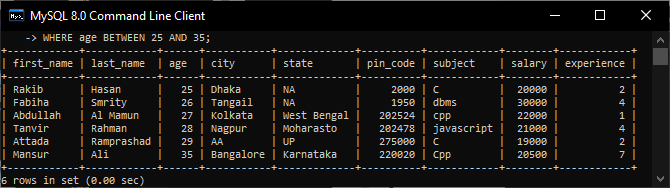
****

**Write a query to print the lecturers between the ages of 25-35**

SELECT \*

FROM info

WHERE age BETWEEN 25 AND 35

****

**Check the experiences of a lecturer, if their experience is greater than 2 years increment their salary by 20%.**

UPDATE info

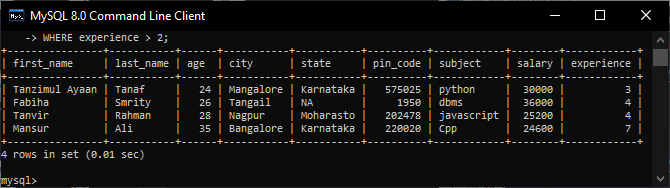
SET salary = salary \* 1.2

WHERE experience > 2;

SELECT \*

FROM info

WHERE experience > 2;

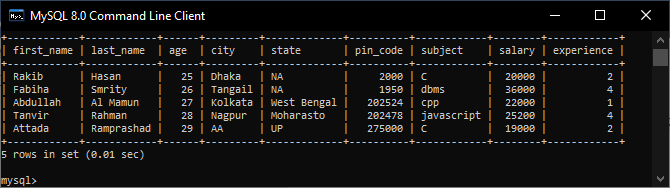
****

**List the names of the lecturers who are not from Karnataka.**

SELECT \*

FROM info

WHERE NOT state = 'Karnataka';

****

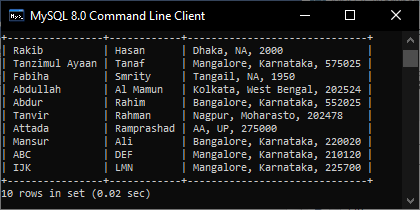
**Create one more column address and print the address combining city, state and pin code.**

ALTER TABLE info

ADD COLUMN address VARCHAR(255);

UPDATE info

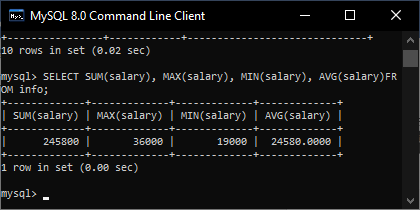
SET address = CONCAT(city, ', ', state, ', ', pin\_code)

****

**Find the sum of salaries of all the lecturers in the table and find out minimum, maximum and average salary.**

SELECT SUM(salary), MAX(salary), MIN(salary), AVG(salary)

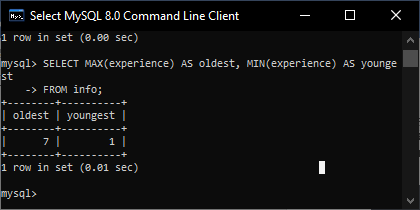
FROM info;

****

**Find out the youngest and oldest lecturer in your table.**

SELECT MAX(experience) AS oldest, MIN(experience) AS youngest

FROM info;

****

**One of the subject ‘C’ was replaced with ‘python’. Write a query to do the same in the table and also print the names of lecturers and their subject after replacement.**

UPDATE info

SET subject = 'Python'

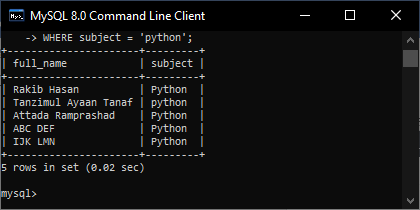
WHERE subject = 'c';

SELECT CONCAT(first\_name, ' ', last\_name) AS full\_name,

    subject

FROM info

WHERE subject = 'python'

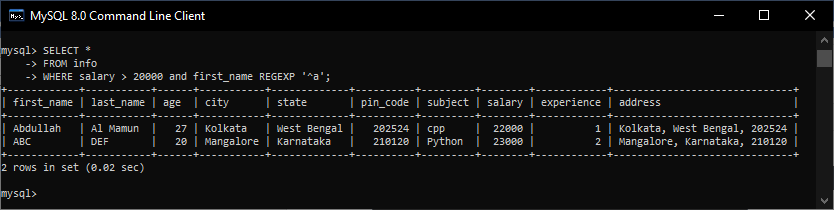
****

**Write a query to retrieve the lecturers whose salary is greater than 20000 and name starts with ‘a’.**

SELECT \*

FROM info

WHERE salary > 20000 and first\_name REGEXP '^a'

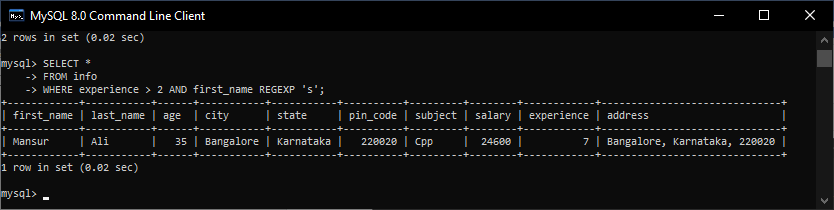
****

**Write a query to retrieve the lecturers whose experience is above 2 years and first name has ‘s’.**

SELECT \*

FROM info

WHERE experience > 2 AND first\_name REGEXP 's'

****

**2. Create a database of movies consisting of movie\_id, movie\_title, actor, actress, year, rating (out of 5), budget, location and director.**

CREATE DATABASE movies;

USE movies;

CREATE TABLE movie\_info (

    movie\_id INT NOT NULL,

    movie\_title VARCHAR(255),

    actor VARCHAR(255),

    actress VARCHAR(255),

    year DATE,

    rating FLOAT,

    budget INT,

    location VARCHAR(255),

    director VARCHAR(255),

    PRIMARY KEY (movie\_id));

INSERT INTO movie\_info

    VALUES

    (101, 'Damayanti', 'Aamir Bashir', 'Fatma Begum', '1995-01-01', 4.9, 100000, 'London', 'Lalita Pawar'),

    (102, 'Kohinoor', 'Aftab Shivdasani', 'Fearless Nadia', '1999-07-01', 4.1, 1050000, 'Kolkata', 'Baap Kamai'),

    (103, 'Baap Kamai', 'Atul Agnihotri', 'Sardar Akhtar', '1995-07-01', 4.3, 10580000, 'Kolkata', 'Madhubala'),

    (104, 'Toofani Tiruni', 'Dulquer Salmaan', 'Sardar Akhtar', '1990-07-01', 4.5, 10480000, 'London', 'Kamini'),

    (105, 'Eye for an eye', 'Fardeen Khan', 'Shobhna Samarth', '1989-07-01', 4.7, 10800000, 'Karnatak', 'Madhubala')

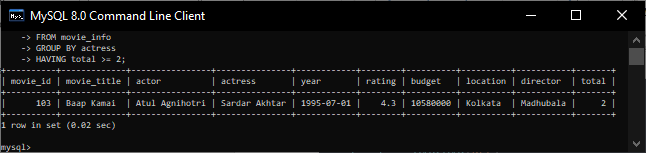
**Write a query to print the movies which have the same actress.**

SELECT \*, count(\*) as total

FROM movie\_info

GROUP BY actress

HAVING total >= 2

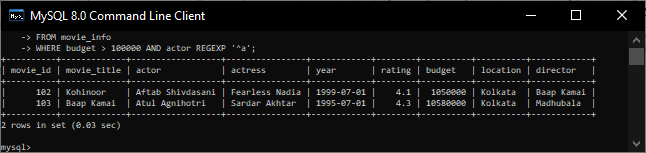
****

**Write a query to print the movies with a budget greater than 100000 and has an actors name starting with A.**

SELECT \*

FROM movie\_info

WHERE budget > 100000 AND actor REGEXP '^a'

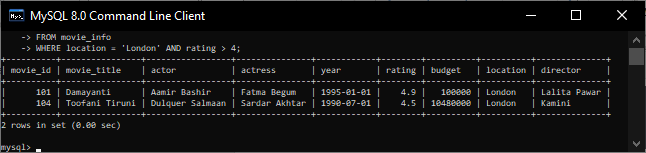
****

**Write a query to filter the movies which were shot in location London and have rating above 4.**

SELECT \*

FROM movie\_info

WHERE location = 'London' AND rating > 4

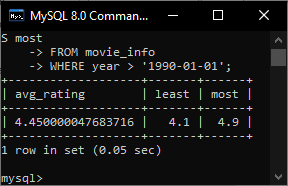
****

**Print the average rating of the movies released after 1990 and find the most and least rated movie.**

SELECT AVG(rating) as avg\_rating, MIN(rating) AS least, MAX(rating) AS most

FROM movie\_info

WHERE year > '1990-01-01'

****

**Update the rating of the movie directed by a particular director with 5 ratings.**

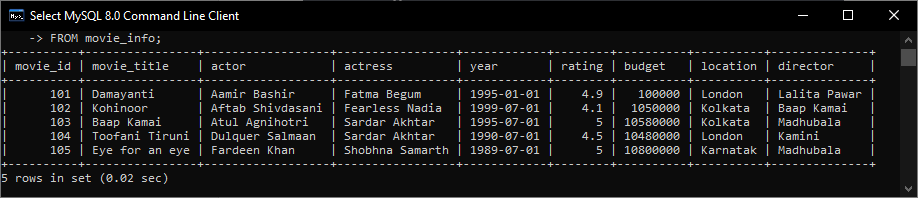
UPDATE movie\_info

SET rating = 5

WHERE director = 'Madhubala';

SELECT \*

FROM movie\_info;

****

**3. Create a student grading database system consisting of:**

**STUDENT(USN, SName, Address, Phone, Gender)**

**IAMARKS(USN, Subcode, Subject name, Test1, Test2, Test3, FinalIA)**

**(Each test is of 10, hence Final IA is of 30)**

CREATE DATABASE student\_grade;

USE student\_grade;

CREATE TABLE student (

    usn INT NOT NULL,

    sname VARCHAR(255),

    address VARCHAR(255),

    phone VARCHAR(20),

    gender VARCHAR(2),

    PRIMARY KEY (usn));

CREATE TABLE iamarks (

    usn INT NOT NULL,

    subcode INT,

    subname VARCHAR(255),

    test1 INT CHECK (test1 <= 10),

    test2 INT CHECK (test2 <= 10),

    test3 INT CHECK (test3 <= 10),

    finalIA INT DEFAULT (test1+test2+test3),

    FOREIGN KEY (usn) REFERENCES student(usn));

INSERT INTO student

    VALUES (101, "Rakib Hasan", "Dhaka", "015215923", "M"),

    (102, "Thamina Akter Liza", "Chandpur", "014012923", "F"),

    (103, "Attada Ramprashad", "UP", "01005", "M"),

    (104, "Tanzimul Ayaan Tanaf", "Rajshahi", "01231661", "M"),

    (105, "Dupur Rahman", "Kolkata", "98752", "F");

INSERT INTO iamarks

    VALUES (101, 51, "C", 9, 7, 10, default),

    (102, 52, "python", 3, 6, 0, default),

    (103, 53, "Cpp", 4, 8, 10, default),

    (104, 54, "java", 10, 5, 4, default),

    (105, 55, "javascript", 10, 8, 6, default);

**Categorize students based on the following criterion and print the table by adding a category column in the student table.**

**If FinalIA = 30 to 20 then CAT = ‘Outstanding’**

**If FinalIA = 20 to 10 then CAT = ‘Average’**

**If FinalIA <10 then CAT = ‘Weak’**

SELECT sname, finalIA,

    CASE

        WHEN finalIA<=30 AND finalIA>20 THEN "Outstanding"

        WHEN finalIA<=20 AND finalIA>10 THEN "Average"

        WHEN finalIA<=10 AND finalIA>=0 THEN "Weak"

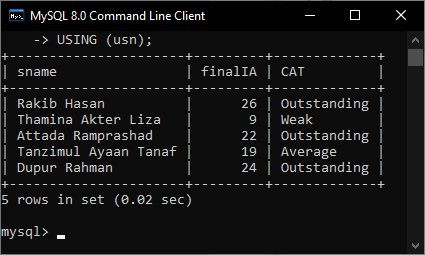
        ELSE "NA"

    END AS CAT

FROM student

JOIN iamarks

    USING (usn)

****