MIS582 Final Course Project

Student Name: _Hardik Maru_____

Date: _____OCT-10-2023_____

Instructions for Students

You will create a database that stores information about ONLY the following entities.

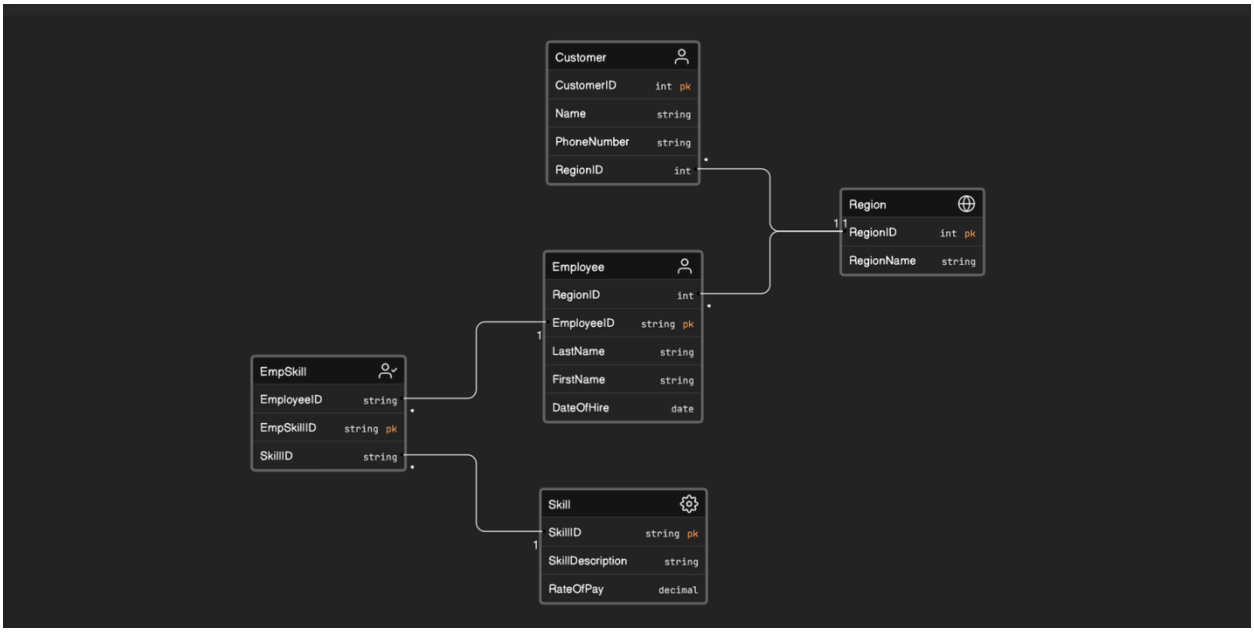
- Customer
- Region
- Employee
- Skill
- EmpSkill

The business rules below give the relationships between these entities.

- A customer is assigned to one region. One region can have several customers.
- An employee can have several skills. One skill can be learned by several employees.
- An employee works for one region. A region can have many employees.

Here are the steps you need to go through to do the project:

1. Given the business rules above, create a logical-level Crow's Foot ERD. ERD can be draw using any online tool such as:
   https://app.diagrams.net/



Please copy and paste your ERD diagram here.

2. original entities and any associative entities (intersection or intermediate tables).

Please copy and paste the script for create table and insert into statements here. Also run a select * statement and copy paste the fully populated tables here.

```
Create database projectDBfinal;

USE ProjectDBfinal;

CREATE TABLE Customer (
    CustomerID INT PRIMARY KEY,
    Name VARCHAR(25),
    PhoneNumber VARCHAR(20),
    RegionID INT,
    FOREIGN KEY (RegionID) REFERENCES Region(RegionID)
);

CREATE TABLE Region (
    RegionID INT PRIMARY KEY,
    RegionName VARCHAR(25)
);

CREATE TABLE Employee (
    EmployeeID VARCHAR(20) PRIMARY KEY,
    LastName VARCHAR(25),
    FirstName VARCHAR(25),
    RegionID INT,
    DateOfHire DATE,
    FOREIGN KEY (RegionID) REFERENCES Region(RegionID)
);
```

```sql
CREATE TABLE Skill (
    SkillID VARCHAR(20) PRIMARY KEY,
    SkillDescription VARCHAR(150),
    RateOfPay DECIMAL(10, 2)
);

CREATE TABLE EmpSkill (
    EmpSkillID VARCHAR(20) PRIMARY KEY,
    EmployeeID VARCHAR(20),
    SkillID varchar(20)

);

ALTER TABLE EmpSkill
ADD CONSTRAINT FK_EmpSkill_Employee
FOREIGN KEY (EmployeeID) REFERENCES Employee(EmployeeID),
ADD CONSTRAINT FK_EmpSkill_Skill
FOREIGN KEY (SkillID) REFERENCES Skill(SkillID);


ALTER TABLE Customer
ADD CONSTRAINT FK_Customer_Region
FOREIGN KEY (RegionID)
REFERENCES Region(RegionID);

INSERT INTO Region (RegionID, RegionName)
VALUES
    (1001, 'Northwest'),
    (1002, 'Southwest'),
    (1003, 'Midwest North'),
    (1004, 'Midwest South'),
    (1005, 'Northeast');

INSERT INTO Customer (CustomerID, Name, PhoneNumber, RegionID)
VALUES
    (1, 'Bellsouth', '222-333-4571', 1003),
    (2, 'Comcast', '253-444-5555', 1003),
    (3, 'Enron', '367-555-6666', 1005),
    (4, 'Exxon', '444-777-7777', 1004);



INSERT INTO Employee (EmployeeID, LastName, FirstName, RegionID, DateOfHire)
VALUES
    ('E1', 'Maru', 'Hardik', 1004, '2019-2-7'),
    ('E2', 'Craig', 'Brett', 1004, '2019-3-30'),
    ('E3', 'Williams', 'Josh', 1005, '1999-3-17'),
    ('E4', 'Cope',  'Leslie', 1002, '2017-4-21'),
    ('E5', 'Mudd', 'Roger', 1001, '2007-10-18');
alter table Employee
    modify EmployeeID VARCHAR(10) null;

INSERT INTO Skill (SkillID, SkillDescription, RateOfPay)
VALUES
    ('s1', 'Data Entry I', 12.00),
    ('S2', 'Java I', 25.00),
    ('S3', 'Python I', 25.00),
```

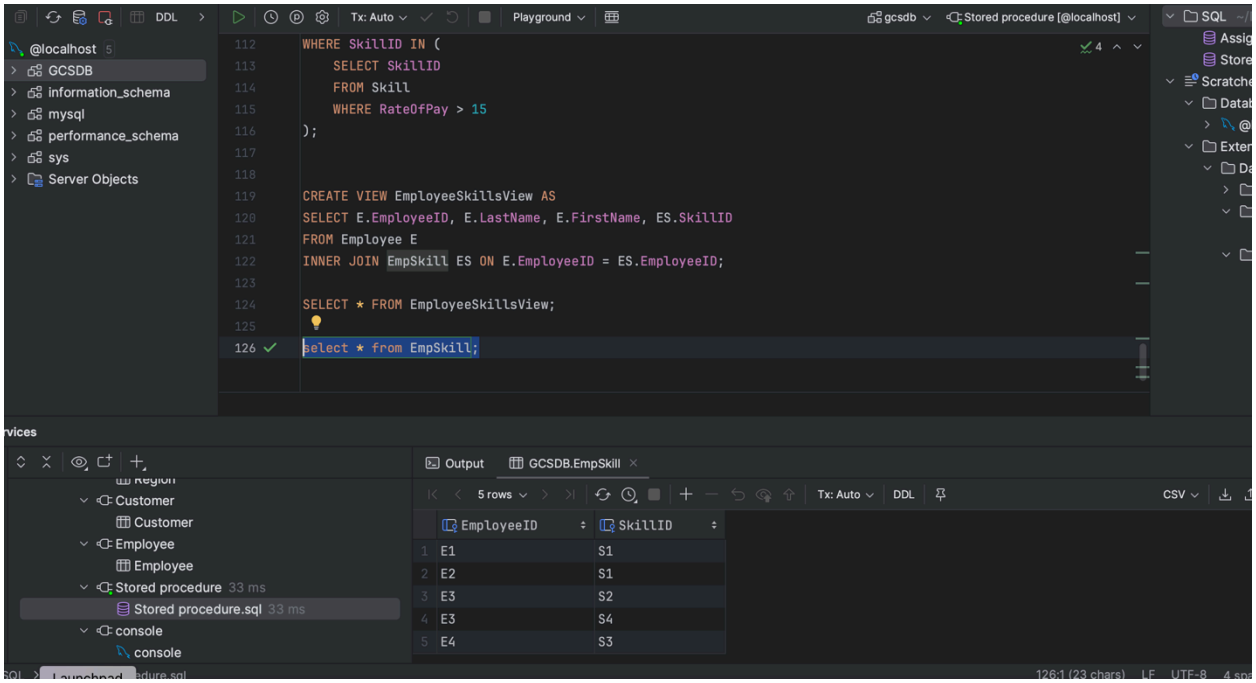
```
    ('S4', 'Python II', 35.00);
INSERT INTO EmpSkill (EmployeeID, SkillID)
VALUES
    ("E1", "s1"),
    ("E2", "s1"),
    ("E3", "s2"),
    ("E3", "s4"),
    ("E4", "S3");

SELECT
    FORMAT(AVG(RateOfPay), 2) AS AverageSkillRate,
    FORMAT(MAX(RateOfPay), 2) AS MaximumSkillRate,
    FORMAT(MIN(RateOfPay), 2) AS MinimumSkillRate
FROM
    Skill;



SELECT Customer.Name
FROM Customer
INNER JOIN Region ON Customer.RegionID = Region.RegionID
WHERE Region.RegionName = 'Northeast';


INSERT INTO EmpSkill (EmployeeID, SkillID)
VALUES
    ("E1", "S1"),
    ("E2", "S1"),
    ("E3", "S2"),
    ("E3", "S4"),
    ("E4", "S3");

select * from EmpSkill;

SELECT DISTINCT EmployeeID
FROM EmpSkill
WHERE SkillID IN (
    SELECT SkillID
    FROM Skill
    WHERE RateOfPay > 15
);




CREATE VIEW EmployeeSkillsView AS
SELECT E.EmployeeID, E.LastName, E.FirstName, ES.SkillID
FROM Employee E
INNER JOIN EmpSkill ES ON E.EmployeeID = ES.EmployeeID;


SELECT * FROM EmployeeSkillsView;
```

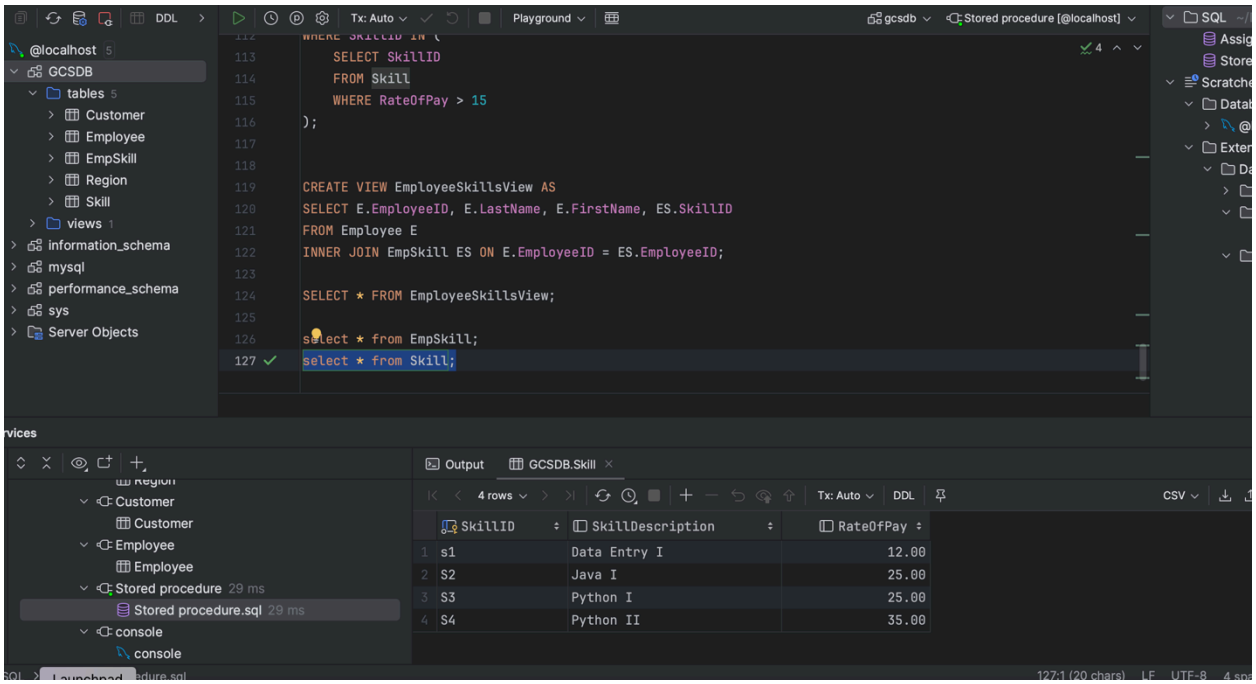
```sql
select * from Customer;


select * from Employee;
```

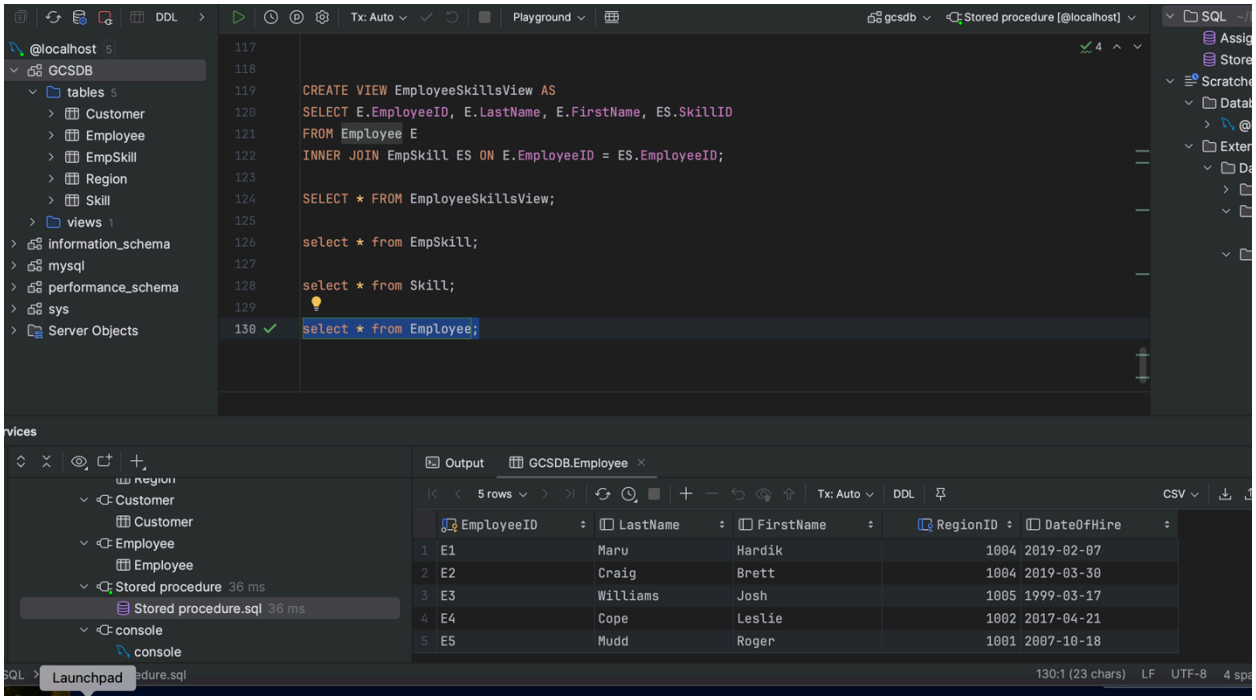
3. Based on the ERD, write and execute a SQL script to create a database and populate it with data based on the tables below. Make sure you clearly identify the primary and foreign keys in your SQL code. Use the attributes shown in the tables below. Data must be entered for all fields except for cusPhone, which can have null values.

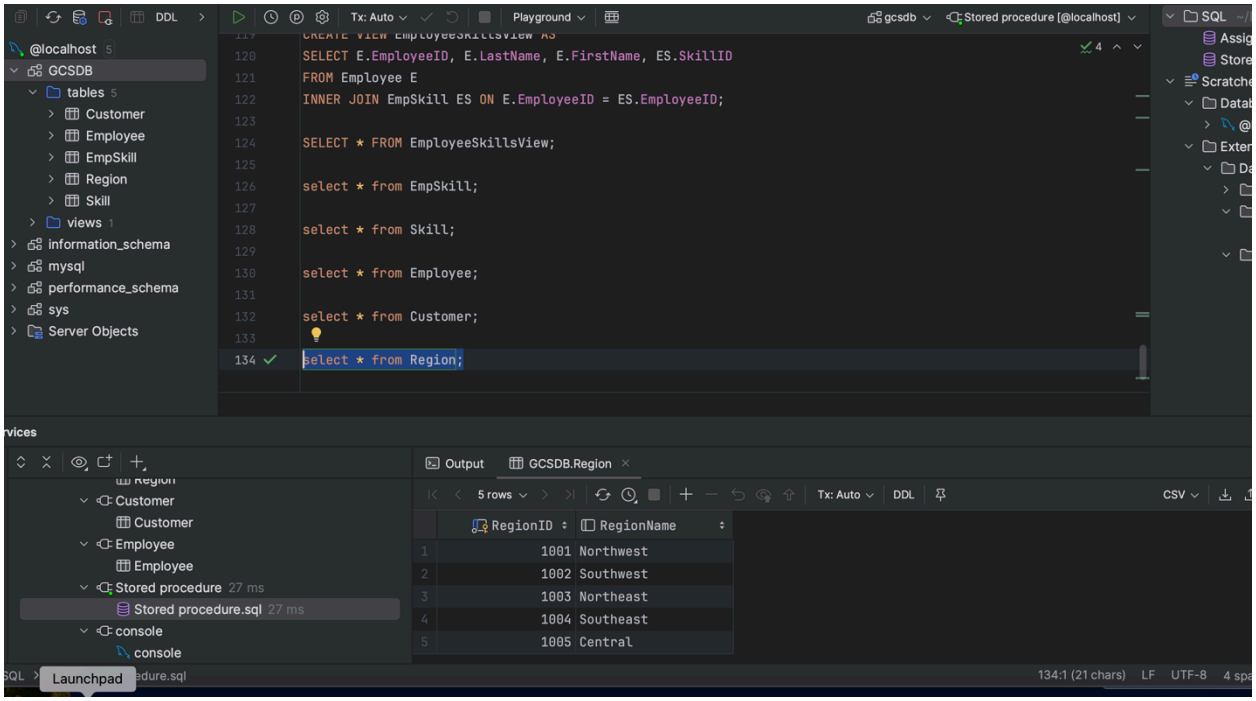
```sql
Create Database projectDBfinal;
CREATE TABLE Customer (
    CustomerID INT PRIMARY KEY,
    Name VARCHAR(25),
    PhoneNumber VARCHAR(20),
    RegionID INT,
    FOREIGN KEY (RegionID) REFERENCES Region(RegionID)
);

CREATE TABLE Region (
    RegionID INT PRIMARY KEY,
    RegionName VARCHAR(25)
);

CREATE TABLE Employee (
    EmployeeID VARCHAR(20) PRIMARY KEY,
    LastName VARCHAR(25),
    FirstName VARCHAR(25),
    RegionID INT,
    DateOfHire DATE,
    FOREIGN KEY (RegionID) REFERENCES Region(RegionID)
);

CREATE TABLE Skill (
    SkillID VARCHAR(20) PRIMARY KEY,
    SkillDescription VARCHAR(150),
    RateOfPay DECIMAL(10, 2)
);

CREATE TABLE EmpSkill (
    EmpSkillID VARCHAR(20) PRIMARY KEY,
    EmployeeID VARCHAR(20),
    SkillID varchar(20)

);

ALTER TABLE EmpSkill
ADD CONSTRAINT FK_EmpSkill_Employee
FOREIGN KEY (EmployeeID) REFERENCES Employee(EmployeeID),
ADD CONSTRAINT FK_EmpSkill_Skill
FOREIGN KEY (SkillID) REFERENCES Skill(SkillID);


ALTER TABLE Customer
ADD CONSTRAINT FK_Customer_Region
```

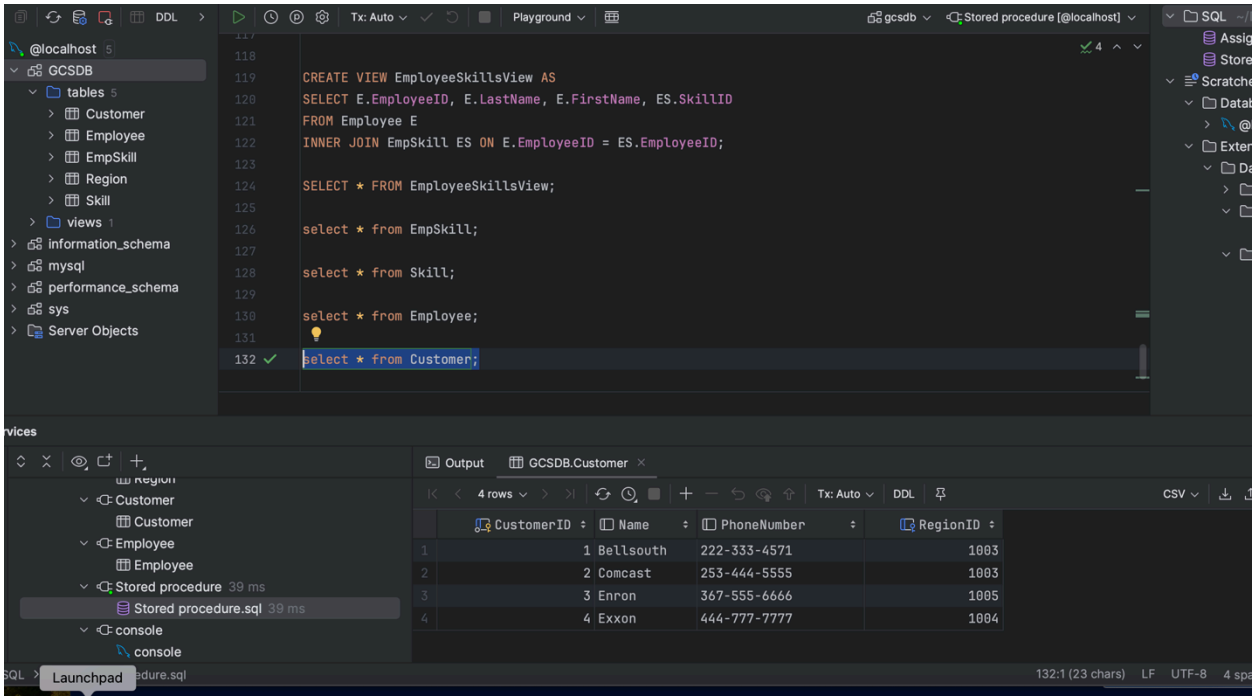
```
FOREIGN KEY (RegionID)
REFERENCES Region(RegionID);
```






DDL
Tx: Auto
Playground
gcsdb
Stored procedure [@localhost]
@localhost 5
GCSDB
tables 5
Customer
Employee
EmpSkill
Region
Skill
views 1
information_schema
mysql
performance_schema
sys
Server Objects
CREATE VIEW EmployeeSkillsView AS
SELECT E.EmployeeID, E.LastName, E.FirstName, ES.SkillID
FROM Employee E
INNER JOIN EmpSkill ES ON E.EmployeeID = ES.EmployeeID;
SELECT * FROM EmployeeSkillsView;
select * from EmpSkill;
select * from Skill;
select * from Employee;
Output
GCSDB.Employee
5 rows
Tx: Auto
DDL
CSV
EmployeeID
LastName
FirstName
RegionID
DateOfHire
1 E1 Maru Hardik 1004 2019-02-07
2 E2 Craig Brett 1004 2019-03-30
3 E3 Williams Josh 1005 1999-03-17
4 E4 Cope Leslie 1002 2017-04-21
5 E5 Mudd Roger 1001 2007-10-18
Stored procedure 36 ms
Stored procedure.sql 36 ms
console
Launchpad
130:1 (23 chars)
LF
UTF-8


DDL
Tx: Auto
Playground
gcsdb
Stored procedure [@localhost]
@localhost 5
GCSDB
tables 5
Customer
Employee
EmpSkill
Region
Skill
views 1
information_schema
mysql
performance_schema
sys
Server Objects
SELECT E.EmployeeID, E.LastName, E.FirstName, ES.SkillID
FROM Employee E
INNER JOIN EmpSkill ES ON E.EmployeeID = ES.EmployeeID;
SELECT * FROM EmployeeSkillsView;
select * from EmpSkill;
select * from Skill;
select * from Employee;
select * from Customer;
select * from Region;
Output
GCSDB.Region
5 rows
Tx: Auto
DDL
CSV
RegionID
RegionName
1 1001 Northwest
2 1002 Southwest
3 1003 Northeast
4 1004 Southeast
5 1005 Central
Stored procedure 27 ms
Stored procedure.sql 27 ms
console
Launchpad
134:1 (21 chars)
LF
UTF-8

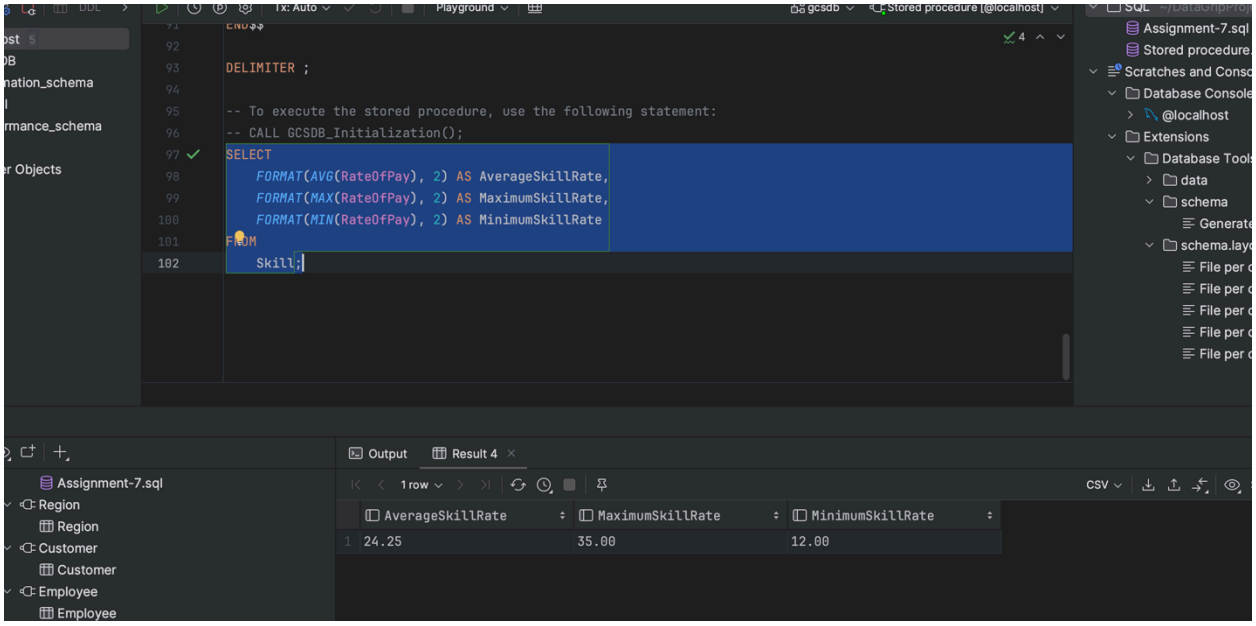

DDL
Tx: Auto
Playground
gcsdb
Stored procedure [@localhost]
@localhost 5
GCSDB
tables 5
Customer
Employee
EmpSkill
Region
Skill
views 1
information_schema
mysql
performance_schema
sys
Server Objects
CREATE VIEW EmployeeSkillsView AS
SELECT E.EmployeeID, E.LastName, E.FirstName, ES.SkillID
FROM Employee E
INNER JOIN EmpSkill ES ON E.EmployeeID = ES.EmployeeID;
SELECT * FROM EmployeeSkillsView;
select * from EmpSkill;
select * from Skill;
select * from Employee;
select * from Customer;
Output
GCSDB.Customer
4 rows
Tx: Auto
DDL
CSV
CustomerID
Name
PhoneNumber
RegionID
1 Bellsouth 222-333-4571 1003
2 Comcast 253-444-5555 1003
3 Enron 367-555-6666 1005
4 Exxon 444-777-7777 1004
Customer
Customer
Employee
Employee
Stored procedure 39 ms
Stored procedure.sql 39 ms
console
console
Launchpad
132:1 (23 chars)
LF
UTF-8

a. Write a query to display average, maximum and minimum skill rate.

The result of the query should be:

24.25	35	12

Copy and paste script as well as result table here.

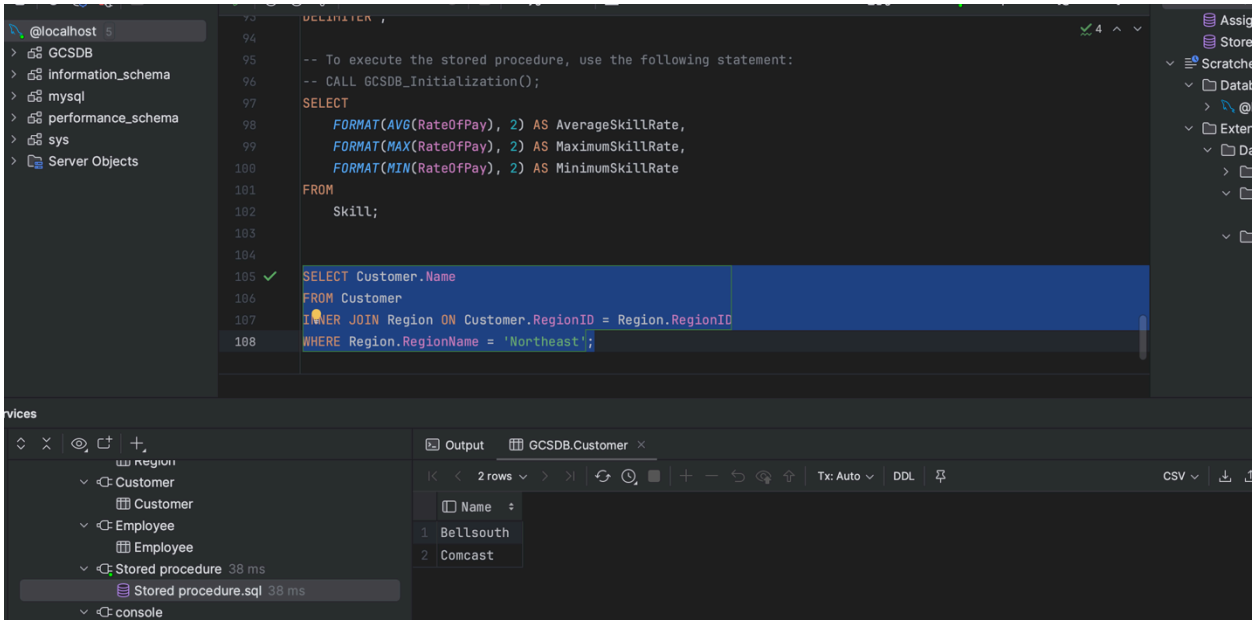


b. Write a query to display the names of all customers in the region named Northeast. You must use a JOIN.

The result of the query should be:

BellSouth
ComCast

Copy and paste script as well as result table here.



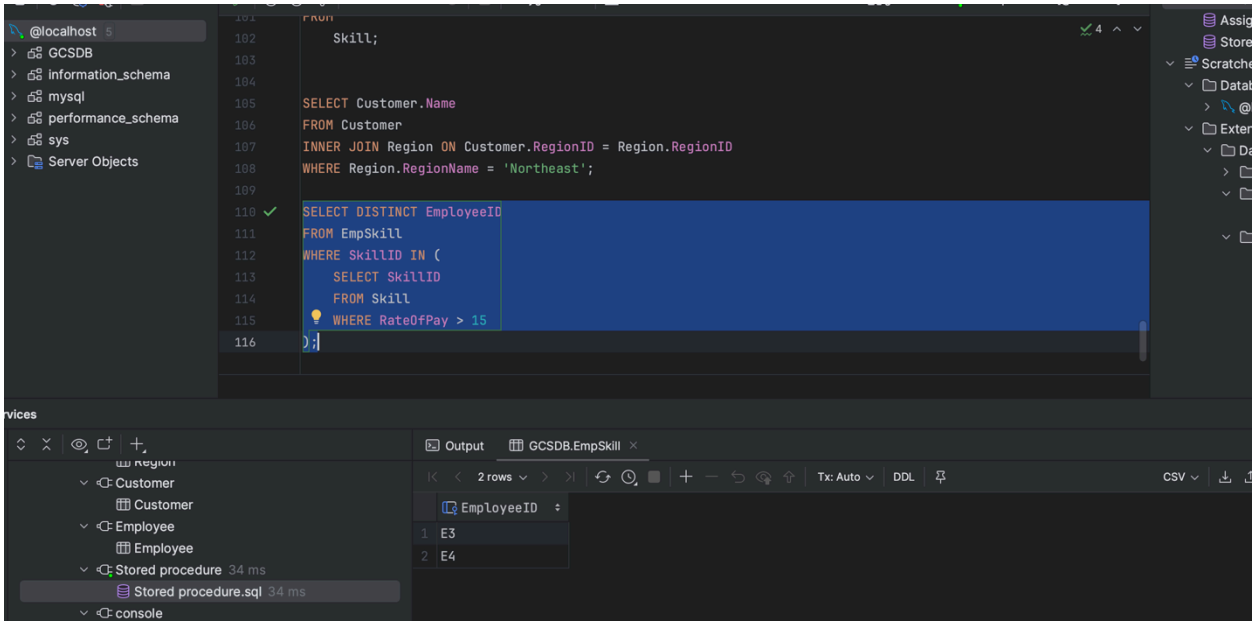
c. Write a query to display employee ID of employees who have skills with that pay more than $15 per hour. You must use a subquery.

The result of the query should be:

E3
E4

HINT: Use the subquery to get a list of skills that pay $15/hour (i.e. SELECT skillID FROM skill WHERE skillRate > 15). Then use main query to find employees whose skill matches one of those in the list.

Copy and paste script as well as result table here.

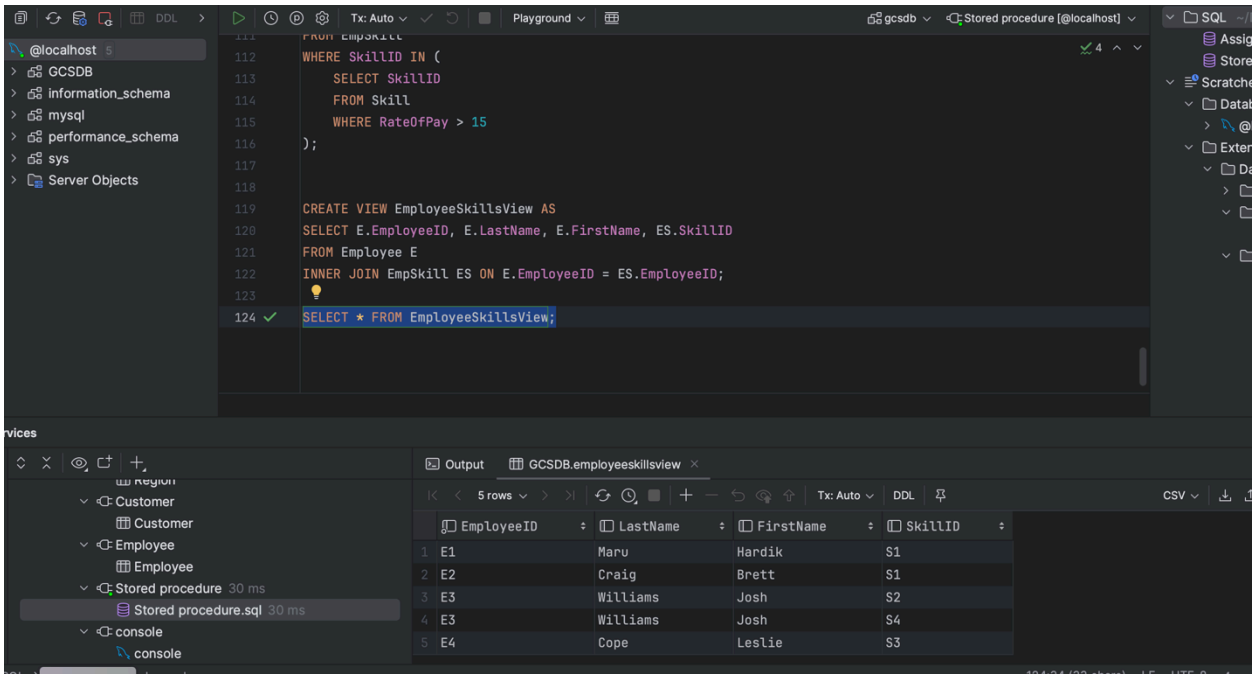


d. Write a query to create view that contains employee id, employee last name, employee first name and skill ID for each employee. After the view is created. Use a SELECT command to display the view.

The result of the SELECT statement should be:

E1	*(your last name)*	*(your first name)*	S1
E2	Craig	Brett	S1
E3	Williams	Josh	S2
E3	Williams	Josh	S4
E4	Cope	Leslie	S3

Copy and paste script as well as result table here.



Here I did additional work, you can just run one stored procedure and entire project would be executed as of requirement given for the project. I would like to shares a complimentary work executed by me.

```
DELIMITER $$
```

```sql
CREATE PROCEDURE DB1_Initialization()
BEGIN
    -- Create tables
    CREATE TABLE IF NOT EXISTS Region (
        RegionID INT PRIMARY KEY,
        RegionName VARCHAR(25)
    );

    CREATE TABLE IF NOT EXISTS Customer (
        CustomerID INT PRIMARY KEY,
        Name VARCHAR(25),
        PhoneNumber VARCHAR(20),
        RegionID INT,
        FOREIGN KEY (RegionID) REFERENCES Region(RegionID)
    );

    CREATE TABLE IF NOT EXISTS Employee (
        EmployeeID VARCHAR(20) PRIMARY KEY,
        LastName VARCHAR(25),
        FirstName VARCHAR(25),
        RegionID INT,
        DateOfHire DATE,
        FOREIGN KEY (RegionID) REFERENCES Region(RegionID)
    );

    CREATE TABLE IF NOT EXISTS Skill (
        SkillID VARCHAR(20) PRIMARY KEY,
        SkillDescription VARCHAR(150),
        RateOfPay DECIMAL(10, 2)
    );

    CREATE TABLE IF NOT EXISTS EmpSkill (
        EmpSkillID VARCHAR(20) PRIMARY KEY,
        EmployeeID VARCHAR(20),
        SkillID VARCHAR(20),
        FOREIGN KEY (EmployeeID) REFERENCES Employee(EmployeeID),
        FOREIGN KEY (SkillID) REFERENCES Skill(SkillID)
    );

    -- Alter table
    ALTER TABLE Employee
    MODIFY EmployeeID VARCHAR(10);

    -- Insert data
    INSERT INTO Region (RegionID, RegionName)
    VALUES
        (1001, 'Northwest'),
        (1002, 'Southwest'),
        (1003, 'Midwest North'),
        (1004, 'Midwest South'),
        (1005, 'Northeast');

    INSERT INTO Customer (CustomerID, Name, PhoneNumber, RegionID)
    VALUES
        (1, 'Bellsouth', '222-333-4571', 1003),
        (2, 'Comcast', '253-444-5555', 1003),
        (3, 'Enron', '367-555-6666', 1005),
```

```
        (4, 'Exxon', '444-777-7777', 1004);

    INSERT INTO Employee (EmployeeID, LastName, FirstName, RegionID,
DateOfHire)
    VALUES
        ('E1', 'Maru', 'Hardik', 1004, '2019-2-7'),
        ('E2', 'Craig', 'Brett', 1004, '2019-3-30'),
        ('E3', 'Williams', 'Josh', 1005, '1999-3-17'),
        ('E4', 'Cope', 'Leslie', 1002, '2017-4-21'),
        ('E5', 'Mudd', 'Roger', 1001, '2007-10-18');

    INSERT INTO Skill (SkillID, SkillDescription, RateOfPay)
    VALUES
        ('s1', 'Data Entry I', 12.00),
        ('s2', 'Java I', 25.00),
        ('s3', 'Python I', 25.00),
        ('s4', 'Python II', 35.00);

    INSERT INTO EmpSkill (EmployeeID, SkillID)
    VALUES
        ('E1', 's1'),
        ('E2', 's1'),
        ('E3', 's2'),
        ('E3', 's4'),
        ('E4', 's3');

    -- Create a view
    CREATE OR REPLACE VIEW EmployeeSkillsView AS
    SELECT E.EmployeeID, E.LastName, E.FirstName, ES.SkillID
    FROM Employee E
    INNER JOIN EmpSkill ES ON E.EmployeeID = ES.EmployeeID;

END$$

DELIMITER ;

-- To execute the stored procedure, use the following statement:
-- CALL DB1_Initialization();
SELECT
    FORMAT(AVG(RateOfPay), 2) AS AverageSkillRate,
    FORMAT(MAX(RateOfPay), 2) AS MaximumSkillRate,
    FORMAT(MIN(RateOfPay), 2) AS MinimumSkillRate
FROM
    Skill;


SELECT Customer.Name
FROM Customer
INNER JOIN Region ON Customer.RegionID = Region.RegionID
WHERE Region.RegionName = 'Northeast';

SELECT DISTINCT EmployeeID
FROM EmpSkill
WHERE SkillID IN (
    SELECT SkillID
    FROM Skill
    WHERE RateOfPay > 15
```

```
);


CREATE VIEW EmployeeSkillsView AS
SELECT E.EmployeeID, E.LastName, E.FirstName, ES.SkillID
FROM Employee E
INNER JOIN EmpSkill ES ON E.EmployeeID = ES.EmployeeID;

SELECT * FROM EmployeeSkillsView;

select * from EmpSkill;

select * from Skill;

select * from Employee;

select * from Customer;

select * from Region;
```

Thank you
Hardik Maru