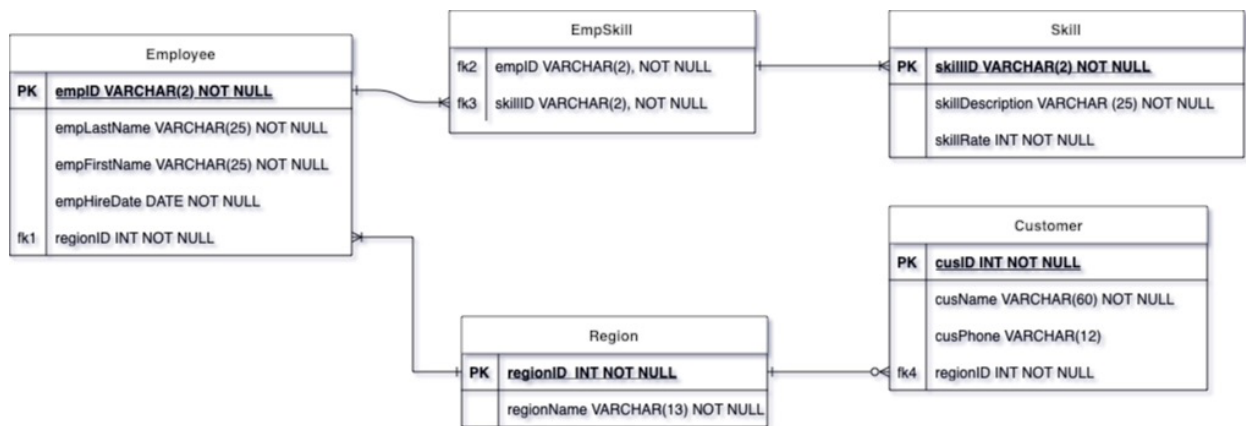


Course Project
DeVry University
College of Engineering and Information Sciences

Course Number: CEIS236
Course Project Deliverable: Final Project
Miguel Mastache
4/22/2023

ERD diagram:



SQL Code to Create Tables;

```
drop table if exists EmpSkill;
drop table if exists Skill;
drop table if exists Employee;
drop table if exists region;
drop table if exists Customer;
```

```
CREATE TABLE IF NOT EXISTS Region (
    regionID INT NOT NULL,
    regionName VARCHAR(13) NOT NULL,
    PRIMARY KEY (regionID)
);
```

```
CREATE TABLE IF NOT EXISTS Employee (
    empID VARCHAR(2) NOT NULL,
    empLastName VARCHAR(25) NOT NULL,
    empFirstName VARCHAR(25) NOT NULL,
    empHireDate DATE NOT NULL,
    regionID INT NOT NULL,
    PRIMARY KEY (empID),
    CONSTRAINT fk1 FOREIGN KEY (regionID)
        REFERENCES region (regionID)
```

```
);
```

```
CREATE TABLE IF NOT EXISTS Skill (  
    skillID VARCHAR(2) NOT NULL,  
    skillDescription VARCHAR(25) NOT NULL,  
    skillRate INT NOT NULL,  
    PRIMARY KEY (skillID)  
);
```

```
CREATE TABLE IF NOT EXISTS EmpSkill (  
    empID VARCHAR(2) NOT NULL,  
    skillID VARCHAR(2) NOT NULL,  
    CONSTRAINT fk2 FOREIGN KEY (empID)  
        REFERENCES employee (empid),  
    CONSTRAINT fk3 FOREIGN KEY (skillID)  
        REFERENCES skill (skillID)  
);
```

```
CREATE TABLE IF NOT EXISTS Customer (  
    cusID INT NOT NULL,  
    cusName VARCHAR(60) NOT NULL,  
    cusPhone VARCHAR(12),  
    regionID INT NOT NULL,  
    PRIMARY KEY (cusID),  
    CONSTRAINT fk4 FOREIGN KEY(regionID)  
        REFERENCES Region(regionID)  
);
```

```
INSERT INTO Region VALUES ('1001', 'Northwest');  
INSERT INTO Region VALUES ('1002', 'Southwest');  
INSERT INTO Region VALUES ('1003', 'Northeast');  
INSERT INTO Region VALUES ('1004', 'Southeast');  
INSERT INTO Region VALUES ('1005', 'Central');
```

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

```
INSERT INTO EMPLOYEE VALUES ('E1', 'Mastache', 'Miguel', '2019-02-  
07', '1004');  
INSERT INTO EMPLOYEE VALUES ('E2', 'Craig', 'Brett', '2019-03-30', '1004');  
INSERT INTO EMPLOYEE VALUES ('E3', 'Williams', 'Josh', '1999-03-17', '1005');  
INSERT INTO EMPLOYEE VALUES ('E4', 'Cope', 'Leslie', '2017-04-21', '1002');  
INSERT INTO EMPLOYEE VALUES ('E5', 'Mudd', 'Roger', '2007-10-18', '1001');
```

```
INSERT INTO SKILL VALUES ('S1', 'Data Entry I', '12');  
INSERT INTO SKILL VALUES ('S2', 'Java I', '25');  
INSERT INTO SKILL VALUES ('S3', 'Python I', '25');  
INSERT INTO SKILL VALUES ('S4', 'Python II', '35');
```

```

INSERT INTO EMP_SKILL VALUES ('E1', 'S1');
INSERT INTO EMP_SKILL VALUES ('E2', 'S1');
INSERT INTO EMP_SKILL VALUES ('E3', 'S2');
INSERT INTO EMP_SKILL VALUES ('E3', 'S4');
INSERT INTO EMP_SKILL VALUES ('E4', 'S3');

```

Code to view Tables:

```

SELECT * FROM REGION;
SELECT * FROM CUSTOMER;
SELECT * FROM EMPLOYEE;
SELECT * FROM SKILL;
SELECT * FROM EMP_SKILL;

```

Select * for Region Table

The screenshot shows the MySQL Workbench interface. The main window displays a SQL query in the editor:

```

83
84 • SELECT * FROM REGION;
85 • SELECT * FROM CUSTOMER;
86 • SELECT * FROM EMPLOYEE;
87 • SELECT * FROM SKILL;
88 • SELECT * FROM EMP_SKILL;
89
90 • SELECT * FROM SKILL;
91 • SELECT
92     AVG(SKILLRATE) AS Average,
93     MAX(SKILLRATE) AS Maximum

```

The 'Result Grid' shows the output of the first query, 'SELECT * FROM REGION;'. The results are as follows:

| regionID | regionName |
|----------|------------|
| 1001 | Northwest |
| 1002 | Southwest |
| 1003 | Northeast |
| 1004 | Southeast |
| 1005 | Central |
| NULL | NULL |

The 'Action Output' pane at the bottom shows the execution details for the first query:

| | Time | Action | Response | Duration / Fetch Time |
|---|----------|------------------------------------|-------------------|-------------------------|
| 1 | 17:19:49 | SELECT * FROM REGION LIMIT 0, 1000 | 5 row(s) returned | 0.00035 sec / 0.0000... |

The status bar at the bottom indicates 'Query Completed'.

Select * for Customer Table

The screenshot shows the MySQL Workbench interface. The left sidebar displays the 'SCHEMAS' tree with 'GCS_dbase' selected, and 'Customer' table highlighted under 'Tables'. The main editor shows a SQL query with line 85 highlighted: `SELECT * FROM CUSTOMER;`. The 'Result Grid' below shows a table with columns: cusID, cusName, cusPhone, regionID. The 'Action Output' at the bottom shows the execution of the query.

Table: Customer

Columns:

- cusID int PK
- cusName varchar(60)
- cusPhone varchar(12)
- regionID int

| cusID | cusName | cusPhone | 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 |
| NULL | NULL | NULL | NULL |

Action Output

| | Time | Action | Response | Duration / Fetch Time |
|-----|----------|--------------------------------------|-------------------|-------------------------|
| ✓ 1 | 17:19:49 | SELECT * FROM REGION LIMIT 0, 1000 | 5 row(s) returned | 0.00035 sec / 0.0000... |
| ✓ 2 | 17:20:06 | SELECT * FROM CUSTOMER LIMIT 0, 1... | 4 row(s) returned | 0.00045 sec / 0.000... |

Query Completed

Select * for Employee Table

The screenshot shows the MySQL Workbench interface. The left sidebar displays the 'SCHEMAS' tree with 'GCS_dbase' expanded to show 'Customer' and 'Employee' tables. The main editor contains a SQL query with line numbers 83-93. The query is:

```
83  
84 • SELECT * FROM REGION;  
85 • SELECT * FROM CUSTOMER;  
86 • SELECT * FROM EMPLOYEE;  
87 • SELECT * FROM SKILL;  
88 • SELECT * FROM EMP_SKILL;  
89  
90 • SELECT * FROM SKILL;  
91 • SELECT  
92     AVG(SKILLRATE) AS Average,  
93     MAX(SKILLRATE) AS Maximum.
```

The 'Result Grid' below the query shows the results of the third query, 'SELECT * FROM EMPLOYEE'. The grid has columns: empID, empLastName, empFirstName, empHireDate, and regionID. The data rows are:

| empID | empLastName | empFirstName | empHireDate | regionID |
|-------|-------------|--------------|-------------|----------|
| E1 | Mastache | Miguel | 2019-02-07 | 1004 |
| E2 | Craig | Brett | 2019-03-30 | 1004 |
| E3 | Williams | Josh | 1999-03-17 | 1005 |
| E4 | Cope | Leslie | 2017-04-21 | 1002 |
| E5 | Mudd | Roger | 2007-10-18 | 1001 |
| NULL | NULL | NULL | NULL | NULL |

The 'Action Output' pane at the bottom shows the execution of the query:

| Time | Action | Response | Duration / Fetch Time |
|------|----------|--------------------------------------|---|
| 3 | 17:20:18 | SELECT * FROM EMPLOYEE LIMIT 0, 1... | 5 row(s) returned 0.00046 sec / 0.000... |

The status bar at the bottom indicates 'Query Completed'.

Select * for Skill Table

The screenshot shows the MySQL Workbench interface. The SQL editor contains the following query:

```
83
84 • SELECT * FROM REGION;
85 • SELECT * FROM CUSTOMER;
86 • SELECT * FROM EMPLOYEE;
87 • SELECT * FROM SKILL;
88 • SELECT * FROM EMP_SKILL;
89
90 • SELECT * FROM SKILL;
91 • SELECT
92     AVG(SKILLRATE) AS Average,
93     MAX(SKILLRATE) AS Maximum;
```

The query results are displayed in the Result Grid, showing a table with the following data:

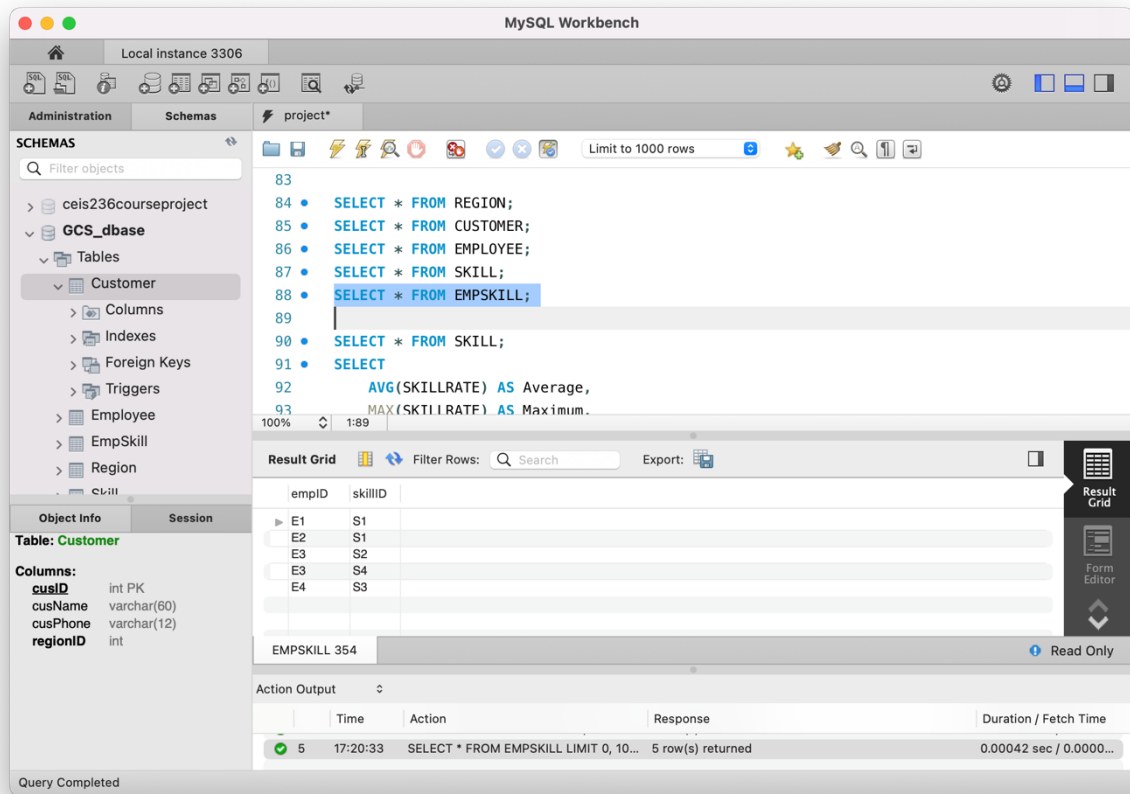
| skillID | skillDescripti... | skillRate |
|---------|-------------------|-----------|
| S1 | Data Entry I | 12 |
| S2 | Java I | 25 |
| S3 | Python I | 25 |
| S4 | Python II | 35 |
| NULL | NULL | NULL |

The Action Output pane shows the following execution details:

| | Time | Action | Response | Duration / Fetch Time |
|---|----------|-----------------------------------|-------------------|------------------------|
| 4 | 17:20:27 | SELECT * FROM SKILL LIMIT 0, 1000 | 4 row(s) returned | 0.00045 sec / 0.000... |

Query Completed

Select * for EmpSkill Table



Code for project Queries:

```
SELECT * FROM SKILL;  
SELECT  
    AVG(SKILLRATE) AS Average,  
    MAX(SKILLRATE) AS Maximum,  
    MIN(SKILLRATE) AS Minimum  
FROM  
    SKILL;
```

```
SELECT  
    cusName  
FROM  
    CUSTOMER  
    JOIN  
    REGION ON REGION.REGIONID = CUSTOMER.REGIONID  
WHERE  
    REGIONNAME = 'Northeast';
```

```
SELECT DISTINCT  
    EMPID  
FROM  
    EMPSKILL
```

```

WHERE
    skillID IN (SELECT
                skillID
                FROM
                SKILL
                WHERE
                skillRate > 15);

CREATE VIEW skilled_employees AS
SELECT
    employee.empID, empLastName, empFirstName, SkillID
FROM
    employee
    JOIN
    empSkill ON employee.empID = empSkill.empID;

SELECT
    *
FROM
    skilled_employees;

```

A query to display average, maximum and minimum skill rate.

The screenshot shows the MySQL Workbench interface. The SQL editor contains the following query:

```

90 • SELECT
91     AVG(SKILLRATE) AS Average,
92     MAX(SKILLRATE) AS Maximum,
93     MIN(SKILLRATE) AS Minimum
94 FROM
95     SKILL;
96
97 • SELECT
98     cusName
99 FROM
100    Customer;

```

The Results window displays the output of the first query:

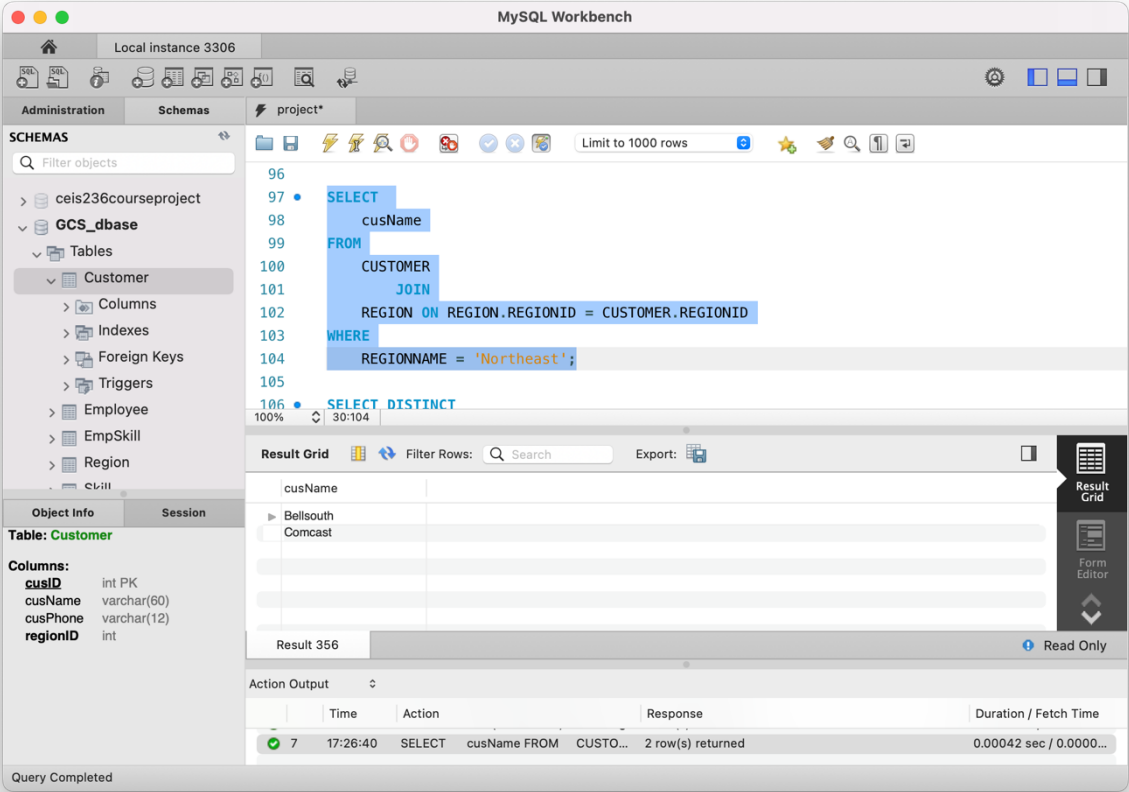
| Average | Maximum | Minimum |
|---------|---------|---------|
| 24.2500 | 35 | 12 |

The Action Output window shows the execution details:

| Time | Action | Response | Duration / Fetch Time |
|------|----------|--|-------------------------|
| 6 | 17:25:39 | SELECT AVG(SKILLRATE) AS Averag... 1 row(s) returned | 0.0011 sec / 0.00001... |

The interface also shows the Schemas pane on the left with the 'Customer' table selected, and the Object Info pane showing its columns: cusID (int PK), cusName (varchar(60)), cusPhone (varchar(12)), and regionID (int).

A query to display the names of all customers in the region named northeast using a join.



A query to display employee id of employees who have skills with that pay more than \$15 per hour using a subquery.

The screenshot shows the MySQL Workbench interface. The SQL editor contains the following query:

```
106 SELECT DISTINCT
107 EMPID
108 FROM
109 EMPSKILL
110 WHERE
111 skillID IN (SELECT
112 skillID
113 FROM
114 SKILL
115 WHERE
116 skillRate > 15);
```

The Result Grid shows the following data:

| EMPID |
|-------|
| E3 |
| E4 |

The Action Output shows the following details:

| ID | Time | Action | Response | Duration / Fetch Time |
|----|----------|---------------------------------|-------------------|------------------------|
| 9 | 17:28:09 | SELECT DISTINCT EMPID FROM E... | 2 row(s) returned | 0.00048 sec / 0.000... |

Query Completed

A query to create view that contains employee id, employee last name, employee first name and skill id for each employee.

The screenshot shows the MySQL Workbench interface. The SQL editor contains the following query:

```
115 WHERE
116     skillRate > 15);
117
118 • CREATE VIEW skilled_employees AS
119     SELECT
120     employee.empID, empLastName, empFirstName, SkillID
121     FROM
122     employee
123     JOIN
124     empSkill ON employee.empID = empSkill.empID;
125
126 • SELECT * FROM skilled_employees;
```

The result grid displays the following data:

| empID | empLastName | empFirstName | SkillID |
|-------|-------------|--------------|---------|
| E1 | Mastache | Miguel | S1 |
| E2 | Craig | Brett | S1 |
| E3 | Williams | Josh | S2 |
| E3 | Williams | Josh | S4 |
| E4 | Cope | Leslie | S3 |

The Action Output pane shows the execution of the query:

| Time | Action | Response | Duration / Fetch Time |
|------|----------|--|------------------------|
| 10 | 17:29:50 | SELECT * FROM skilled_employees LIM... 5 row(s) returned | 0.00089 sec / 0.000... |