This lab is designed to enable our students to practice SQL queries and demonstrate an understanding of the logical scheme of relational database.

We use the classicmodels database as a MySQL sample database to help you work with MySQL quickly and effectively. The classicmodels database is a retailer of scale models of classic cars database. It contains typical business data such as customers, products, sales orders, sales order line items, etc.

**Team member:**

. .

**Installation**

* Follow the instruction and install AWS RDS MySQL DB instance:

<https://uncc.instructure.com/courses/155810/modules/items/3737295>

* Download Sample database at <https://uncc.instructure.com/files/13686949/download?download_frd=1>
* Open MySQL **Workbench**
  + Click “Database” -> “Manage Connections”
  + Specify connection name, AWS RDS database management system instance endpoint, port (3306 by default), and username
  + Click “Test connection”
  + Type your password to the admin user in AWS RDS database management system instance
  + Once it succeeds, click “Close”
  + Click “Database” -> “Connect to database”
* Import sample database to your AWS RDS MySQL database instance through your MySQL **workbench**
  + Click “File” -> “Open SQL script”
  + Click “Query” -> “Execute (All or Selection)”

Or run it in the OS terminal and type :

> mysql -h ${AWS RDS MySQL DB instance endpoint} -P 3306 -u admin -p -t < ${path}/mysqlsampledatabase.sql

Note that ${AWS RDS MySQL DB instance endpoint} should be changed to your own AWS RDS MySQL DB instance endpoint and ${path} should be replaced.

**Database schema**

The MySQL sample database schema consists of the following tables:

**Customers**: stores customer’s data.

**Products**: stores a list of scale model cars.

**ProductLines**: stores a list of product line categories.

**Orders**: stores sales orders placed by customers.

**OrderDetails**: stores sales order line items for each sales order.

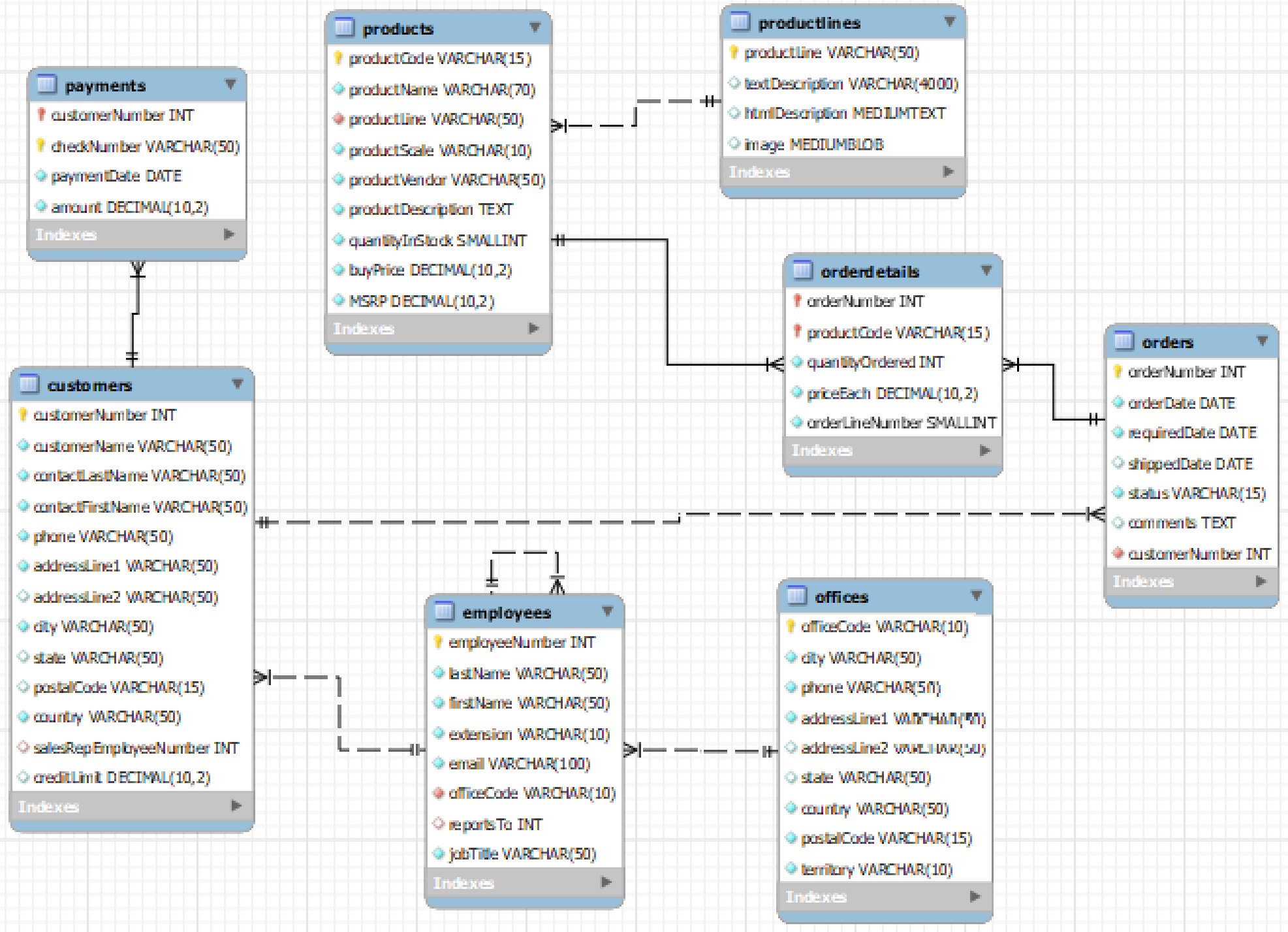
**Payments**: stores payments made by customers based on their accounts.

**Employees**: stores all employee information as well as the organization structure such as who reports to whom.

**Offices**: stores sales office data.

**Structure**

* The following diagram provides an overview of the structure of the Employees sample database.



**Query Usage**

**Activity 1.**

Let us try the following queries together in your MySQL Workbench or MySQL client terminal.

1. Examine what databases we have access to by

SHOW DATABASES;

1. Switch to the ***classicmodels*** database by

USE classicmodels;

1. Check how many tables this database includes by

SHOW TABLES;

1. Check what makes up the employees by

DESCRIBE employees;

**Single-entity model query**

1. Examine how many employees are there in total by

SELECT count(employeeNumber) FROM employees;

1. List all employee records by

SELECT \* FROM employees;

1. List employeenumber, first name, last name, email, job title of employees

SELECT employeeNumber, firstName, lastName, email, jobTitle FROM employees

1. List employeenumber, first name, last name, email, job title of employees whose last name is “Patternson” by

SELECT employeeNumber, firstName, lastName, email, jobTitle FROM employees

WHERE lastName="Patterson";

1. List president’s first name, last name, email, job title of employees

SELECT employeeNumber, firstName, lastName, email, jobTitle FROM employees

WHERE jobTitle="President";

1. Query how many different job titles in the employee entity table by

SELECT DISTINCT jobTitle FROM employees;

1. Query how many employees in each unique job title in the employee entity table by

SELECT jobTitle, count(\*) as count FROM employees

group by jobTitle;

1. Query Which job title has more than 5 employees by

SELECT jobTitle, count(\*) as count FROM employees

group by jobTitle HAVING count > 5;

1. Examine how many employs sharing unique last names employees have by

SELECT lastname, count(\*) as count FROM employees

group by lastName HAVING count > 1;

**Activity 2. Single-entity model query**

Form a team and choose a team leader.

Team leader copies this document to a google drive and share it with your team members.

Each team member choose one or more than one group of queries, write SQL statement in your MySQL Workbench or MySQL terminal, and execute it. If the number of subgroups is less than the number of your team members, it is okay to share. But please write down who work on which subquestion.

Discuss with your team member. Once all of you agree with the solution, please copy and paste your SQL statement in the google document.

Each team member should submit it **individually**.

Please **mark which part has been finished by whom**.

**Query group 2.A Customer Entity Table**

1. Query all customer information
2. Query customerName, phone, city, state, postalCode, the country of all customers;
3. Query customerName, phone, city, state, postalCode, the country of all USA customers;
4. Query how many total number of customer
5. Query the unique countries that customers come from
6. Query which country has the maximal number of customers in the database

**Query group 2.B Customer Entity Table**

1. Query all customer information
2. Query customerName, phone, city, state, postalCode, the country of all customers;
3. Query customerName, phone, city, state, postalCode, the country of all USA customers;
4. Query the total number of customers from USA
5. Query the list of unique states that customers come from
6. Query whether any customer comes from NC (North Carolina). If yes, please also list his/her customerName, phone, city

**Query group 2.C Office Entity or Employee Table**

1. Query all offices information
2. Query office code, phone, city, state, postalCode, the country of all offices;
3. Query office code, phone, city, state, postalCode, the country of all USA offices;
4. Query the total number of offices
5. Query the total number of offices from USA
6. Query employees from CA office

**Query group 2.D Product Entity Table**

1. Query all products information
2. Query product code, product name, product scale, product vendor, product line, quantity in stock, buy price of all products;
3. Query product name, product scale, product vendor, quantity in stock, buy price of “Classic cars” product line;
4. Query the total number of products
5. Query the total number of products of “Classic cars” product line
6. Query the number of products for each product line
7. Query the total number of product quantity for each product line
8. Query the total amount of money that these products are worthy of

**Activity 3.**

**(Optional) Multiple-entity model query**

1. List employeenumber, first name, last name, email, job title, office city of employees
2. Query which city has the most number of employees
3. Query which sales representative has a most number of customers
4. Query which production lines has a most number of products
5. Query which customer spent a most amount of money on products of “Classic cars” product line
6. Query which office sell the most number of products