# Johnson DAD 220 Cardinality and Targeted Data Template

Replace the bracketed text in this template with your screenshots and responses. Then submit it to the Module Four Lab for submission, grading, and feedback. Screenshots should be sized to approximately one quarter of a page. Written responses should be in complete sentences. Rename this document by adding your last name to the file name before you submit.

1. **Retrieve employee tuples and identify the number of employees** in San Francisco and New York.

A screenshot of a computer

Description automatically generated

In this screenshot, I was able to determine the number of employees as 6 in San Francisco and 2 in New York using the following commands: **SELECT firstName, lastName, jobTitle, offices.city**

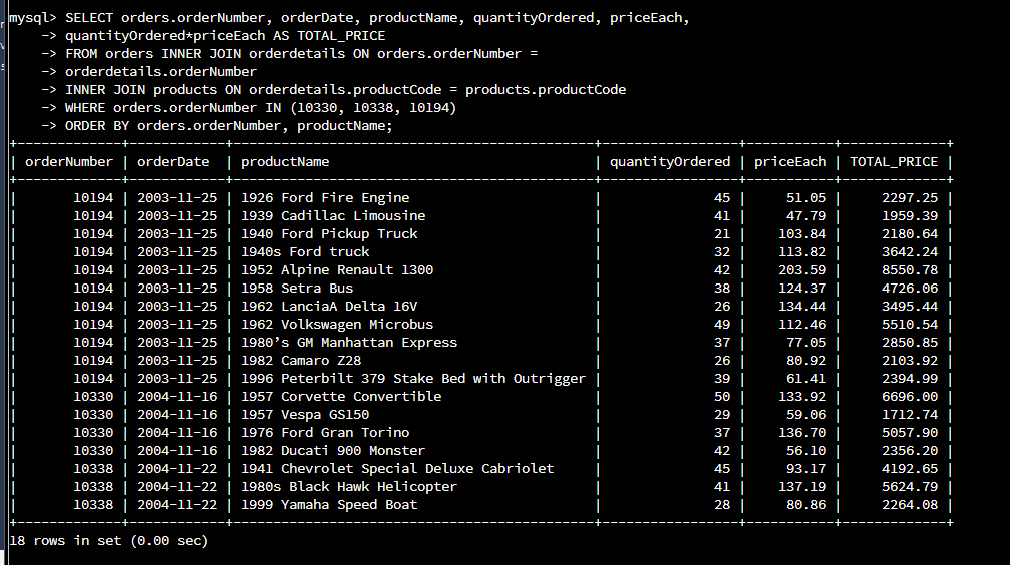
**FROM employees INNER JOIN offices ON employees.officeCode = offices.officeCode**

**WHERE state = 'CA';**

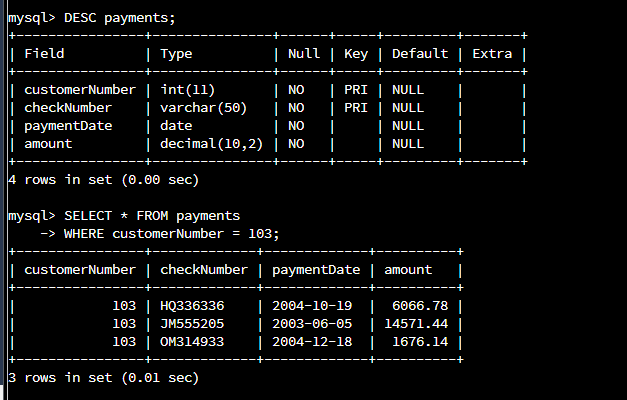
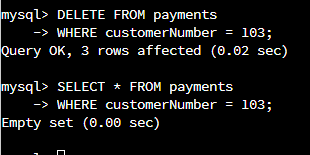
**SELECT firstName, lastName, jobTitle, offices.city**

**FROM employees INNER JOIN offices ON employees.officeCode = offices.officeCode**

**WHERE state = 'NY';**

1. **Retrieve order details** for orderNumber 10330, 10338, and 10194 and **identify** what **type of cardinality** this represents in the entity relationship model.
   1. 
2. With a many-to-one cardinality, I was able to retrieve order details for order numbers 10330, 10338, and 10194 using command: **SELECT orders.orderNumber, orderDate, productName, quantityOrdered, priceEach,**
3. **quantityOrdered\*priceEach AS TOTAL\_PRICE**
4. **FROM orders INNER JOIN orderdetails ON orders.orderNumber =**
5. **orderdetails.orderNumber**
6. **INNER JOIN products ON orderdetails.productCode = products.productCode**
7. **WHERE orders.orderNumber IN (10330, 10338, 10194)**

**ORDER BY orders.orderNumber, productName;**

1. **Delete records** from the payments table where the customer number equals 103.
   1. 
   2. 
2. These two screenshots, displays the payments, the deletion of the payments, and the outcome of deletion respectively using the following commands: **SELECT \* FROM payments**

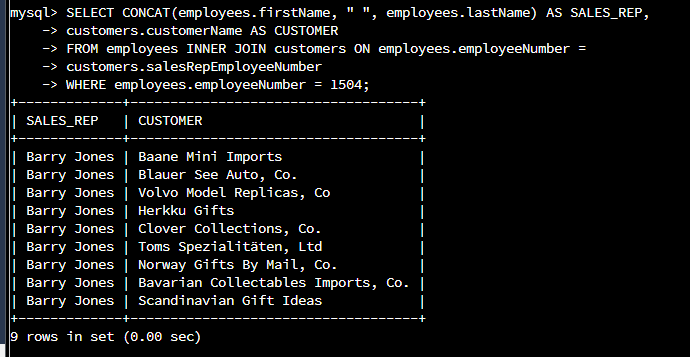
**WHERE customerNumber = 103;**

**DELETE FROM payments**

**WHERE customerNumber = 103;**

**SELECT \* FROM payments**

**WHERE customerNumber = 103;**

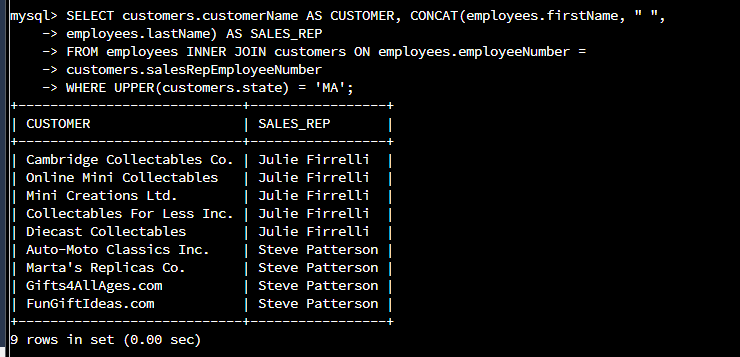
1. **Retrieve customer records** for sales representative Barry Jones and **identify** if the **relationships** are one-to-one or one-to-many**.**
   1. 
2. With the one-to-many relationship, I was able to determine the customer records for Barry Jones’ sales: **SELECT CONCAT(employees.firstName, " ", employees.lastName) AS SALES\_REP,**

**customers.customerName AS CUSTOMER**

**FROM employees INNER JOIN customers ON employees.employeeNumber =**

**customers.salesRepEmployeeNumber**

**WHERE employees.employeeNumber = 1504;**

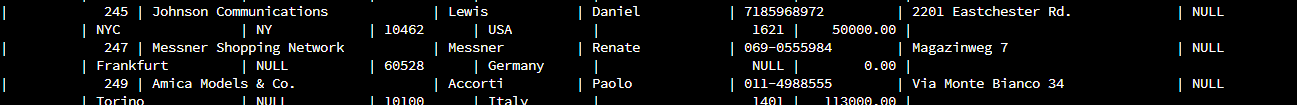
1. **Retrieve records** for customers who reside in Massachusetts and **identify** **their sales rep and the relationship of entities**. Identify if these entities demonstrate one-to-one or many-to-many relationships.
   1. 
2. With another one-to-many relationship, I was able to retrieve records for customers who reside in Massachusetts and identify their sales rep and the relationship of entities with the following command: **SELECT customers.customerName AS CUSTOMER, CONCAT(employees.firstName, " ",**

**employees.lastName) AS SALES\_REP**

**FROM employees INNER JOIN customers ON employees.employeeNumber =**

**customers.salesRepEmployeeNumber**

**WHERE UPPER(customers.state) = 'MA';**

1. **Add one customer record** with your last name using an INSERT statement. You may use the name of a celebrity or fictional character if you don’t use your own name.
   1. 

In this screenshot, I was able to add Johnson Communications to the customer records, with the additional attributes with the following command:

**INSERT INTO customers (customerNumber, customerName, contactLastName, contactFirstName, phone, addressLine1, addressLine2, city, state, postalCode, country, salesRepEmployeeNumber, creditLimit)**

**VALUES**

**(245, 'Johnson Communications', 'Lewis', 'Daniel', 7185968972, '2201 Eastchester Rd.', 'NULL', 'NYC', 'NY', 10462, 'USA', 1621, 50000.00);**

1. **Reflection**
   1. **Define how cardinality is applied** to the databases you’ve been working with and why different numbers of records returned from the different offices.
      1. The cardinality of a column is determined and deposited in the database tables for advanced purpose when conducting statistics. There are different numbers of records returned from the different offices based on updates.
   2. **Compare and contrast** the different **queries** you ran and how cardinality applies to them.
      1. Many-to-one had attributes like the order date and order numbers which had duplicate values that were paired with unique values. The one-to-many screenshots had attributes with unique values that were paired with attributes with duplicate values such as Barry Jones with multiple customers or one employee with many customers.
   3. **Describe two** of the crucial **benefits** **of cardinality** in this type of database.
      1. Cardinality assists database administrators faster by establishing the employee/sales rep to customer relationship and sales in a database. This simplifies seeing the interaction between employee/sales rep, customer, location, and contact information when searching for specific data or files for users.