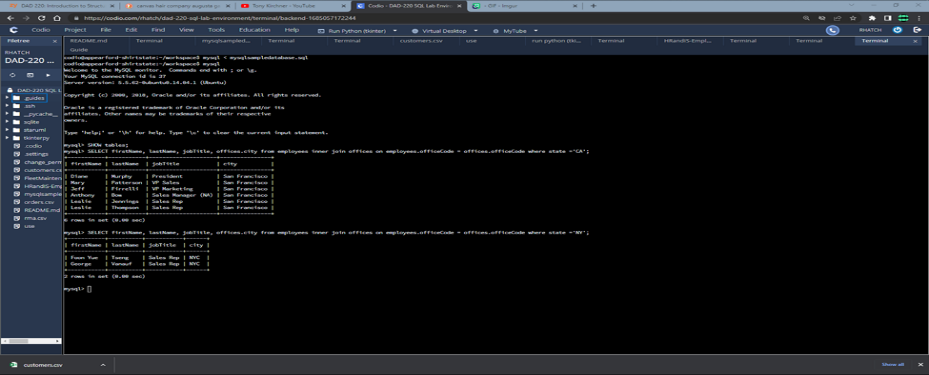
# Ryan Hatch

# SNHU

# Module Four Lab DAD 220 Cardinality and Targeted Data Template

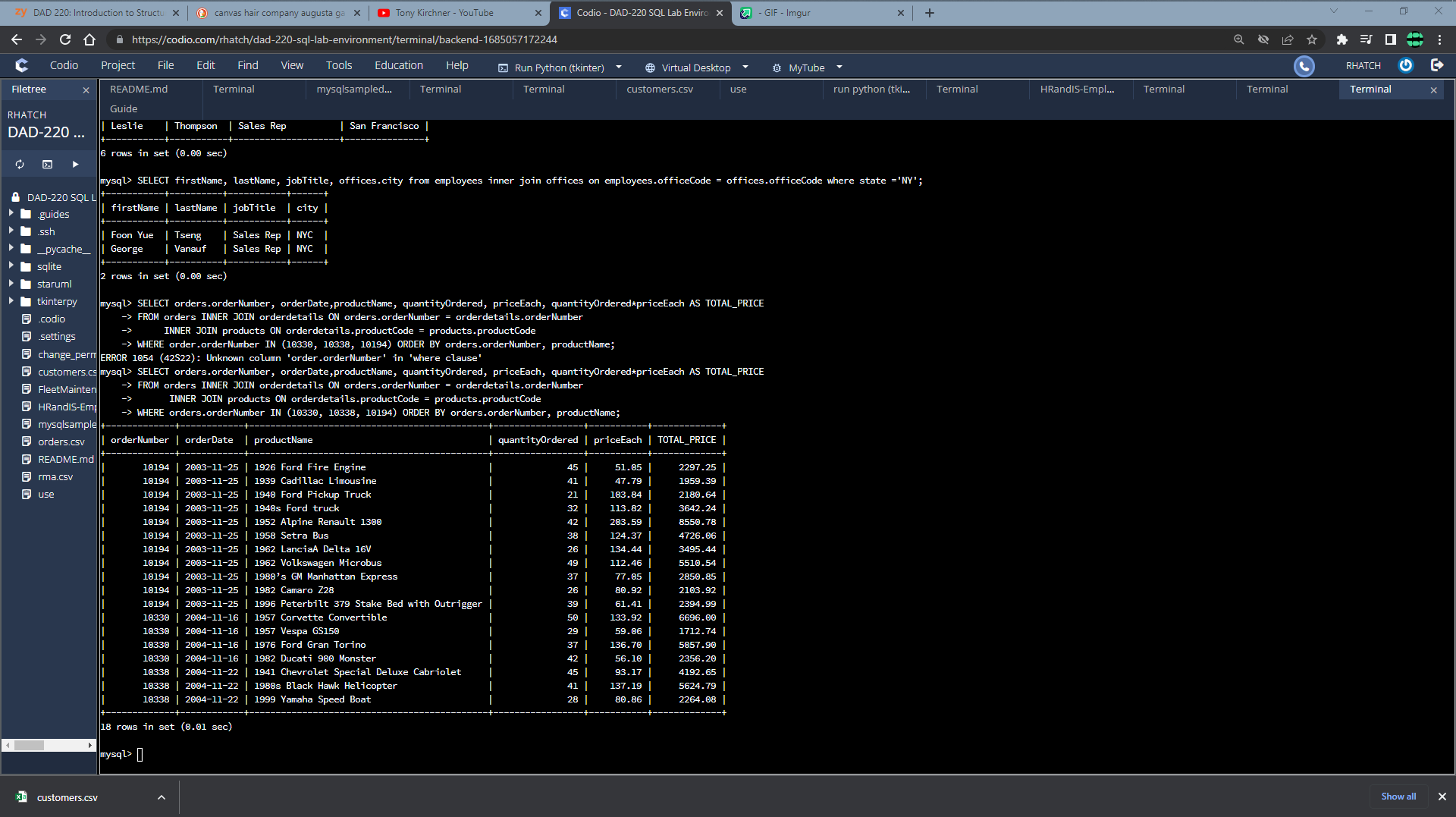
1. **Retrieve employee tuples and identify the number of employees** in San Francisco and New York.  
     
   In the output I pulled the employees for both California and New York.  
   I first opened the database, showed the tables and then I ran SELECT commands to get the Employees that worked in New York City and San Francisco.  
   SELECT firstName, lastName, jobTitles, offices.city from employees INNER JOIN offices on employees.officeCode = offices.officeCode where state = ‘CA’;  
   SELECT firstName, lastName, jobTitles, offices.city from employees INNER JOIN offices on employees.officeCode = offices.officeCode where state = ‘NY’;



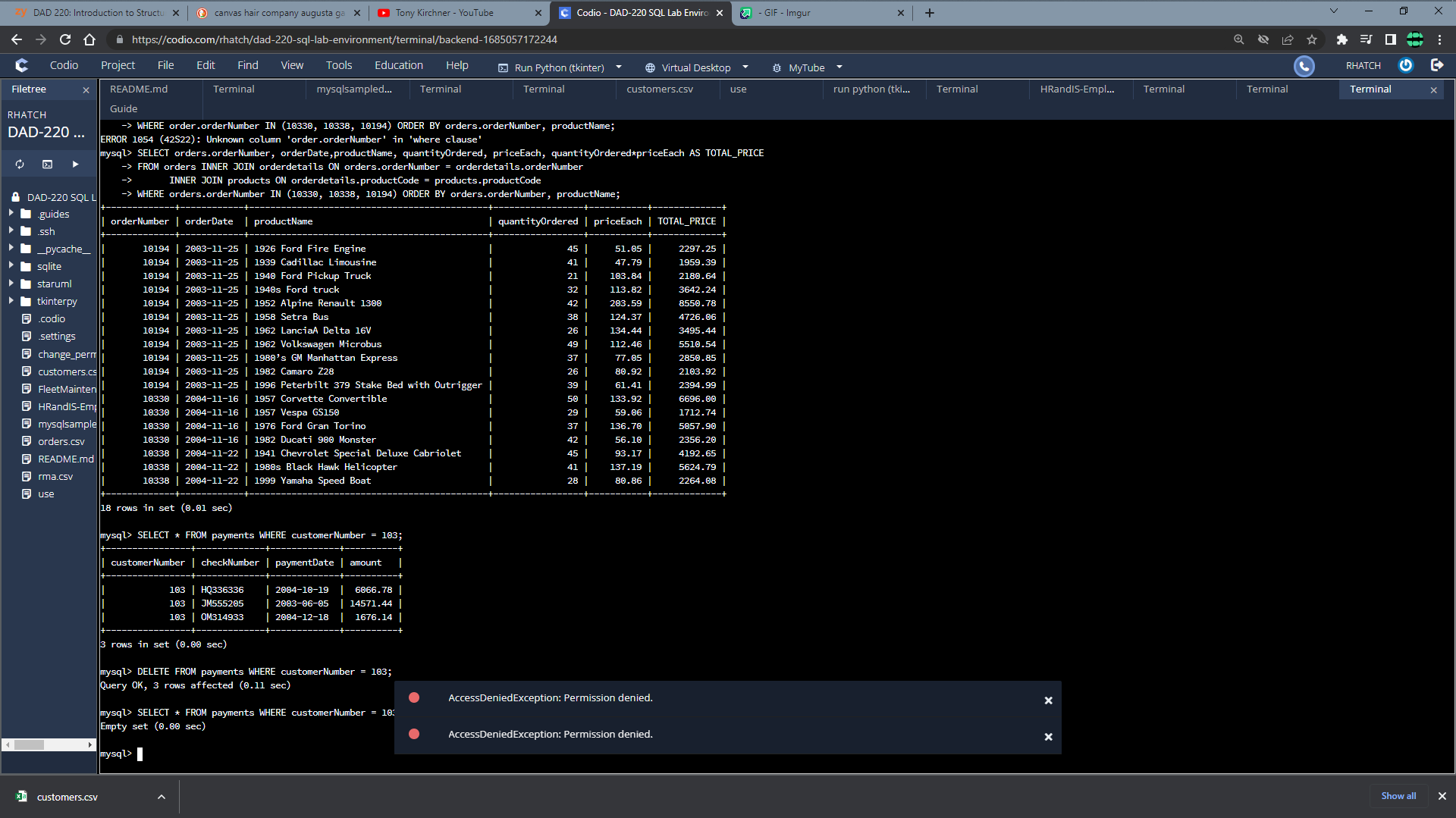
1. Retrieve order details for orderNumber 10330, 10338, and 10194 and identify what type of cardinality this represents in the entity relationship model.  
     
   I used the following input in order to retrieve the details for order 10330, 10338, and 10194:  
     
   FROM orders INNER JOIN orderdetails ON orders.orderNumber = orderdetails.orderNumber

INNER JOIN products ON orderdetails.productCode = products.productCode

WHERE orders.orderNumber IN (10330, 10338, 10194) ORDER BY orders.orderNumber, productName;  
  
The cardinality is high with a one-to-zero minima relationship.



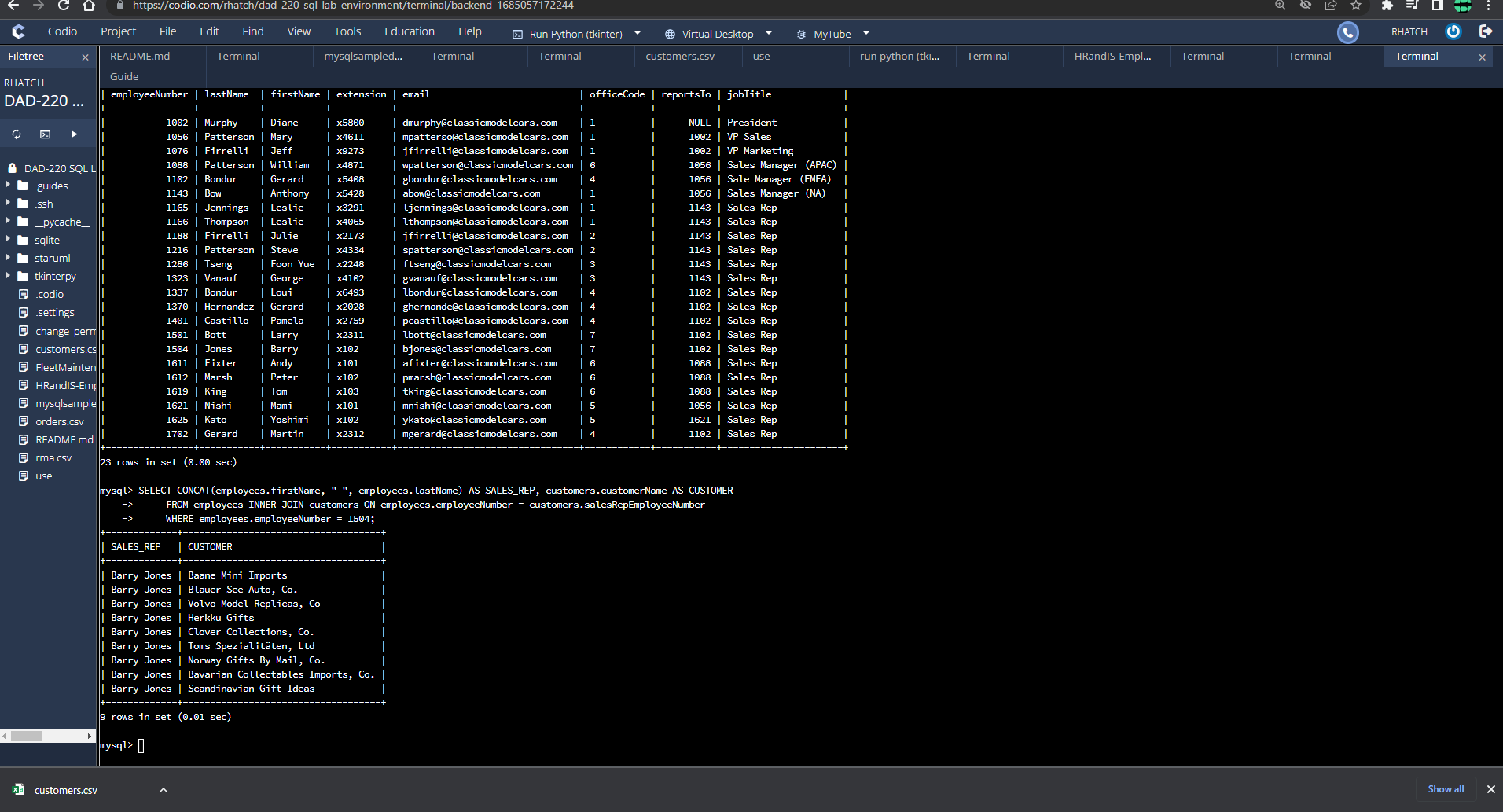
1. **Delete records** from the payments table where the customer number equals 103.  
     
   The input for this was as follows: SELECT \* FROM payments WHERE customerNumber = 103;



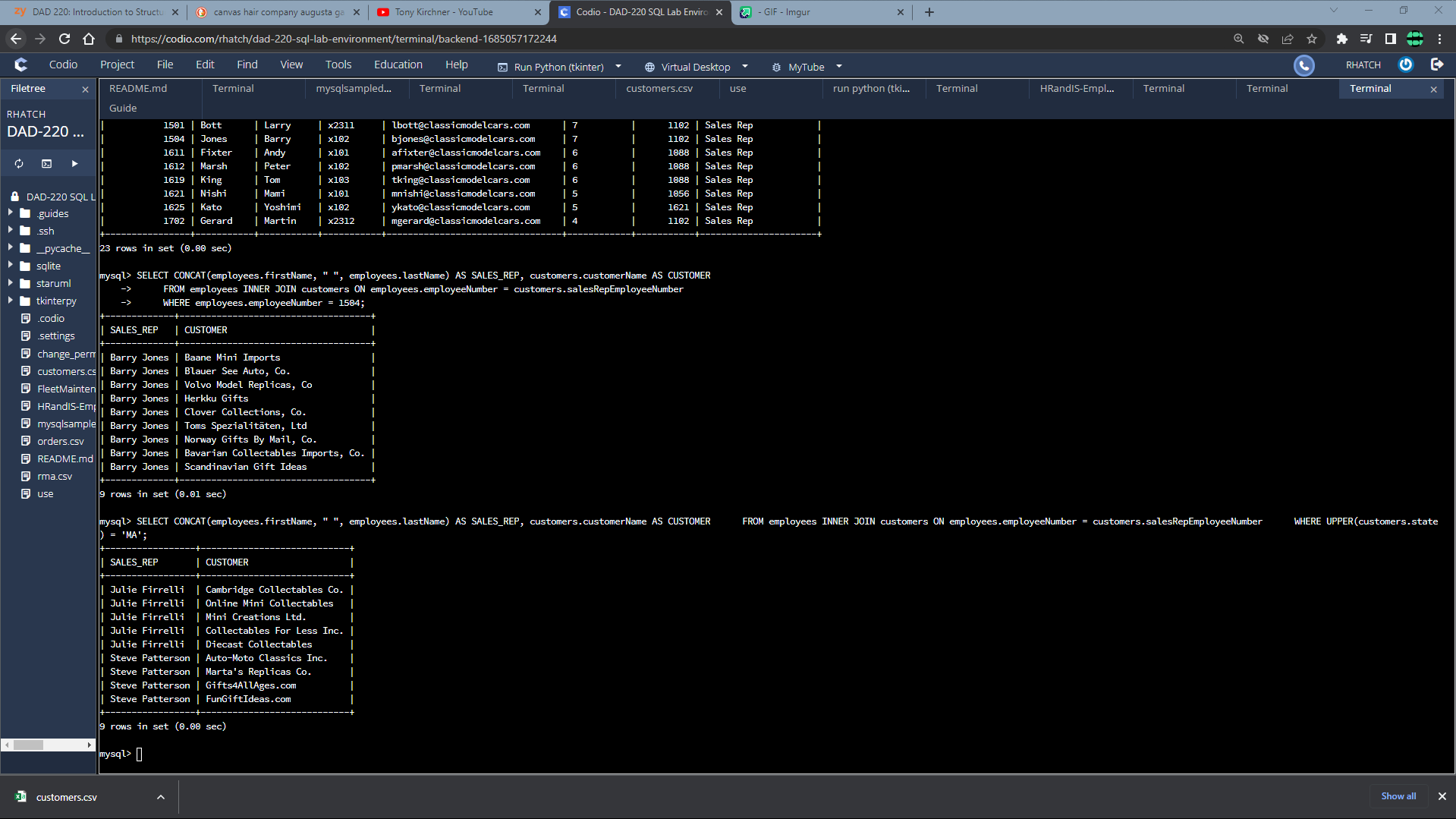
1. **Retrieve customer records** for sales representative Barry Jones and **identify** if the **relationships** are one-to-one or one-to-many**.**SELECT CONCAT(employees.firstName, " ", employees.lastName) AS SALES\_REP, customers.customerName AS CUSTOMER

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

WHERE employees.employeeNumber = 1504;



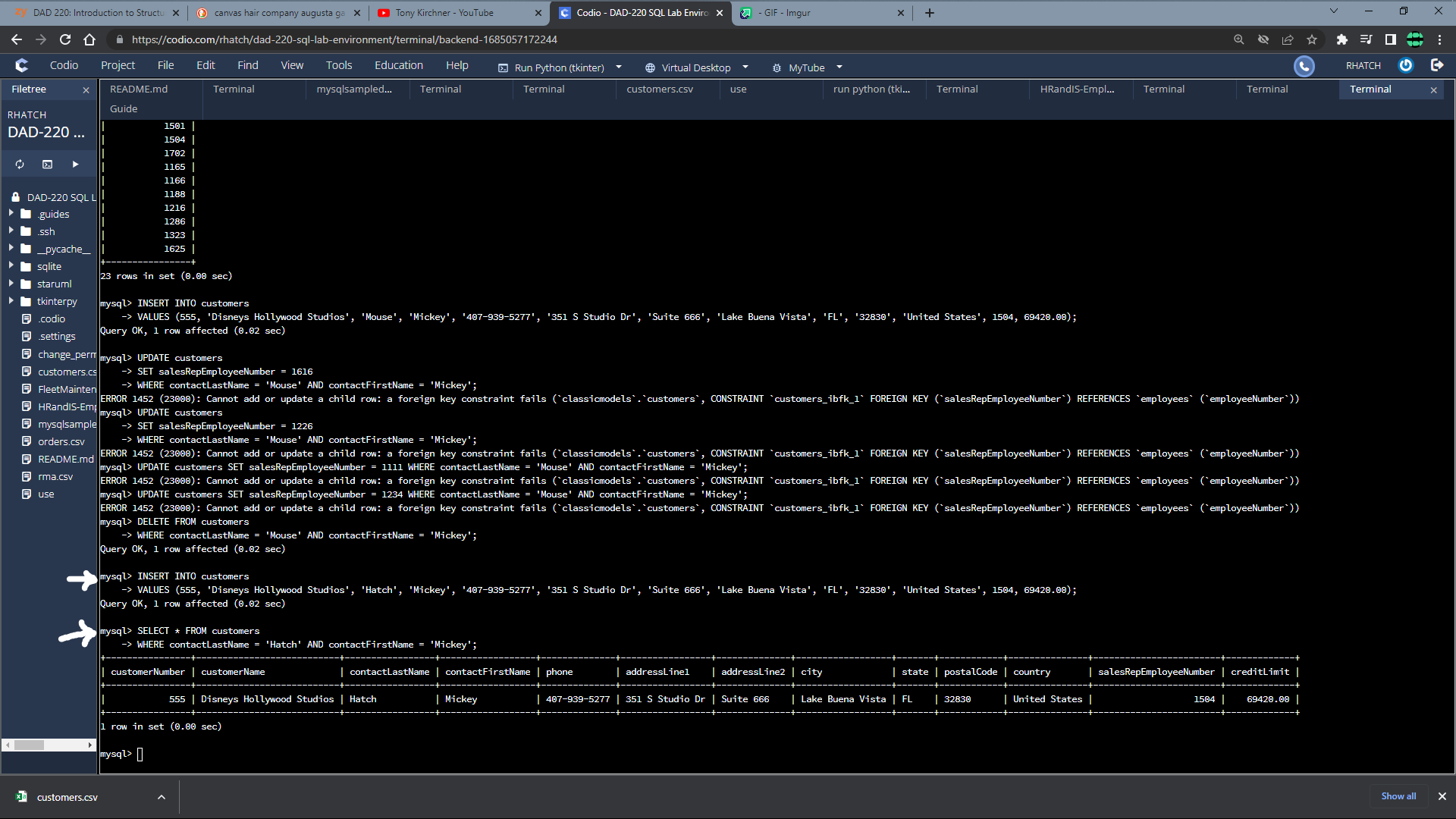
1. **Retrieve records** for customers who reside in Massachusetts and **identify** **their sales rep and the relationship of entities**. Identify if these entities demonstrate one-to-one or many-to-many relationships.  
     
   SELECT CONCAT(employees.firstName, " ", employees.lastName) AS SALES\_REP, customers.customerName AS CUSTOMER 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.  
     
   This query was to add Mickey Hatch into the database:  
   INSERT INTO customers VALUES (555, 'Disneys Hollywood Studios', 'Hatch', 'Mickey', '407-939-5277', '351 S Studio Dr', 'Suite 666', 'Lake Buena Vista', 'FL', '32838', 'United States', 1504, 69420.00);

This query was to show the addition of the new customer:

SELECT \* FROM customers WHERE contactLastName = 'Hatch' AND contactFirstName = 'Mickey';



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.  
        
      Cardinality in a database determines the uniqueness of values within a column and has a significant impact on the number of records retrieved from different offices. Offices with columns having higher cardinality, indicating more distinct values, will result in a greater number of records being returned. Due to varying numbers of unique values in their columns, offices can have different total record counts. When executing queries with different attributes, cardinality defines the relationship between tables, enabling the retrieval of information based on specified join conditions.
   2. **Compare and contrast** the different **queries** you ran and how cardinality applies to them.  
        
      Cardinality plays a role in several queries I executed since it involves understanding the relationships between data. Furthermore, queries serve as inputs and requests for specific data from the database. When making these requests, I connected various tables that shared a related attribute and then merged the data in to a single table. By combining the orders table with the order details table using order numbers, I was able to retrieve additional information for each order. This involved joining the primary key column to one table with the foreign key column of another table to obtain a matching output for those specific records I was handling.
   3. **Describe two** of the crucial **benefits** **of cardinality** in this type of database.

Understanding table cardinality is essential in database design. This allows you to identify areas where the normalization process might have faltered. By assessing cardinality, you can uncover improper relationships between tables and gather insights about how tables with multiple vales are structured. One of the key benefits of cardinality is its ability to establish connections between different tables. Without cardinality, there would be no way to establish relationships between entities, rendering the database fragmented and lacking the necessary links to navigate and retrieve meaningful information and data.