Southern New Hampshire University

DAD 220 Module Two Activity Template
Prepared on: 23 Mar., 2022
Prepared for: Prof. Aastha Agarwal
Prepared by: Alexander Ahmann

Contents

1 Problem set

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

   Answer: The base query to work out this question is:

   ```
   SELECT firstName, lastName, jobTitle, offices.city
   FROM employees
   INNER JOIN offices
   ON employees.officeCode = offices.OfficeCode
   WHERE state = '<state code>';
   ```

   Here, `<state code>` are the initials that identify a state in the United States of America. The initials `CA` and `NY` can be substituted to answer the question, and the screenshot (Fig. 1) demonstrates this.

2. **Retrieve order details for orderNumber** 10330, 10338 and 10194, and **identify what type of cardinality this represents in the entity relationship model.**

   Answer: The exact query to get this information is:

```
SELECT productCode, quantityOrdered, priceEach, orderLineNumber
FROM orders INNER JOIN orderdetails
ON orders.orderNumber = orderdetails.orderNumber
WHERE orderdetails.orderNumber = 10330
    OR orderdetails.orderNumber = 10338
    OR orderdetails.orderNumber = 10194;
```

This SQL query executed successfully (see Fig. 2).

Regarding the cardinality of the `classicmodels` database, I shall consult the E.R. diagram for `classicmodels` presented to students (Fig. 3). Looking at the relationship between `orders` and `orderdetails`, it seems to be a many-to-one relationship based on the markings from the connectors describing the relationship between the two tables.

3. **Delete records** from the payments table where the customer number equals 103.

   Answer: See Fig. 4 and Fig. 5 for evidence that I have indeed deleted entries where the `customerNumber = 104`.

4. **Retrieve customer records** for employee Rep Barry Jones and **identify** if the relationships are one-to-one or one-to-many.

   Answer: The exact query that I used to work out this problem is:

```
SELECT employees.lastName AS "Rep. Surname", employees.firstName
    AS "Rep. Given Name",
contactLastName, contactFirstName, phone, addressLine1,
    city, state, postalcode, country
FROM customers INNER JOIN employees
ON customers.salesRepEmployeeNumber = employees.employeeNumber
WHERE customers.salesRepEmployeeNumber = 1504;
```

   See Fig. 6 for my SQL query and its results. Judging from the nature of the query and the E.R. diagram, this appears to be a one-to-many relationship.

5. Retrieve records for customers who reside in Massachusetts, then identify their sales rep and the relationship of entities. Identify if these entities demonstrate one-to-one or many-to-many relationships.

   Answer: The exact SQL statement that I used to work this problem out is:

```
SELECT salesRepEmployeeNumber AS "Rep. Number", customerName,
    contactLastName, contactFirstName, addressLine1, postalCode
FROM customers INNER JOIN employees
ON customers.salesRepEmployeeNumber = employees.employeeNumber
WHERE customers.state = "MA";
```

See Fig. 7 for the screenshot in where I demonstrate this query returning the selected queries. Judging from the nature of the query and

6. 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.

   Answer: The exact SQL query statement that I used to complete this task is:

```
INSERT INTO customers
    (customerNumber, customerName, contactLastName,
    contactFirstName, phone, addressLine1, addressLine2,
    city, state, postalCode, country,
    salesRepEmployeeNumber, creditLimit)
VALUES (497, "La Rochelle Gifts", "Reinhard", "Claudia",
    "03604 57 18 84", "Heinrich Heine Platz 36",
    "99954 Bad Tennstedt", "Berlin", NULL, NULL, "Germany",
1286, 50666.00);
```

   and the SQL query that I used to confirm my results is:

```
SELECT * FROM customers
WHERE customerNumber = 497;
```

   See Fig. 8 for the screenshot demonstrating that I was able to complete this task.

7. **Reflection**

   (a) **Define how cardinality** is applied to the databases you've been working with and why different numbers of records from the different offices.

Answer: In the context of databases and entity-relationship modelling, *cardinality* can be described as a "range," for lack of a better term, between the smallest number of attribute instances and the largest numbers of attribute instances.[1]

In this lab, cardinality was used a lot in relationships between different tables in the MySQL database that I worked on. In particular, there were various "one-to-many" and "one-to-one" relationships in this database between the `customers` and `employees` table that were linked by the `customers`' foreign key `salesRepEmployeeNumber` to the `employees`' `employeeNumber`'s primary key.

(b) **Compare and contrast** the different queries you ran and how cardinality applies to them.

Answer: The `SELECT` and `INNER JOIN` queries can be used to join tables together where the primary key is equal to the foreign key and the `INSERT` and `DELETE` queries can be used to modify tables even if they have "one-to-one," "one-to-many" or "many-to-many" relationships.

The similarities is that they both make use of cardinality, and the difference is that `SELECT-INNER JOIN` statements retrieves data by joining databases, and `INSERT` or `DELETE` statements affect adds or removes records from the database while using notions in cardinality to prevent the relationship from becoming "unstable."

(c) **Describe two of the crucial benefits** of the cardinality in this type of database.

Answer: One benefit of cardinality is that is can help data engineers organise their thoughts and another benefit of cardinality is that it can help the database maintain its integrity.

2 Appendix: Importing the database

This project uses the `classicmodels` database, and the screenshot (Fig. 9) demonstrates that I was able to import it into my MySQL instance.

[1] Attribute instances describe the "relationships" between database entities

3 Figures

```
mysql> # Return employee tuples and identify the number of employees in San Francisco
mysql> SELECT firstName, lastName, jobTitle, offices.city
    -> FROM employees
    -> INNER JOIN offices ON employees.officeCode = offices.OfficeCode
    -> WHERE state = 'CA';
+-----------+-----------+--------------------+---------------+
| firstName | lastName  | jobTitle           | city          |
+-----------+-----------+--------------------+---------------+
| Diane     | Murphy    | President          | San Francisco |
| Mary      | Patterson | VP Sales           | San Francisco |
| Jeff      | Firrelli  | VP Marketing       | San Francisco |
| Anthony   | Bow       | Sales Manager (NA) | San Francisco |
| Leslie    | Jennings  | Sales Rep          | San Francisco |
| Leslie    | Thompson  | Sales Rep          | San Francisco |
+-----------+-----------+--------------------+---------------+
6 rows in set (0.00 sec)

mysql> # Return employee tuples and identify the number of employees in New York
mysql> SELECT firstName, lastName, jobTitle, offices.city
    -> FROM employees
    -> INNER JOIN offices on employees.officeCode = offices.officeCode
    -> WHERE state = 'NY';
+-----------+----------+-----------+------+
| firstName | lastName | jobTitle  | city |
+-----------+----------+-----------+------+
| Foon Yue  | Tseng    | Sales Rep | NYC  |
| George    | Vanauf   | Sales Rep | NYC  |
+-----------+----------+-----------+------+
2 rows in set (0.00 sec)
```

Figure 1: **“My solution to the first question.”**

```
mysql> # Retrive order details for orderNumbers 10330, 10338, 10194
mysql> SELECT productCode, quantityOrdered, priceEach, orderLineNumber
    -> FROM orders INNER JOIN orderdetails
    -> ON orders.orderNumber = orderdetails.orderNumber
    -> WHERE orderdetails.orderNumber = 10330
    -> OR orderdetails.orderNumber = 10338
    -> OR orderdetails.orderNumber = 10194;
+-------------+-----------------+-----------+-----------------+
| productCode | quantityOrdered | priceEach | orderLineNumber |
+-------------+-----------------+-----------+-----------------+
| S10_1949    |              42 |    203.59 |              11 |
| S10_4962    |              26 |    134.44 |               4 |
| S12_1666    |              38 |    124.37 |               8 |
| S18_1097    |              21 |    103.84 |              10 |
| S18_2432    |              45 |     51.05 |               2 |
| S18_4600    |              32 |    113.82 |               5 |
| S18_4668    |              41 |     47.79 |               9 |
| S24_2300    |              49 |    112.46 |               1 |
| S32_1268    |              37 |     77.05 |               3 |
| S32_3522    |              39 |     61.41 |               7 |
| S700_2824   |              26 |     80.92 |               6 |
| S18_3482    |              37 |    136.70 |               3 |
| S18_3782    |              29 |     59.06 |               2 |
| S18_4721    |              50 |    133.92 |               4 |
| S24_2360    |              42 |     56.10 |               1 |
| S18_1662    |              41 |    137.19 |               1 |
| S18_3029    |              28 |     80.86 |               3 |
| S18_3856    |              45 |     93.17 |               2 |
+-------------+-----------------+-----------+-----------------+
18 rows in set (0.00 sec)
```

Figure 2: **“My solution to the second question.”**

DAD 220 Module Four MYSQL ERD

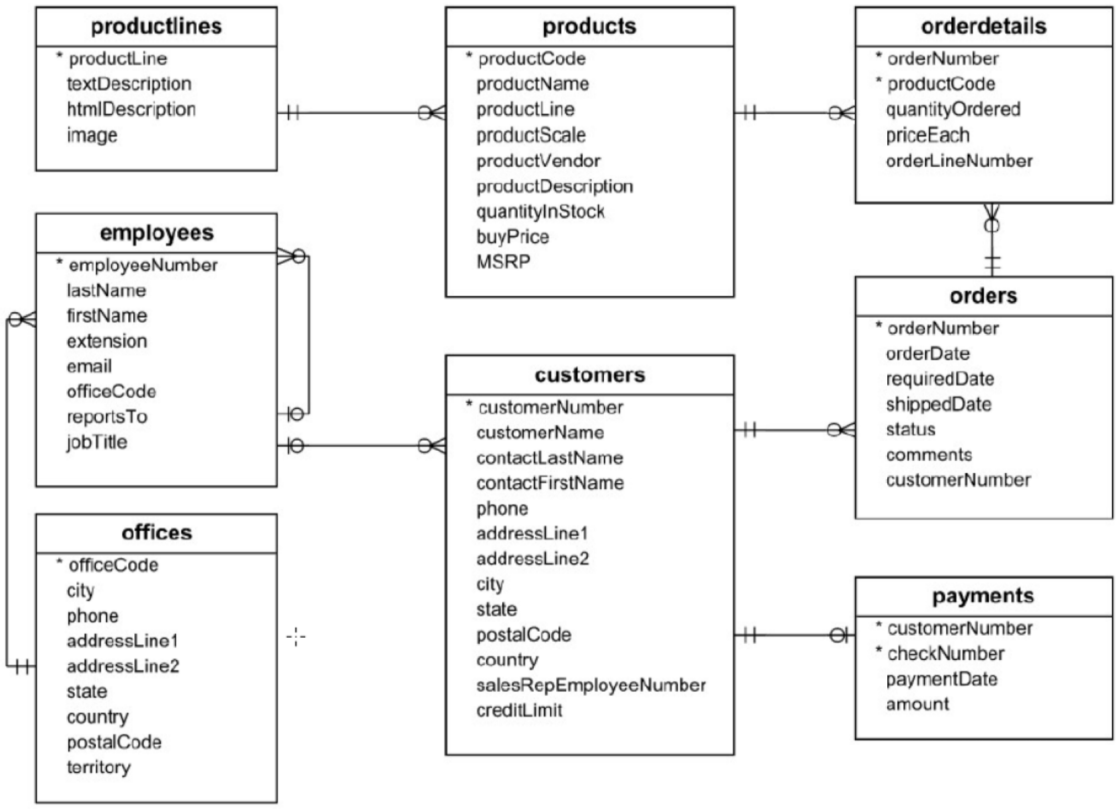

Figure 3: **"Entity-relationship diagram for** `classicmodels`**."**

```
mysql> # Describe the payments table
mysql> DESCRIBE payments;
+----------------+---------------+------+-----+---------+-------+
| Field          | Type          | Null | Key | Default | Extra |
+----------------+---------------+------+-----+---------+-------+
| customerNumber | int(11)       | NO   | PRI | NULL    |       |
| checkNumber    | varchar(50)   | NO   | PRI | NULL    |       |
| paymentDate    | date          | NO   |     | NULL    |       |
| amount         | decimal(10,2) | NO   |     | NULL    |       |
+----------------+---------------+------+-----+---------+-------+
4 rows in set (0.01 sec)

mysql> SELECT * FROM payments WHERE customerNumber = 103;
+----------------+-------------+-------------+----------+
| customerNumber | checkNumber | paymentDate | amount   |
+----------------+-------------+-------------+----------+
|            103 | HQ336336    | 2004-10-19  |  6066.78 |
|            103 | JM555205    | 2003-06-05  | 14571.44 |
|            103 | OM314933    | 2004-12-18  |  1676.14 |
+----------------+-------------+-------------+----------+
3 rows in set (0.02 sec)
```

Figure 4: **"Showing records from payments where customerNumber = 103."**

```
mysql> DELETE FROM payments WHERE customerNumber = 103;
Query OK, 3 rows affected (0.02 sec)

mysql> SELECT * FROM payments;
+----------------+-------------+-------------+-----------+
| customerNumber | checkNumber | paymentDate | amount    |
+----------------+-------------+-------------+-----------+
|            112 | BO864823    | 2004-12-17  |  14191.12 |
|            112 | HQ55022     | 2003-06-06  |  32641.98 |
|            112 | ND748579    | 2004-08-20  |  33347.88 |
|            114 | GG31455     | 2003-05-20  |  45864.03 |
|            114 | MA765515    | 2004-12-15  |  82261.22 |
|            114 | NP603840    | 2003-05-31  |   7565.08 |
|            114 | NR27552     | 2004-03-10  |  44894.74 |
|            119 | DB933704    | 2004-11-14  |  19501.82 |
|            119 | LN373447    | 2004-08-08  |  47924.19 |
|            119 | NG94694     | 2005-02-22  |  49523.67 |
|            121 | DB889831    | 2003-02-16  |  50218.95 |
|            121 | FD317790    | 2003-10-28  |   1491.38 |
|            121 | KI831359    | 2004-11-04  |  17876.32 |
|            121 | MA302151    | 2004-11-28  |  34638.14 |
|            124 | AE215433    | 2005-03-05  | 101244.59 |
|            124 | BG255406    | 2004-08-28  |  85410.87 |
|            124 | CQ287967    | 2003-04-11  |  11044.30 |
|            124 | ET64396     | 2005-04-16  |  83598.04 |
|            124 | HI366474    | 2004-12-27  |  47142.70 |
```

Figure 5: **"Deleting records from payments where customerNumber = 103."**

```
mysql> SELECT employees.lastName AS "Rep. Surname", employees.firstName AS "Rep. Given Name",
    -> contactLastName, contactFirstName, phone, addressLine1, city, state, postalcode, country
    -> FROM customers INNER JOIN employees
    -> ON customers.salesRepEmployeeNumber = employees.employeeNumber
    -> WHERE customers.salesRepEmployeeNumber = 1504;
+--------------+-----------------+----------------+-----------------+-----------------+-----------------------+-----------+-------+------------+---------+
| Rep. Surname | Rep. Given Name | contactLastName | contactFirstName | phone           | addressLine1          | city      | state | postalcode | country |
+--------------+-----------------+----------------+-----------------+-----------------+-----------------------+-----------+-------+------------+---------+
| Jones        | Barry           | Bergulfsen     | Jonas           | 07-98 9555      | Erling Skakkes gate 78 | Stavern   | NULL  | 4110       | Norway  |
| Jones        | Barry           | Keitel         | Roland          | +49 69 66 90 2555 | Lyonerstr. 34       | Frankfurt | NULL  | 60528      | Germany |
| Jones        | Barry           | Berglund       | Christina       | 0921-12 3555    | Berguvsvägen  8       | Luleå     | NULL  | S-958 22   | Sweden  |
| Jones        | Barry           | Oeztan         | Veysel          | +47 2267 3215   | Brehmen St. 121       | Bergen    | NULL  | N 5804     | Norway  |
| Jones        | Barry           | Cassidy        | Dean            | +353 1862 1555  | 25 Maiden Lane        | Dublin    | NULL  | 2          | Ireland |
| Jones        | Barry           | Pfalzheim      | Henriette       | 0221-5554327    | Mehrheimerstr. 369    | Köln      | NULL  | 50739      | Germany |
| Jones        | Barry           | Klaeboe        | Jan             | +47 2212 1555   | Drammensveien 126A    | Oslo      | NULL  | N 0106     | Norway  |
| Jones        | Barry           | Donnermeyer    | Michael         |  +49 89 61 08 9555 | Hansastr. 15       | Munich    | NULL  | 80686      | Germany |
| Jones        | Barry           | Larsson        | Martha          | 0695-34 6555    | Åkergatan 24          | Bräcke    | NULL  | S-844 67   | Sweden  |
+--------------+-----------------+----------------+-----------------+-----------------+-----------------------+-----------+-------+------------+---------+
9 rows in set (0.00 sec)
```

Figure 6: **“My solution to question 4.”**

```
mysql> SELECT salesRepEmployeeNumber AS "Rep. Number", customerName, contactLastName, contactFirstName, addressLine1,
 postalCode
    -> FROM customers INNER JOIN employees
    -> ON customers.salesRepEmployeeNumber = employees.employeeNumber
    -> WHERE customers.state = "MA";
+-------------+--------------------------+-----------------+------------------+-----------------------+------------+
| Rep. Number | customerName             | contactLastName | contactFirstName | addressLine1          | postalCode |
+-------------+--------------------------+-----------------+------------------+-----------------------+------------+
|        1188 | Cambridge Collectables Co. | Tseng         | Jerry            | 4658 Baden Av.        | 51247      |
|        1188 | Online Mini Collectables | Barajas         | Miguel           | 7635 Spinnaker Dr.    | 58339      |
|        1188 | Mini Creations Ltd.      | Huang           | Wing             | 4575 Hillside Dr.     | 50553      |
|        1188 | Collectables For Less Inc. | Nelson        | Allen            | 7825 Douglas Av.      | 58339      |
|        1188 | Diecast Collectables     | Franco          | Valarie          | 6251 Ingle Ln.        | 51003      |
|        1216 | Auto-Moto Classics Inc.  | Taylor          | Leslie           | 16780 Pompton St.     | 58339      |
|        1216 | Marta's Replicas Co.     | Hernandez       | Marta            | 39323 Spinnaker Dr.   | 51247      |
|        1216 | Gifts4AllAges.com        | Yoshido         | Juri             | 8616 Spinnaker Dr.    | 51003      |
|        1216 | FunGiftIdeas.com         | Benitez         | Violeta          | 1785 First Street     | 50553      |
+-------------+--------------------------+-----------------+------------------+-----------------------+------------+
9 rows in set (0.00 sec)
```

Figure 7: **"My solution to question 5."**

```
mysql> INSERT INTO customers (customerNumber, customerName, contactLastName, contactFirstName, phone, addressLine1,
addressLine2, city, state, postalCode, country, salesRepEmployeeNumber, creditLimit) VALUES (497, "La Rochelle Gifts
", "Reinhard", "Claudia", "03604 57 18 84", "Heinrich Heine Platz 36", "99954 Bad Tennstedt", "Berlin", NULL, NULL,
"Germany", 1286, 50666.00);
Query OK, 1 row affected (0.02 sec)

mysql> SELECT * FROM customers WHERE customerNumber = 497;
+----------------+--------------------+-----------------+------------------+----------------+-----------------+
| customerNumber | customerName       | contactLastName | contactFirstName | phone          | addressLine1    |
| addressLine2   | city   | state | postalCode | country | salesRepEmployeeNumber | creditLimit |
+----------------+--------------------+-----------------+------------------+----------------+-----------------+
|            497 | La Rochelle Gifts | Reinhard        | Claudia          | 03604 57 18 84 | Heinrich Heine Platz 36
 | 99954 Bad Tennstedt | Berlin | NULL  | NULL    | Germany |                1286 |   50666.00 |
+----------------+--------------------+-----------------+------------------+----------------+-----------------+
1 row in set (0.00 sec)
```

Figure 8: **"My solution to question 6."**

```
codio@mangogorilla-ohiochef:~/workspace$ mysql < mysqlsampledatabase.sql
codio@mangogorilla-ohiochef:~/workspace$ mysql
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 37
Server version: 5.5.62-0ubuntu0.14.04.1 (Ubuntu)

Copyright (c) 2000, 2018, Oracle and/or its affiliates. All rights reserved.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> show databases;
+--------------------+
| Database           |
+--------------------+
| information_schema |
| QuantigrationRMA   |
| ahmann             |
| classicmodels      |
| mysql              |
| performance_schema |
+--------------------+
6 rows in set (0.00 sec)

mysql> use classicmodels;
Reading table information for completion of table and column names
You can turn off this feature to get a quicker startup with -A

Database changed
mysql> show tables;
+-------------------------+
| Tables_in_classicmodels |
+-------------------------+
| customers               |
| employees               |
| offices                 |
| orderdetails            |
| orders                  |
| payments                |
| productlines            |
| products                |
+-------------------------+
8 rows in set (0.00 sec)
```

Figure 9: **"Housekeeping work to get the database initialised."**