IS 456 Database Systems Management

HOP09 – Advanced SQL Queries

1/6/2019 Developed by Clark Ngo
Center for Information Assurance (CIAE) @City University of Seattle (CityU)

Before You Start

- Version numbers may not match with the most current version at the time of writing. If given the option to choose between stable release (long-term support) or most recent, please choose the stable release rather than beta-testing version.
- This tutorial targets Windows users and MacOS users.
- There might be subtle discrepancies along the steps. Please use your best judgement while going through this cookbook style tutorial to complete each step.
- For your working directory, use your course number. This tutorial may use a different course number as an example.
- The directory path shown in screenshots may be different from yours.
- If you are not sure what to do or confused with any steps:
  1. Consult the resources listed below.
  2. If you cannot solve the problem after a few tries, ask a TA for help.

Learning Outcomes

Students will be able to:

- Join Tables
- INNER JOIN
- INNER JOIN with operators
- LEFT JOIN
- RIGHT JOIN

Resources

- SQL Tutorial – https://www.w3schools.com/sql/default.asp

Preparation

Run your Docker Application

Find the Docker App and double-click

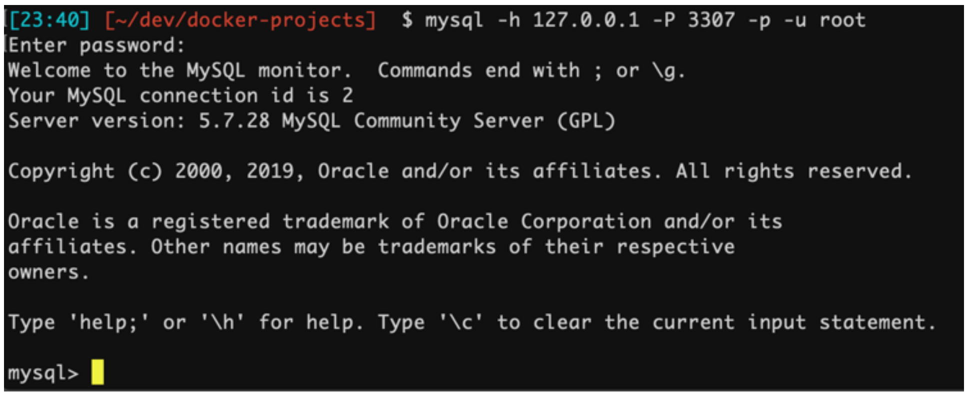
Run an MySQL interactive shell

Open your terminal / command prompt and type the following:

```
mysql -h 127.0.0.1 -P 3307 -p -u root
```

When prompted for password: *passwd*

Example output in MacOS:



```
[23:40] [~/dev/docker-projects]  $ mysql -h 127.0.0.1 -P 3307 -p -u root
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 2
Server version: 5.7.28 MySQL Community Server (GPL)

Copyright (c) 2000, 2019, 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>
```

Use a Database
Syntax: USE *database_name*;

```
USE classicmodels;
```

The WHERE clause

Extract records that satisfy a condition
Syntax: SELECT *column_name* FROM *table_name* WHERE *condition*;

Match a string

```
SELECT contactFirstName, city FROM customers WHERE country = 'USA';
```

```
+------------------+---------------+
| contactFirstName | city          |
+------------------+---------------+
| Jean             | Las Vegas     |
| Susan            | San Rafael    |
| Julie            | San Francisco |
```

Match numerical values

```
SELECT customerNumber, amount FROM payments WHERE amount >= 50000;
```

```
+----------------+-----------+
| customerNumber | amount    |
+----------------+-----------+
|            114 |  82261.22 |
|            121 |  50218.95 |
|            124 | 101244.59 |
|            124 |  85410.87 |
|            124 |  83598.04 |
|            124 |  55639.66 |
```

Match a pattern

```
SELECT contactFirstName, city FROM customers WHERE city LIKE 's%';
```

```
+------------------+------------------+
| contactFirstName | city             |
+------------------+------------------+
| Jonas            | Stavern          |
| Susan            | San Rafael       |
| Julie            | San Francisco    |
| Eric             | Singapore        |
| Wendy            | Singapore        |
```

Questions you can answer for submission:

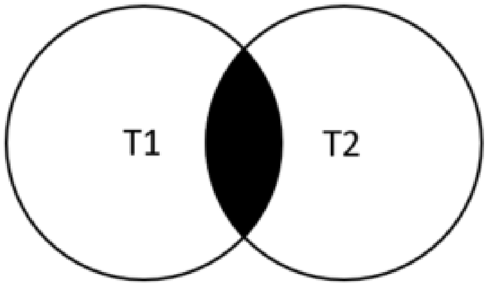
Technical: What is the command for? Why would you use the command?

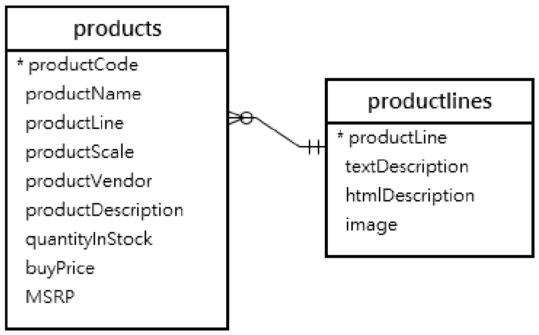
Knowledge: Would a business user understand this data?

SQL JOIN

INNER JOIN - Combine rows from two or more tables

Syntax: SELECT *column/s* FROM *table1* INNER JOIN *table2* ON *join_condition*;





Source: https://www.mysqltutorial.org/mysql-inner-join.aspx

In this diagram, the table *products* has the column *productLine* that references the column *productline* of the table *productlines* . The column *productLine* in the table *products* is called the *foreign key* column.
Typically, you join tables that have foreign key relationships like the *productlines* and *products* tables.
Suppose you want to get:

- The *productCode* and *productName* from the *products* table.
- The *textDescription* of product lines from the *productlines* table.

To do this, you need to select data from both tables by matching rows based on values in the *productline* column using the INNER JOIN clause as follows:

Extract records that matches in both tables

```
SELECT productCode, productName, textDescription FROM products t1 INNER JOIN productlines t2 ON t1.productline =
t2.productline;
```

```
| S10_1949    | 1952 Alpine Renault 1300              | Attention car enthusiasts: Make your wildest car ownership dreams come true. Whether you are looking for classic muscle
cars, dream sports cars or movie-inspired miniatures, you will find great choices in this category. These replicas feature superb attention to detail and craftsmanship and offer
features such as working steering system, opening forward compartment, opening rear trunk with removable spare wheel, 4-wheel independent spring suspension, and so on.
The models range in size from 1:10 to 1:24 scale and include numerous limited edition and several out-of-production vehicles. All models include a certificate of authenticity
from their manufacturers and come fully assembled and ready for display in the home or office. |
```

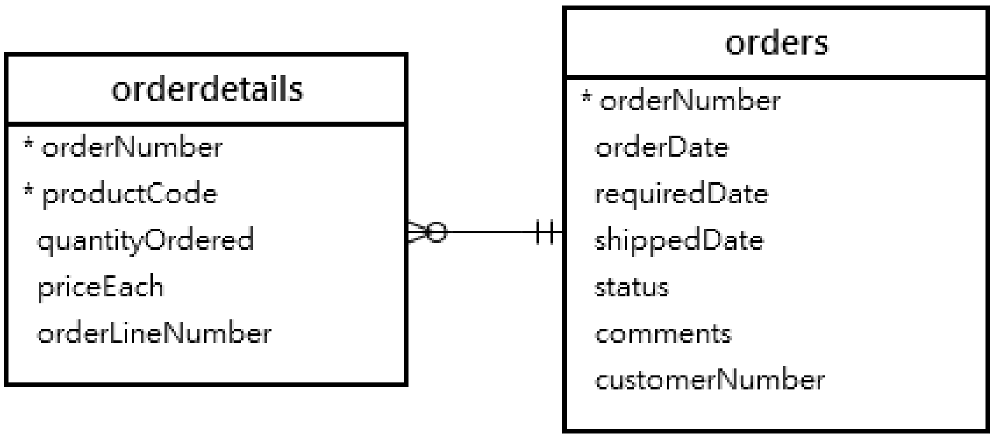
Questions you can answer for submission:

Technical: What is the command for? Why would you use the command?

Knowledge: Would a business user understand this data?

INNER JOIN, GROUP BY, and SUM

This query returns order number, order status and total sales from the *orders* and *orderdetails* tables using the INNER JOIN clause with the GROUP BY clause:



Source: https://www.mysqltutorial.org/mysql-inner-join.aspx

Extract records that matches in both tables and group by

```
SELECT t1.orderNumber, t1.status, SUM(quantityOrdered * priceEach) total FROM orders t1 INNER JOIN orderdetails t2
ON t1.orderNumber = t2.orderNumber GROUP BY orderNumber;
```

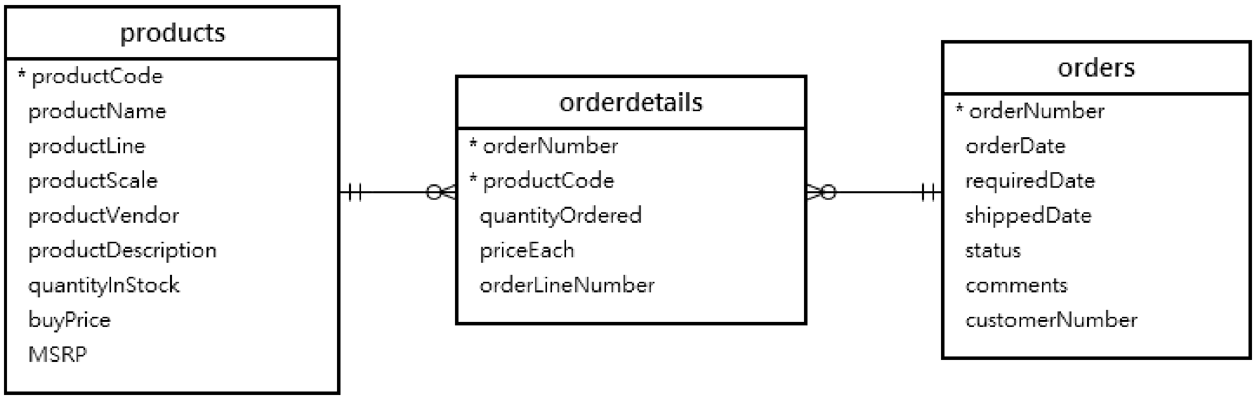
```
+-------------+------------+----------+
| orderNumber | status     | total    |
+-------------+------------+----------+
|       10100 | Shipped    | 10223.83 |
|       10101 | Shipped    | 10549.01 |
|       10102 | Shipped    |  5494.78 |
|       10103 | Shipped    | 50218.95 |
|       10104 | Shipped    | 40206.20 |
```

Questions you can answer for submission:

Technical: What is the command for? Why would you use the command?

Knowledge: Would a business user understand this data?

INNER JOIN 3 Tables



Source: https://www.mysqltutorial.org/mysql-inner-join.aspx

Join 3 tables

SELECT orderNumber, orderDate, orderLineNumber, productName, quantityOrdered, priceEach FROM orders INNER JOIN orderdetails USING (orderNumber) INNER JOIN products USING (productCode) ORDER BY orderNumber, orderLineNumber;

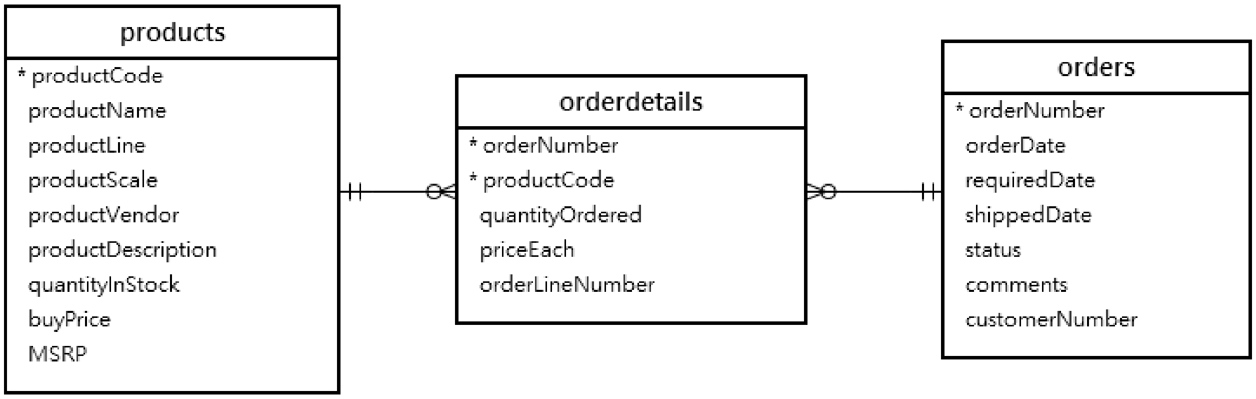
```
+-------------+------------+-----------------+--------------------------------------------+-----------------+-----------+
| orderNumber | orderDate  | orderLineNumber | productName                                | quantityOrdered | priceEach |
+-------------+------------+-----------------+--------------------------------------------+-----------------+-----------+
|       10100 | 2003-01-06 |               1 | 1936 Mercedes Benz 500k Roadster           |              49 |     35.29 |
|       10100 | 2003-01-06 |               2 | 1911 Ford Town Car                         |              50 |     55.09 |
|       10100 | 2003-01-06 |               3 | 1917 Grand Touring Sedan                   |              30 |    136.00 |
|       10100 | 2003-01-06 |               4 | 1932 Alfa Romeo 8C2300 Spider Sport        |              22 |     75.46 |
|       10101 | 2003-01-09 |               1 | 1928 Mercedes-Benz SSK                     |              26 |    167.06 |
```

Questions you can answer for submission:

Technical: What is the command for? Why would you use the command?

Knowledge: Would a business user understand this data?

INNER JOIN using operators



Source: https://www.mysqltutorial.org/mysql-inner-join.aspx

So far, you have seen that the join condition used the equal operator (=) for matching rows.

In addition to the equal operator (=), you can use other operators such as greater than (>), less than (<), and not-equal (<>) operator to form the join condition.
The following query uses a less-than (<) join to find sales price of the product whose code is S10_1678 that is less than the manufacturer's suggested retail price (MSRP) for that product.

```
SELECT orderNumber, productName, msrp, priceEach FROM products p INNER JOIN orderdetails o ON p.productcode =
o.productcode AND p.msrp > o.priceEach WHERE p.productcode = 'S10_1678';
```

```
+-------------+---------------------------------------+-------+-----------+
| orderNumber | productName                           | msrp  | priceEach |
+-------------+---------------------------------------+-------+-----------+
|       10107 | 1969 Harley Davidson Ultimate Chopper | 95.70 |     81.35 |
|       10121 | 1969 Harley Davidson Ultimate Chopper | 95.70 |     86.13 |
|       10134 | 1969 Harley Davidson Ultimate Chopper | 95.70 |     90.92 |
|       10145 | 1969 Harley Davidson Ultimate Chopper | 95.70 |     76.56 |
|       10159 | 1969 Harley Davidson Ultimate Chopper | 95.70 |     81.35 |
```

Questions you can answer for submission:

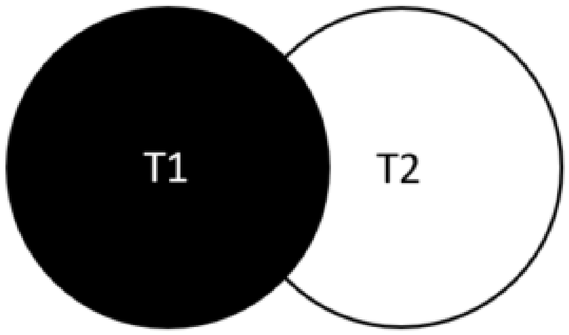
Technical: What is the command for? Why would you use the command?

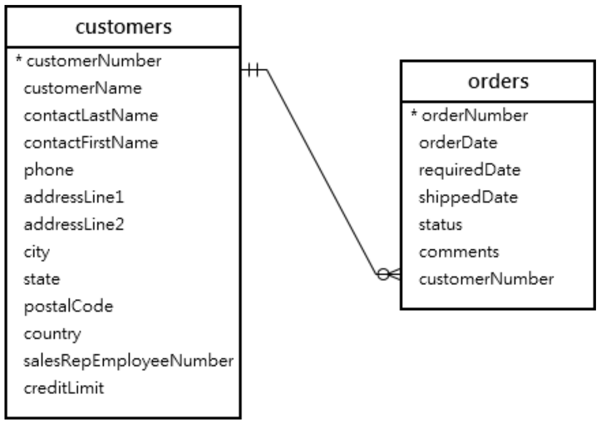
Knowledge: Would a business user understand this data?

LEFT JOIN

The LEFT JOIN allows you to query data from two or more tables. Similar to the INNER JOIN clause, the LEFT JOIN is an optional clause of the SELECT statement, which appears immediately after the FROM clause.
Suppose that you want to join two tables t1 and t2.

Syntax: SELECT *column/s* FROM *table1* LEFT JOIN *table2* ON *join_condition*;





Source: https://www.mysqltutorial.org/mysql-left-join.aspx

```
SELECT customers.customerNumber, customerName, orderNumber, status FROM customers LEFT JOIN orders ON
orders.customerNumber = customers.customerNumber;
```

```
+----------------+------------------------------------+-------------+------------+
| customerNumber | customerName                       | orderNumber | status     |
+----------------+------------------------------------+-------------+------------+
|            103 | Atelier graphique                  |       10123 | Shipped    |
|            103 | Atelier graphique                  |       10298 | Shipped    |
|            103 | Atelier graphique                  |       10345 | Shipped    |
|            112 | Signal Gift Stores                 |       10124 | Shipped    |
```

In this example:

- The customers is the left table and orders is the right table.
- The LEFT JOIN clause returns all customers including the customers who have no order. If a customer has no order, the values in the column orderNumber and status are NULL.

Questions you can answer for submission:

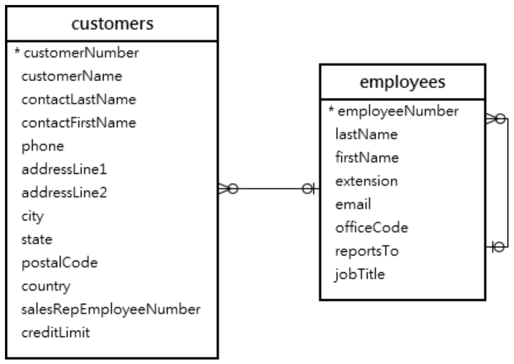
Technical: What is the command for? Why would you use the command?

Knowledge: Would a business user understand this data?

RIGHT JOIN

Suppose that you want to join two tables t1 and t2. MySQL RIGHT JOIN is similar to LEFT JOIN, except that the treatment of the joined tables is reversed.

Syntax: SELECT *column/s* FROM *table1* RIGHT JOIN *table2* ON *join_condition*;



Source: https://www.mysqltutorial.org/mysql-right-join/

```
SELECT employeeNumber, customerNumber FROM customers RIGHT JOIN employees ON salesRepEmployeeNumber =
employeeNumber ORDER BY employeeNumber;
```

```
+----------------+----------------+
| employeeNumber | customerNumber |
+----------------+----------------+
|           1002 |           NULL |
|           1056 |           NULL |
|           1076 |           NULL |
|           1088 |           NULL |
|           1102 |           NULL |
|           1143 |           NULL |
|           1165 |            124 |
|           1165 |            129 |
```

In this example:

- The RIGHT JOIN returns all rows from the table employees whether rows in the table employees have matching values in the column salesRepEmployeeNumber of the table customers.
- If a row from the table employees has no matching row from the table customers , the RIGHT JOIN uses NULL for the customerNumber column.

Questions you can answer for submission:

Technical: What is the command for? Why would you use the command?

Knowledge: Would a business user understand this data?