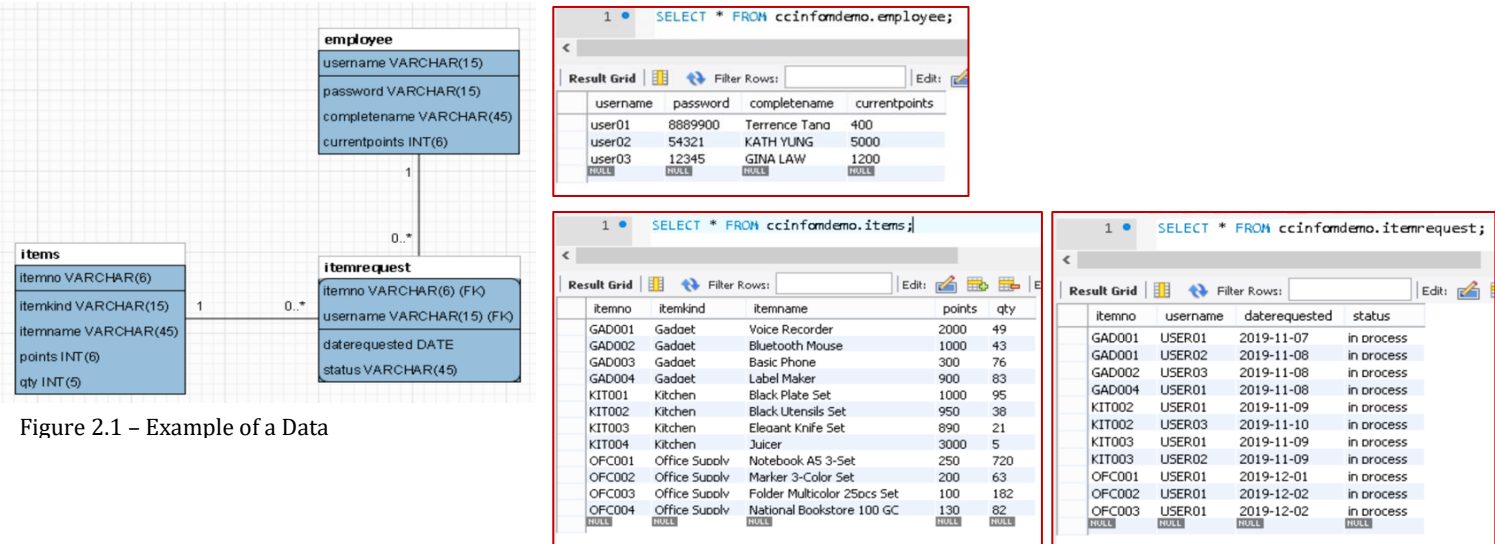


Requirements for this lesson

1. Installed MYSQL and MYSQL Workbench
2. Downloaded MYSQL Script File (ccinfomdemo.sql and dbworld.sql)

Lesson 5: Writing SQL SELECT Statements using INNER JOINS

For this lesson, we will be using the database described below.



To better understand how SQL statements using INNER JOINS are written correctly, let have this sample information requirement.

Generate the list of items (item no, item name) that is more than 1000 points and were requested by USER02.

Before answering the requirement using SQL SELECT Statement, identify and organize what we will be needing

(a). **Let's identify the data we need to fulfill the requirement.**

1. We need *username* from ITEMREQUEST since it contains the username of the employee that requested something
2. We need *points* from ITEMS since it contains the points of an item
3. We need *itemno* and *itemname* from ITEMS since the resulting information requires those

From the list of data we need to fulfill the information requirement, we can see that we need data from two tables – ITEMS and ITEMREQUEST. What is the combining condition necessary to combine the records from ITEMS and ITEMREQUEST, check the foreign key involved. And that is, itemno.

(b). **What ways of using data from two tables are we going to need?** We are not using cartesian product as already explained in the previous lesson. We are left with the following:

1. Are we going to use INNER JOIN? YES, since we need records in ITEMREQUEST that is related to ITEMS
2. Are we going to use LEFT JOIN? NO, since the expected result does not need all in ITEMS, nor all in ITEMREQUEST
3. Are we going to use UNION? NO, since ITEMS and ITEMREQUEST are not compatible tables

(c). **What conditions in the data we need to have?** Based on the requirements, we need the following:

1. username = 'USER02'
2. points > 1000

After gathering everything that you need before writing the SQL SELECT statement, we can now write the SQL Statement.

1. Write the FROM Clause **FROM items i JOIN itemrequest ir ON i.itemno = ir.itemno**
2. Write the WHERE Clause **FROM items i JOIN itemrequest ir ON i.itemno = ir.itemno
WHERE i.points > 1000 AND ir.username = 'USER02'**
3. Write the SELECT Clause **SELECT i.itemno, i.itemname
FROM items i JOIN itemrequest ir ON i.itemno = ir.itemno
WHERE i.points > 1000 AND ir.username = 'USER02'**
4. Write the ORDER BY Clause **SELECT i.itemno, i.itemname
FROM items i JOIN itemrequest ir ON i.itemno = ir.itemno
WHERE i.points > 1000 AND ir.username = 'USER02'**



ORDER BY i.itemname

That's not that difficult right?

Let's have another example

Generate the list of users (username and complete name) that requested items in November 2019

Before answering the requirement using SQL SELECT Statement, identify and organize what we will be needing

(a). **Let's identify the data we need to fulfill the requirement.**

- a.1. We need *username* and *completename* from EMPLOYEE since it contains the username and complete name of employees.
- a.2. We need *daterequested* from ITEMREQUEST to check if it was requested in November 2019.

From the list of data we need to fulfill the information requirement, we can see that we need data from two tables – EMPLOYEE and ITEMREQUEST

What is the combining condition necessary to combine the records from EMPLOYEE and ITEMREQUEST, check the foreign key involved. And that is, username.

(b). **What ways of using data from two tables are we going to need?** We are not using cartesian product as already explained in the previous lesson.

We are left with the following:

- b.1. Are we going to use INNER JOIN? YES, since we need records in ITEMREQUEST that is related to EMPLOYEE
- b.2. Are we going to use LEFT JOIN? NO, since the expected result does not need all in EMPLOYEE, nor all in ITEMREQUEST
- b.3. Are we going to use UNION? NO, since EMPLOYEE and ITEMREQUEST are not compatible tables

(c). **What conditions in the data we need to have?** Based on the requirements, we need the following:

- c.1. *daterequested* BETWEEN '2019-11-01' AND '2019-11-30'

After gathering everything that you need before writing the SQL SELECT statement, we can now write the SQL Statement.

1. Write the FROM Clause **FROM employee e JOIN itemrequest ir ON e.username = ir.username**
2. Write the WHERE Clause **FROM employee e JOIN itemrequest ir ON e.username = ir.username
WHERE e.daterequested BETWEEN '2019-11-01' AND '2019-11-30'**
3. Write the SELECT Clause **SELECT e.username, e.completename
FROM employee e JOIN itemrequest ir ON e.username = ir.username
WHERE e.daterequested BETWEEN '2019-11-01' AND '2019-11-30'**
4. Write the ORDER BY Clause **SELECT e.username, e.completename
FROM employee e JOIN itemrequest ir ON e.username = ir.username
WHERE daterequested BETWEEN '2019-11-01' AND '2019-11-30'
ORDER BY e.completename**
5. If you will execute the SQL statement we have so far, you will notice duplicates in the result. So it may be best to add DISTINCT in the SELECT clause. Our final SQL Statement will be: **SELECT DISTINCT e.username, e.completename
FROM employee e JOIN itemrequest ir ON e.username = ir.username
WHERE daterequested BETWEEN '2019-11-01' AND '2019-11-30'
ORDER BY e.completename**

Let's have another example

Generate the list of users (username and complete name) that requested gadget items in November 2019, include the name of the gadget the user requested. Before answering the requirement using SQL SELECT Statement, identify and organize what we will be needing

(a). **Let's identify the data we need to fulfill the requirement.**

- a.1. We need *username* and *completename* from EMPLOYEE since it contains the username and complete name of employees.
- a.2. We need *daterequested* from ITEMREQUEST to check if it was requested in November 2019.
- a.3. We need *itemkind* from ITEMS to check if the item is a Gadget
- a.4. We need *itemname* from ITEMS since the name of the gadget is needed in the result.

From the list of data we need to fulfill the information requirement, we can see that we need data from three tables – EMPLOYEE, ITEMREQUEST and ITEMS.

The sequence of combining tables is based on the relationship path in the data model. For example, we cannot combine EMPLOYEE and ITEMS, they are not related in the data model. But we can join EMPLOYEE with ITEMREQUEST, then ITEMREQUEST with ITEMS.

What is the combining condition necessary to combine the records from EMPLOYEE and ITEMREQUEST, check the foreign key involved. It is username.

What is the combining condition necessary to combine the records from ITEMREQUEST and ITEMS, check the foreign key involved. It is itemno.

(b). **What ways of using data from two tables are we going to need?** We will not be using cartesian product as already explained in the previous lesson. We are left with the following:

- b.1. Are we going to use INNER JOIN? YES, since we need records in ITEMREQUEST that is related to EMPLOYEE
YES, since we need records in ITEMREQUEST that is related to ITEMS
- b.2. Are we going to use LEFT JOIN? NO, since the expected result does not need all in EMPLOYEE, nor all in ITEMREQUEST, nor all in ITEMS
- b.3. Are we going to use UNION? NO, since EMPLOYEE, ITEMREQUEST and ITEMS, are not compatible tables

(c). **What conditions in the data we need to have?** Based on the requirements, we need the following:

- c.1. *daterequested* BETWEEN '2019-11-01' AND '2019-11-30'
- c.2. *itemkind* = 'Gadget'

After gathering everything that you need before writing the SQL SELECT statement, we can now write the SQL Statement.

1. Write the FROM Clause

FROM	employee e	JOIN	itemrequest ir	ON e.username = ir.username
		JOIN	items i	ON ir.itemno = i.itemno
2. Write the WHERE Clause

FROM	employee e	JOIN	itemrequest ir	ON e.username = ir.username
		JOIN	items i	ON ir.itemno = i.itemno
WHERE	e.daterequested BETWEEN '2019-11-01' AND '2019-11-30'			
AND	i.itemkind = 'Gadget'			
3. Write the SELECT Clause

SELECT	e.username, e.completename, i.itemname			
FROM	employee e	JOIN	itemrequest ir	ON e.username = ir.username
		JOIN	items i	ON ir.itemno = i.itemno
WHERE	e.daterequested BETWEEN '2019-11-01' AND '2019-11-30'			
AND	i.itemkind = 'Gadget'			
4. Write the ORDER BY Clause

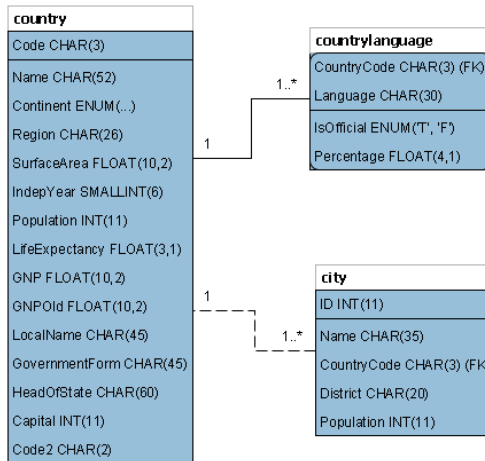
SELECT	e.username, e.completename, i.itemname			
FROM	employee e	JOIN	itemrequest ir	ON e.username = ir.username
		JOIN	items i	ON ir.itemno = i.itemno
WHERE	daterequested BETWEEN '2019-11-01' AND '2019-11-30'			
AND	i.itemkind = 'Gadget'			
ORDER BY	e.completename, i.itemname			

- The ORDER BY clause above will sort the result by complete name, and if ever there are two or more same complete names, it will sort it by the itemname. Check and compare the result of the SQL statement without the itemname in the ORDER By clause to that with the itemname in the ORDER By clause.

5. If you will execute the SQL statement we have so far, you will notice NO duplicates in the result. So we do not need to add DISTINCT in the SELECT clause.

EXERCISE:

Given the database below, write the SQL statement necessary to fulfill the information requirements below. Use DBWorld.sql if you have not created the DBWorld database. You may want to write your answer in a text file and save it using <section>-<lastname>-<firstname>-M2L05.sql. Just be ready with the file, in case your teacher will collect it for formative assessment. Prepare your questions and clarifications as these are important indicators that you went through this exercise.



1. Generate the list of countries (country name and continent) that uses English as the official language
2. Generate the list of official languages used by countries in Africa
3. Generate the list of cities (city name and country name) in North America with a population of more than 1 million
4. Can you think of a requirement that will join all the tables in this database? Write the requirement and its corresponding SQL Statement