CCINFOM

Module 2 Lesson 06: Queries involving Outer Join

Estimated Time to consume material – 6.0 Hours

Requirements for this lesson

1. Installed MYSQL and MYSQL Workbench
2. Downloaded MYSQL Script File (dbsimple.sql and dbemployees.sql)

Lesson 6: Queries involving Outer Join

For this lesson, we will be using the database described in the data model below. The database can be created (if you have not created it yet) using the script file dbsimple.sql.

Fruits

fruitcode	description	qty
APPLE	Red Fruit	20
BANANA	Yellow Fruit	10
MANGO	Yelow Fruit	10
TOMATO	Red Fruit	20

Students

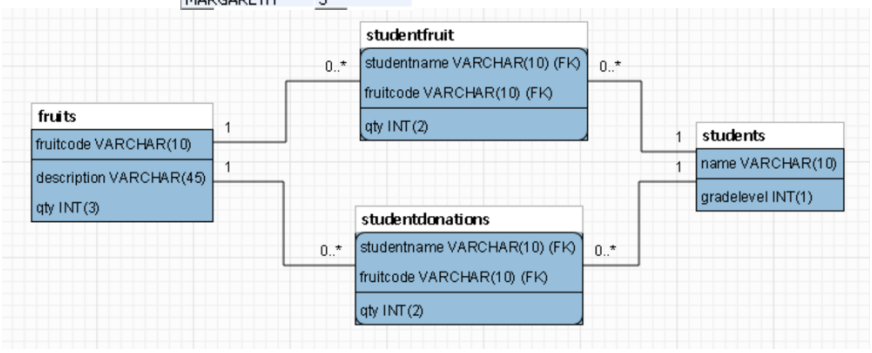
name	gradelevel
ALEXANDRA	2
ANGELA	6
CHRISTIAN	4
GERONE	2
JEHOVAH	5
LEON	5
LISSA	4
MARGARETH	3

StudentDonation

studentname	fruitcode	qty
ANGELA	MANGO	4
GERONE	BANANA	10
LEON	APPLE	4
LISSA	APPLE	4

StudentFruits

studentname	fruitcode	qty
ANGELA	MANGO	3
JEHOVAH	APPLE	2
LEON	APPLE	3



As explained in Lesson 4, another way of using data from two or more tables is by using an Outer Join or in SQL – Left Join. An outer join, in general sense, is taking all the records of one table regardless whether they have a matching record/s on another table. In this lesson, we will be taking some other examples of LEFT JOIN and the clues that will tell us that a LEFT JOIN is needed.

Take a look at the list of the requirements below. All of the requirements in the first column, will use a LEFT JOIN.
Investigate, what is common to the requirements that gives you a clue that a LEFT JOIN is needed.

Will use LEFT JOIN	Will use a JOIN
1. List all the student, and the donations (if applicable) they gave. 2. List all the students, and the fruits donated to them (if they received fruits). 3. List all the fruits (regardless whether donated or not), the student-donors and the quantity the donor donated. 4. List all the fruits, and the student that received the fruit (if there is a student that received the fruit)	1. List all the student and the fruit donations they gave 2. List all the students and the fruits donated to them 3. List all the fruits, the student-donor and the quantity the donor donated 4. List all the fruits acquired by students, and the student that received the fruit

From the examples of requirements above, if the requirement indicates phrases that suggests that regardless if the relationship of a record exists with another record exists or not, then that's a clue for the use of LEFT JOIN. Just take example #2, the phrase "if they received fruits" suggests that the list will show ALL the STUDENTS, and should show the fruits donated to the student if they received fruits, and will not show it if the student did not receive any donations. Unlike in example #2 under JOIN, there is no phrase suggesting that ALL students need to be in the result regardless whether they received donations or not.

Let's now write the SQL SELECT Statements to fulfill the requirements.

List all the student, and the donations (if applicable) they gave
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 *all student data* from STUDENTS
a.2. We need *fruitcode, and qty* from STUDENTDONATIONS

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

(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? NO, since there is a clue that all records of STUDENTS are needed in the result regardless if there is a related donation record
b.2. Are we going to use LEFT JOIN? YES
b.3. Are we going to use UNION? NO, since STUDENTS and STUDENTDONATIONS are not compatible tables

(c). **What conditions in the data we need to have?** Based on the requirements, we need NONE

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 students s LEFT JOINstudentdonations sd ON s.name=sd.studentname
```
2. Write the WHERE Clause
```
FROM students s LEFT JOINstudentdonations sd ON s.name=sd.studentname
```
3. Write the SELECT Clause
```
SELECT s.name, s.gradelevel, sd.fruitcode, sd.qty
FROM students s LEFT JOINstudentdonations sd ON s.name=sd.studentname
```
4. Write the ORDER BY Clause
```
SELECT s.name, s.gradelevel, sd.fruitcode, sd.qty
FROM students s LEFT JOINstudentdonations sd ON s.name=sd.studentname
ORDER BY s.name
```

Executing our query will result to:

name	gradelevel	fruitcode	qty
ALEXANDRA	2	NULL	NULL
ANGELA	6	MANGO	4
CHRISTIAN	4	NULL	NULL
GERONE	2	BANANA	10
JEHOVAH	5	NULL	NULL
LEON	5	APPLE	4
LISSA	4	APPLE	4
MARGARETH	3	NULL	NULL

Let's answer requirement #2
List all the students, and the fruits donated to them (if they received fruits).
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 *all student data* from STUDENTS
a.2. We need *fruitcode, and qty* from STUDENTFRUIT

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

(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? NO, since there is a clue that all records of STUDENTS are needed in the result regardless if there is a related receiving of fruit record
b.2. Are we going to use LEFT JOIN? YES
b.3. Are we going to use UNION? NO, since STUDENTS and STUDENTFRUIT are not compatible tables

(c). **What conditions in the data we need to have?** Based on the requirements, we need NONE

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 students s LEFT JOINstudentfruit sf ON s.name=sf.studentname
```
2. Write the WHERE Clause
```
FROM students s LEFT JOINstudentfruit sf ON s.name=sf.studentname
```
3. Write the SELECT Clause
```
SELECT s.name, s.gradelevel, sf.fruitcode, sf.qty
FROM students s LEFT JOINstudentfruit sf ON s.name=sf.studentname
```
4. Write the ORDER BY Clause
```
SELECT s.name, s.gradelevel, sf.fruitcode, sf.qty
FROM students s LEFT JOINstudentfruit sf ON s.name=sf.studentname
ORDER BY s.name
```

Executing our query will result to:

name	gradelevel	fruitcode	qty
ALEXANDRA	2	NULL	NULL
ANGELA	6	MANGO	3
CHRISTIAN	4	NULL	NULL
GERONE	2	NULL	NULL
JEHOVAH	5	APPLE	2
LEON	5	APPLE	3
LISSA	4	NULL	NULL
MARGARETH	3	NULL	NULL

Before we go to the 3rd example, add a record in STUDENTFRUIT by executing this SQL Statement - INSERT INTO studentfruit VALUES ('ANGELA','BANANA', 10)
Then, execute the SQL SELECT Statement for our example #2. What did you observe in the result?

name	gradelevel	fruitcode	qty
ALEXANDRA	2	NULL	NULL
ANGELA	6	BANANA	10
ANGELA	6	MANGO	3
CHRISTIAN	4	NULL	NULL
GERONE	2	NULL	NULL
JEHOVAH	5	APPLE	2
LEON	5	APPLE	3
LISSA	4	NULL	NULL
MARGARETH	3	NULL	NULL

There will be two records involving ANGELA. That's correct, since there are two records of her receiving fruits – a Banana, and a Mango. But the students that did not receive fruits are still in the result.

Let's answer requirement #3
List all the fruits (regardless whether donated or not), the student-donors and the quantity the donor donated.
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 *all fruit data* from FRUITS
a.2. We need *studentname, and qty* from STUDENTDONATIONS

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

(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? NO, since there is a clue that all records of FRUITS are needed in the result regardless if there are related donations
b.2. Are we going to use LEFT JOIN? YES
b.3. Are we going to use UNION? NO, since FRUITS and STUDENTDONATIONS are not compatible tables

(c). **What conditions in the data we need to have?** Based on the requirements, we need NONE

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 fruits f LEFT JOINstudentdonations sd ON f.fruitcode=sd.fruitcode
```

2. Write the WHERE Clause
```
FROM fruits f LEFT JOINstudentdonations sd ON f.fruitcode=sd.fruitcode
```

3. Write the SELECT Clause
```
SELECT f.fruitcode, f.description, f.qty as FRUITQTY,
       sd.studentname, sd.qty as DONATIONQTY
FROM fruits f LEFT JOINstudentdonations sd ON f.fruitcode=sd.fruitcode
```

- *Note that you need to have an alias for f.qty and sd.qty since they have the same column name*

4. Write the ORDER BY Clause
```
SELECT f.fruitcode, f.description, f.qty as FRUITQTY,
       sd.studentname, sd.qty as DONATIONQTY
FROM fruits f LEFT JOINstudentdonations sd ON f.fruitcode=sd.fruitcode
ORDER BY f.description
```

Executing our query will result to:

fruitcode	description	FRUITQTY	studentname	DONATIONQTY
APPLE	Red Fruit	20	LEON	4
APPLE	Red Fruit	20	LISSA	4
TOMATO	Red Fruit	20	NULL	NULL
BANANA	Yellow Fruit	10	GERONE	10
MANGO	Yelow Fruit	10	ANGELA	4

Let's answer requirement #4

List all the fruits, and the student that received the fruit (if there is a student that received the fruit)
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 *all fruit data* from FRUITS
a.2. We need *studentname, and qty* from STUDENTFRUITS

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

(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? NO, since there is a clue that all records of FRUITS are needed in the result regardless if there are related records in STUDENTFRUITS
b.2. Are we going to use LEFT JOIN? YES
b.3. Are we going to use UNION? NO, since FRUITS and STUDENTFRUITS are not compatible tables

(c). **What conditions in the data we need to have?** Based on the requirements, we need NONE

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 fruits f LEFT JOINstudentfruit sf ON f.fruitcode=sf.fruitcode
```

2. Write the WHERE Clause
```
FROM fruits f LEFT JOINstudentfruit sf ON f.fruitcode=sf.fruitcode
```

3. Write the SELECT Clause
```
SELECT f.fruitcode, f.description, f.qty as FRUITQTY,
sf.studentname, sf.qty as DONATIONQTY
FROM fruits f LEFT JOINstudentfruit sf ON f.fruitcode=sf.fruitcode
```

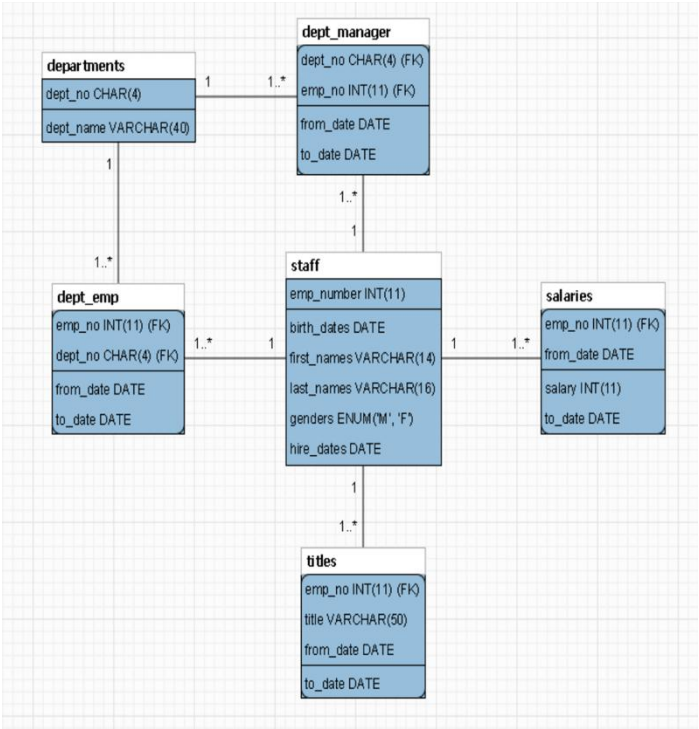
- *Note that you need to have an alias for f.qty and sd.qty since they have the same column name*

4. Write the ORDER BY Clause
```
SELECT f.fruitcode, f.description, f.qty as FRUITQTY,
sf.studentname, sf.qty as DONATIONQTY
FROM fruits f LEFT JOINstudentfruit sf ON f.fruitcode=sf.fruitcode
ORDER BY f.description
```

Executing our query will result to:

fruitcode	description	FRUITQTY	studentname	DONATIONQTY
APPLE	Red Fruit	20	JEHOVAH	2
APPLE	Red Fruit	20	LEON	3
TOMATO	Red Fruit	20	NULL	NULL
BANANA	Yellow Fruit	10	ANGELA	10
MANGO	Yelow Fruit	10	ANGELA	3

EXERCISE:
Given the database below, write the SQL statement necessary to fulfill the information requirements below.
The script file to create the database below is DBEmployees.sql. You do not need to wait until its completely loaded, just download the file. The script file is around 162MB. This is a BIG script, so do not be surprised if it takes time to be imported using MYSQL Workbench to create the DB. You may want to write your answer in a text file and save it using <section>-<lastname>-<firstname>-M2L06.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. List all the employees, and the department currently being or previously managed and the date of assignment as manager (if they manage a department). Sort the list by the date of assignment as manager (latest assignment first).
2. List all the employees (employee number and complete name), and the titles they were assigned (if they were assigned titles). Sort the list such that employees without titles appear first and should they have titles, sort if by employee name).
3. List all the departments (department name), and all the current and previous managers (employee number and date of assignment). Sort the list by department name and latest assignment first.