CCINFOM

Module 2 Lesson 04: The thinking discipline in writing SQL SELECT Statements

Estimated Time to consume material – 3.0 Hour

Requirements for this lesson

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

Lesson 4: The thinking discipline in writing SQL SELECT Statements

Writing SQL SELECT statement is a discipline. A discipline that also reflects the correct sequence of thinking to fulfill the information requirements. An SQL SELECT statement is made up of the following basic clauses (there are other clauses in a SELECT statement), these clauses are shown below. The order in which the clause should appear in the statement is the same order as it is listed below.

SELECT clause	What data will appear in the information result
FROM clause	Where will the data be coming from
WHERE clause	What conditions will be used to filter the data
ORDER BY clause	What fields will the information be sorted on

Table 3.1 – Sequence of Clauses in a SELECT Statement

The order though on which it is written is different. Most students when writing SQL statements, write the SELECT clause first. If this is their style of writing, so be it.
But it is not the recommended order to write SQL statements. For example. the FROM clause is written first before all other clauses. The order of writing clauses also reflects the recommended sequence of thinking to fulfill the information requirements:

1. Write the FROM clause - Identify first, where will the data you need will come from
2. Write the WHERE clause - Identify next, what conditions you need to filter the data.
3. Write the SELECT clause - Identify next, the fields needed to represent the information requirement
4. Write the ODER BY clause - Identify next, what fields in the SELECT clause, you want the information to be sorted on

Even if the sequence of writing clauses is like what is shown above, the resulting SELECT statement, the clauses should still be in the order as shown in table 3.1.
Given the database described below, let's have a single table information requirement as an example.

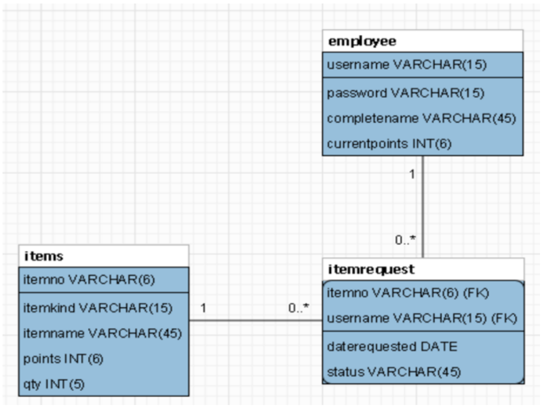
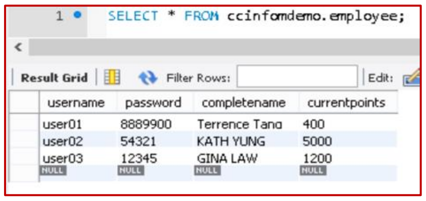
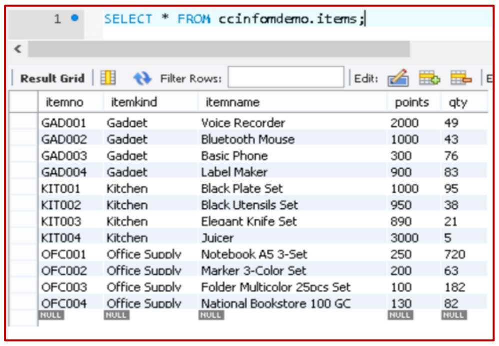


Figure 2.1 – Example of a Data



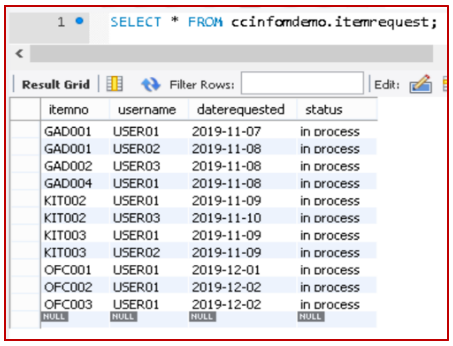
```
SELECT * FROM ccinfomdemo.employee;
```

username	password	completename	currentpoints
user01	8889900	Terrence Tang	400
user02	54321	KATH YUNG	5000
user03	12345	GINA LAW	1200
NULL	NULL	NULL	NULL



```
SELECT * FROM ccinfomdemo.items;
```

itemno	itemkind	itemname	points	qty
GAD001	Gadget	Voice Recorder	2000	49
GAD002	Gadget	Bluetooth Mouse	1000	43
GAD003	Gadget	Basic Phone	300	76
GAD004	Gadget	Label Maker	900	83
KIT001	Kitchen	Black Plate Set	1000	95
KIT002	Kitchen	Black Utensils Set	950	38
KIT003	Kitchen	Elegant Knife Set	890	21
KIT004	Kitchen	Juicer	3000	5
OFC001	Office Supply	Notebook A5 3-Set	250	720
OFC002	Office Supply	Marker 3-Color Set	200	63
OFC003	Office Supply	Folder Multicolor 25pcs Set	100	182
OFC004	Office Supply	National Bookstore 100 GC	130	82
NULL	NULL	NULL	NULL	NULL



```
SELECT * FROM ccinfomdemo.itemrequest;
```

itemno	username	daterequested	status
GAD001	USER01	2019-11-07	in process
GAD001	USER02	2019-11-08	in process
GAD002	USER03	2019-11-08	in process
GAD004	USER01	2019-11-08	in process
KIT002	USER01	2019-11-09	in process
KIT002	USER03	2019-11-10	in process
KIT003	USER01	2019-11-09	in process
KIT003	USER02	2019-11-09	in process
OFC001	USER01	2019-12-01	in process
OFC002	USER01	2019-12-02	in process
OFC003	USER01	2019-12-02	in process
NULL	NULL	NULL	NULL

(a) Generate the list of items (item no, item name) that are more than 1000 points

a.1. With an SQL SELECT writing discipline, the FROM clause will be written first, indicating what table, the data we need will come from. For this example, it will come from ITEMS, and we write:

```
FROM    items
```

a.2. Next, the WHERE clause will be written, indicating what condition/s we want to filter the data. For this example, we are only interested on items with points > 1000. We add to what we already have:

```
FROM    items
WHERE   points > 1000
```

a.3. Next, the SELECT clause will be written, indicating the fields of the resulting information. For this example, we wanted itemno and itemname in the result. We add to what we already have:

```
SELECT  itemno, itemname
FROM    items
WHERE   points > 1000
```

a.4. And lastly, the ORDER BY clause will be written, indicating what fields in the SELECT clause we want the information to be sorted on. For this example, since the requirement did not state how it should be sorted, we have to use our best judgement. Since the information will be a list of items, it might be best to sort it by its name, for this we add:

```
SELECT    itemno, itemname
FROM      items
WHERE     points > 1000
ORDER BY  itemname
```

Let's have another example:

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

b.1. With an SQL SELECT writing discipline, the FROM clause will be written first, indicating what table/s, the data we need will come from. For this example, it will not just come from ITEMS, but the data we need will also need records from ITEMREQUEST (since we need the user that requested the item). And so we write:

```
FROM    items i    JOIN    itemrequest ir    ON i.itemno=ir.itemno
```

Notice that we used a table alias to simplify our SQL Statement – i for items and ir for itemrequest.

b.2. Next, the WHERE clause will be written, indicating what conditions we want to filter the data. For this example, we are only interested on items with points > 1000, and those requested by USER02. And so we write:

```
FROM     items i    JOIN    itemrequest ir    ON i.itemno=ir.itemno
WHERE    i.points > 1000 AND ir.username='USER02'
```

b.3. Next, the SELECT clause will be written, indicating the fields of the resulting information. For this example, we wanted itemno and itemname in the result, and we add to what we already have:

```
SELECT   i.itemno, i.itemname
FROM     items i    JOIN    itemrequest ir    ON i.itemno=ir.itemno
WHERE    i.points > 1000 AND ir.username='USER02'
```

b.4. And lastly, the ORDER BY clause will be written, indicating what fields in the SELECT clause we want the information to be sorted on. For this example, since the requirement did not state how it should be sorted, we have to use our best judgement. Since the information will be a list of items, it might be best to sort it by its name, for this we add:

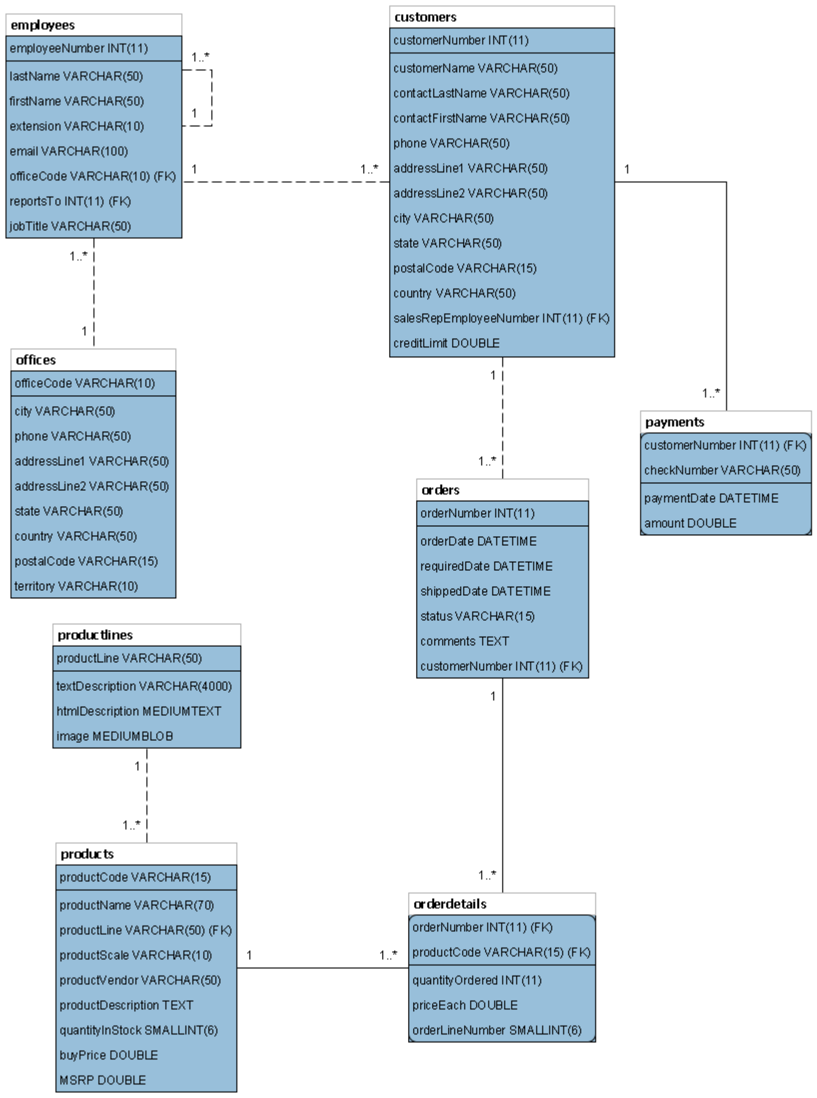
```
SELECT    itemno, itemname
FROM      items i    JOIN    itemrequest ir    ON i.itemno=ir.itemno
WHERE     i.points > 1000 AND ir.username='USER02'
ORDER BY  itemname;
```

CCINFOM

Module 2 Lesson 04: The thinking discipline in writing SQL SELECT Statements

Estimated Time to consume material – 3.0 Hour

EXERCISE:
Try the following information requirements using the dbsales database. To create the dbsales database, use dbsales.sql in MYSQL Workbench. You may want to write your answer in a text file and save it using <section>-<lastname>-<firstname>-M2L04.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 a report showing the customers (customer number and complete name), the country they are located and the complete name of the sales representative handling her/him.
2. Generate the list of orders completed in May 2005 showing the order number, the order and shipped date and the complete name of the customer that made the order.
3. Generate the list of products below 300 pieces. Show in the list the product's code and name, and the product line the product belongs to
4. Generate the complete directory of employees, showing their complete name, email, extension number, the office code they belong to as well as the office phone number.