LAB02: Subquery
_____

Submission:

    - Submit a lab file named "int205_lab02_xxxxxxxxxxx.docx/.pdf" into the LEB2 system. xxxxxxxxxxx = your student id

Due Date & Time:

    - Lecturer will inform the LAB02 due date and time in lab class.

_____

Guideline for writing sub-query

- The subquery (inner query) executes once before the main query (outer query).
- The result of the subquery is used by the main query.
- Enclose subqueries in parentheses.
- Place subqueries on the right side of the comparison condition.
- Use **single-row operators (>, <, =, <=, >=, <>)** with single-row subqueries (return ONLY one value), and use **multiple-row operators (IN, >ANY, >ALL, <ANY, <ALL)** with multiple-row subqueries (return one or more values).
- A common problem with subqueries occurs when no rows are returned by the subquery. So, the main query also returns no rows.
- The ORDER BY clause in the subquery is NOT needed unless you are performing Top-N analysis.

Type 1 – Nested Subquery

- Database evaluates the whole query in two steps:
  - First, execute the subquery (inner query).
  - Second, use the result of the subquery in the parent statement (outer query).

```
select customername
from customers
where creditlimit > ( select creditlimit
                      from customers
                      where customername = 'Land of Toys Inc.')
```

Type 2 - Correlated Subquery

- Database evaluated once for each row processed by the parent statement.
  - This operation is used when a subquery refers to a column from a table in an outer query.
  - The unqualified columns in the subquery are resolved by looking in the tables named in the inner query and then in the tables named in the outer query.

```
select lastname
from employees e
where exists (select *
              from offices
              where officecode = e.officecode)
```

The Syntax of SELECT statement:
Documentation: https://dev.mysql.com/doc/refman/8.0/en/select.html

<u>Note:</u> The MySQL error code 1064 is a syntax error. This means the reason there's a problem is because MySQL doesn't understand what you're asking it to do.

<u>Switch to SQL Editor</u>
- You should specify the classicmodels database before writing SQL statements using the following command:
       USE db_name;

The USE statement tells MySQL to use the named database as the default (current) database for subsequent statements. This statement requires some privilege for the database or some object within it.

The example of the result screen: Please insert your student id in the first line of the query section

The ER diagram for the classicmodels.
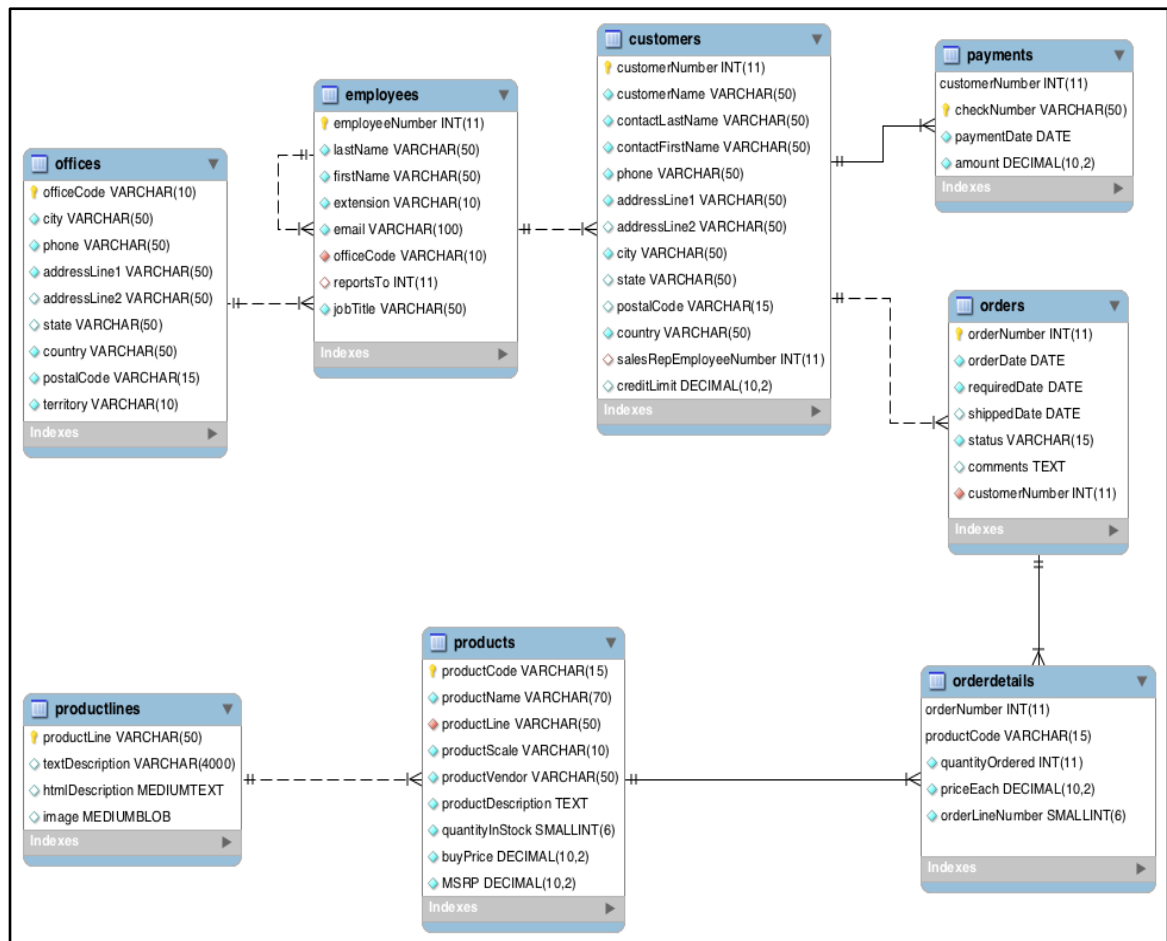
Note: The MSRP is "Manufacturer's suggested retail price" (ราคาขายปลีกแนะนำของผู้ผลิต).

Task 1: Using the "classicmodels" database and write SQL statements to answer the following questions.

3.1 List the customer name of all customers who live in the same country of customer named "Mini Classics". Sort the results in ascending order by customer name.

-- Capture the SQL statement + Result Screen and place here

```
18   select lastname
19   from employees e
20   where exists (select * from offices where officecode = e.officecode);
21
22   -- 3.1 64130500083 List the customer name of all customers who live in the same country of customer named "Mini Classics". Sort the results in ascending order by customer name.
23   select customername
24   from customers
25   where country = (select country from customers where customername = 'Mini Classics');
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content: IA

customername
Corporate Gift Ideas Co.
Tekni Collectables Inc.
Classic Gift Ideas, Inc
Men 'R' US Retailers, Ltd.
Gifts4AllAges.com
Online Diecast Creations Co.
Collectables For Less Inc.
Classic Legends Inc.
Gift Ideas Corp.
The Sharp Gifts Warehouse
Super Scale Inc.
Microscale Inc.
FunGiftIdeas.com
West Coast Collectables Co.
Motor Mint Distributors Inc.
Signal Collectibles Ltd.
Diecast Collectables

customers 5 ×

Output

Action Output

#	Time	Action	Message
✓	4 09:44:33	select firstname from employees where officecode = (select officecode from employees where employeenumber = 1002) LIMIT 0, 1000	6 row(s) returned
✓	5 09:58:49	select customername from customers where creditlimit > (select creditlimit from customers where customername = 'Land of Toys inc.') LIMIT 0, 1000	14 row(s) returned
✗	6 10:03:39	select lastname from employee e where exists (select * from offices where officecode = e.officecode) LIMIT 0, 1000	Error Code: 1146. Table 'classicmodels.employee' doesn't exist
✓	7 10:03:46	select lastname from employees e where exists (select * from offices where officecode = e.officecode) LIMIT 0, 1000	23 row(s) returned
✓	8 10:05:09	select lastname from employees e where not exists (select * from offices where officecode = e.officecode) LIMIT 0, 1000	0 row(s) returned
✓	9 10:11:10	select customername from customers where country = (select country from customers where customername = 'Mini Classics') LIMIT 0, 1000	36 row(s) returned

3.2 List the customer name of all customers who live in the same country of customer named "Mini Classics" and their customer names start with "Mini".

```
27   -- 3.2 64130500083 List the customer name of all customers who live in the same country of customer named "Mini Classics" and their customer names start with "Mini".
28   select customername
29   from customers c
30   where country in (select country from customers where customername = ('Mini Classics'))
31   and customername like 'Mini%';
```

esult Grid | Filter Rows: | Export: | Wrap Cell Content: IA

customername
Mini Gifts Distributors Ltd.
Mini Wheels Co.
Mini Classics
Mini Creations Ltd.

3.3 List the product name and quantity in stock of the product that has the maximum quantities in stock.

```
33   -- 3.3 64130500083 List the product name and quantity in stock of the product that has the maximum quantities in stock.
34   select productname,quantityinstock
35   from products
36   where quantityinstock in (select max(quantityinstock) from products);
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content: IA

productname	quantityinstock
2002 Suzuki XREO	9997

3.4 List the order number and the total amount of sales of all orders that their total amount of sales is more than the total amount of sales order number 10103. Name the total amount of sales column to "total_amount".

```
38      -- 3.4 64130500083 List the order number and the total amount of sales of all orders that their total amount of sales is more than the total amount of sales order number 10103.
39      -- Name the total amount of sales column to "total_amount".
40   •  select ordernumber, sum(quantityordered * priceeach) as total_amount
41      from orderdetails
42      group by ordernumber
43      having total_amount > (select sum(quantityordered * priceeach) from orderdetails where ordernumber = 10103);
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

ordernumber	total_amount
10105	53959.21
10106	52151.81
10108	51001.22
10122	50824.66
10126	57131.92
10127	58841.35
10135	55601.84
10142	56052.56
10145	50342.74
10159	54682.68
10165	67392.85
10168	50743.65
10181	55069.55
10185	52548.49
10192	55425.77
10204	58793.53
10207	59265.14

Result 12 ×

Output

Action Output

#	Time	Action	Message
11	10:39:42	select customername from customers c where country in (select country from customers where customername = ('Mini Classics')) and customername l...	4 row(s) returned
12	10:45:07	select productname,quantityinstock from products where quantityinstock in (select max(quantityinstock) from products) LIMIT 0, 1000	1 row(s) returned
13	10:59:02	select ordernumber, sum(quantityordered * priceeach) as total_amount from orderdetails group by ordernumber having total_amount > (select sum(qu...	0 row(s) returned
14	10:59:19	select ordernumber, sum(quantityordered * priceeach) as total_amount from orderdetails group by ordernumber having total_amount > (select sum(qu...	34 row(s) returned
15	11:01:55	select ordernumber, sum(quantityordered * priceeach) as total_amount from orderdetails having total_amount > (select sum(quantityordered * priceea...	1 row(s) returned
16	11:02:04	select ordernumber, sum(quantityordered * priceeach) as total_amount from orderdetails group by ordernumber having total_amount > (select sum(qu...	34 row(s) returned

3.5 List the customer name and the employee last name of all customers that their sales rep employee worked in the office located in London city. [Write two SQL statements using each type of subquery => Type I: Nested Subquery and Type II: Correlated Subquery subquery]

```
45      -- 3.5 64130500083 List the customer name and the employee last name of all customers that their sales rep employee worked in the office located in London city.
46   •  select c.customername, e.lastname
47      from customers c join employees e on c.salesrepemployeenumber = e.employeenumber
48      where e.officecode in (select officecode from offices where city = 'London');
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

customername	lastname
Toys of Finland, Co.	Bott
AV Stores, Co.	Bott
UK Collectables, Ltd.	Bott
giftsbymail.co.uk	Bott
Oulu Toy Supplies, Inc.	Bott
Stylish Desk Decors, Co.	Bott
Suominen Souveniers	Bott
Double Decker Gift Stores, Ltd	Bott
Baane Mini Imports	Jones
Blauer See Auto, Co.	Jones
Volvo Model Replicas, Co	Jones
Herkku Gifts	Jones
Clover Collections, Co.	Jones
Toms Spezialitäten, Ltd	Jones
Norway Gifts By Mail, Co.	Jones
Bavarian Collectables Imports...	Jones
Scandinavian Gift Ideas	Jones

Result 17 ×

Output

Action Output

#	Time	Action	Message
17	11:04:26	select * from customers LIMIT 0, 1000	122 row(s) returned
18	11:04:51	select * from employees LIMIT 0, 1000	23 row(s) returned
19	11:04:58	select * from customers LIMIT 0, 1000	122 row(s) returned
20	11:05:24	select * from employees LIMIT 0, 1000	23 row(s) returned
21	11:10:05	select c.customername, e.lastname from customers c join employees e on c.salerepemployeenumber = e.employeenumber where e.officecode in (se...	Error Code: 1054. Unknown column 'c.salerepemployeenumber' in 'on clause'
22	11:10:23	select c.customername, e.lastname from customers c join employees e on c.salesrepemployeenumber = e.employeenumber where e.officecode in (s...	17 row(s) returned

3.6 List all cities that are both an office location and a customer location. Remove the duplicate answer. [Write two SQL statements using each type of subquery => Type I: Nested Subquery and Type II: Correlated Subquery subquery]

```
56      -- 3.6 64130500083 List all cities that are both an office location and a customer location. Remove the duplicate answer
57      -- nested
58 •    select distinct city
59      from offices o
60      where o.city in (select city from customers where city = city);
61
62      -- correlated
63 •    select distinct city
64      from offices o
65      where city in (select city from customers where o.city = city);
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

city
San Francisco
Boston
NYC
Paris
London

3.7 List all cities where customers who do not live in the same city of their sales rep employee's office city.   Remove the duplicate answer. [Write two SQL statements using each type of subquery => Type I: Nested Subquery and Type II: Correlated Subquery subquery]

```
66      -- 3.7 64130500083 List all cities where customers who do not live in the same city of their sales rep employee's office city.   Remove the duplicate answer
67      -- nested
68 •    select distinct c.city
69      from customers c, offices o
70      where c.city not in (select c.city from customers c, offices o where c.city = o.city)
71      group by c.city;
72
73      -- correlated
74 •    select distinct c.city
75      from customers c
76      where not exists (select city from offices where c.city = city);
77
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

city
Nantes
Las Vegas
Melbourne
Stavern
San Rafael
Warszawa
Frankfurt
Madrid
Luleå
Kobenhavn
Lyon
Singapore

Result 25

Output

Action Output

#	Time	Action	Message
39	20:09:13	select distinct c.city from customer c where exists (select city from offices where c.city != city) LIMIT 0, 1000	Error Code: 1146. Table 'classicmodels.customer' doesn't exist
40	20:09:17	select distinct c.city from customers c where exists (select city from offices where c.city != city) LIMIT 0, 1000	95 row(s) returned
41	20:09:28	select distinct c.city from customers c where not exists (select city from offices where c.city = city) LIMIT 0, 1000	90 row(s) returned
42	20:10:05	select distinct c.city from customers c, offices o where c.city not in (select c.city from customers c, offices o where c.city = o.city) group by c.city LIMIT 0...	90 row(s) returned

3.8 List the customer name of all customers who have a credit limit greater than all average credit limits of customers in each city.

```
78      -- 3.8 64130500083 List the customer name of all customers who have a credit limit greater than all average credit limits of customers in each city.
79 •    select customername
80      from customers
81      where creditlimit >any (select sum(creditlimit)/count(customernumber) from customers)
82      group by city;
83
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

customername
Signal Gift Stores
Australian Collectors, Co.
La Rochelle Gifts
Baane Mini Imports
Mini Gifts Distributors Ltd.
Land of Toys Inc.
Euro+ Shopping Channel
Danish Wholesale Imports
Saveley & Henriot, Co.
Dragon Souveniers, Ltd.
Diecast Classics Inc.
Technics Stores Inc.

customers 27 ×

Output

Action Output

#	Time	Action	Message
41	20:09:28	select distinct c.city from customers c where not exists (select city from offices where c.city = city) LIMIT 0, 1000	90 row(s) returned
42	20:10:05	select distinct c.city from customers c, offices o where c.city not in (select c.city from customers c, offices o where c.city = o.city) group by c.city LIMIT 0...	90 row(s) returned
43	20:10:29	select distinct c.city from customers c where not exists (select city from offices where c.city = city) LIMIT 0, 1000	90 row(s) returned
44	20:10:43	select customername from customers where creditlimit >any (select sum(creditlimit)/count(customernumber) from customers) group by city LIMIT 0, 1000	58 row(s) returned

3.9 List the customer name of all customers who have a credit limit greater than their average credit limits of customers in their cities.

```
84      -- 3.9 64130500083 List the customer name of all customers who have a credit limit greater than their average credit limits of customers in their cities.
85 •    select c1.customername
86      from customers c1, customers c2
87      where c1.creditlimit >any (select sum(c1.creditlimit)/count(c1.customernumber) and c1.city = c2.city)
88      group by customername;
89
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

customername
Kelly's Gift Shop
Diecast Collectables
Double Decker Gift Stores, Ltd
Signal Collectibles Ltd.
Motor Mint Distributors Inc.
Iberia Gift Imports, Corp.
West Coast Collectables Co.
Frau da Collezione
Australian Collectables, Ltd
FunGiftIdeas.com
Corrida Auto Replicas, Ltd
Microscale Inc.

Result 28 ×

Output

Action Output

#	Time	Action	Message
42	20:10:05	select distinct c.city from customers c, offices o where c.city not in (select c.city from customers c, offices o where c.city = o.city) group by c.city LIMIT 0...	90 row(s) returned
43	20:10:29	select distinct c.city from customers c where not exists (select city from offices where c.city = city) LIMIT 0, 1000	90 row(s) returned
44	20:10:43	select customername from customers where creditlimit >any (select sum(creditlimit)/count(customernumber) from customers) group by city LIMIT 0, 1000	58 row(s) returned
45	20:11:04	select c1.customername from customers c1, customers c2 where c1.creditlimit >any (select sum(c1.creditlimit)/count(c1.customernumber) and c1.city...	98 row(s) returned

3.10 List the products that were never ordered by any customers.

```
90      -- 3.10 64130500083 List the products that were never ordered by any customers.
91 •    select *
92      from products p
93      where not exists (select productcode from orderdetails where p.productcode = productcode);
```

Result Grid | Filter Rows: | Edit: | Export/Import: | Wrap Cell Content:

productCode	productName	productLine	productScale	productVendor	productDescription	quantityInStock	buyPrice	MSRP
S18_3233	1985 Toyota Supra	Classic Cars	1:18	Highway 66 Mini Classics	This model features soft rubber tires, working s...	7733	57.01	107.57
NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL