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

Note: 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.

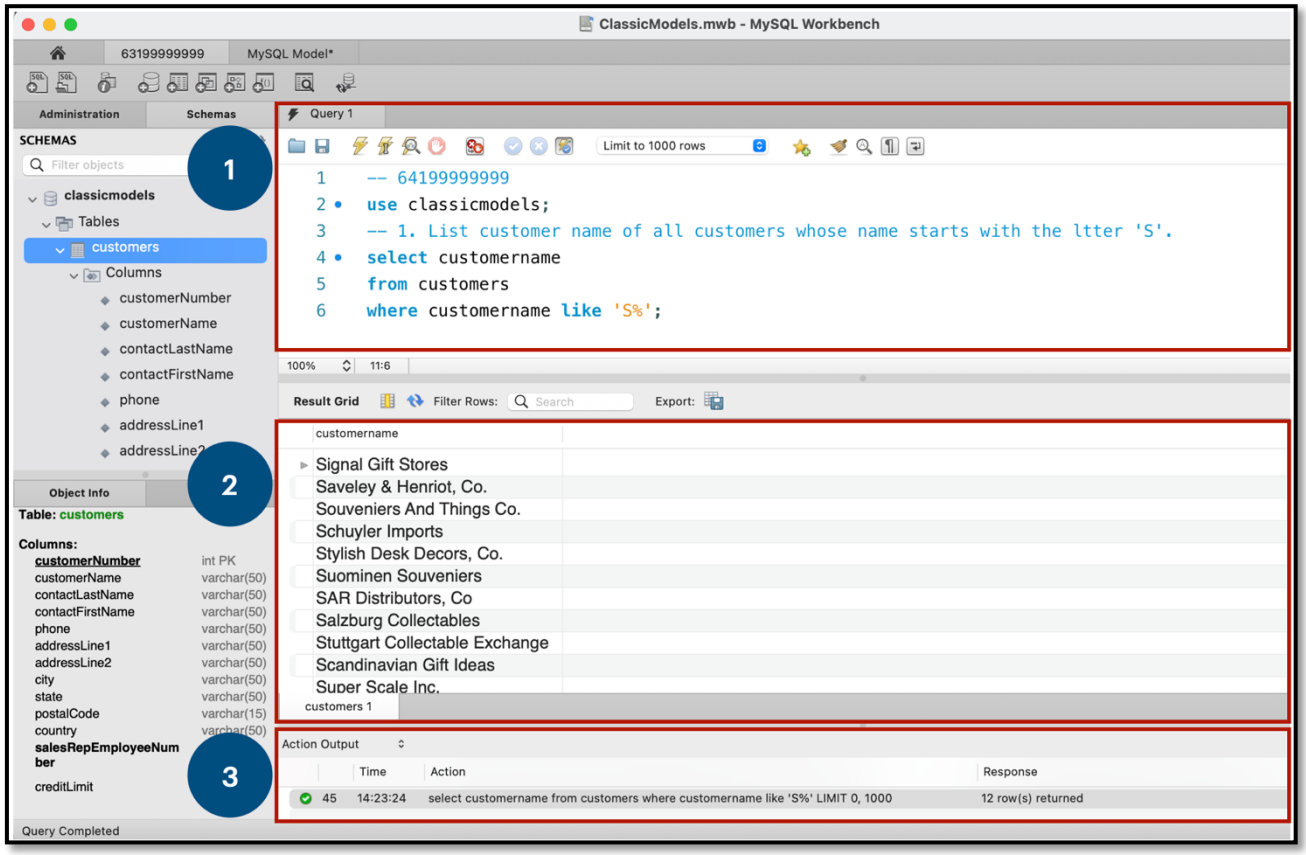
Switch to SQL Editor

- You should specify the classicmodels database before writing SQL statements using the following command:

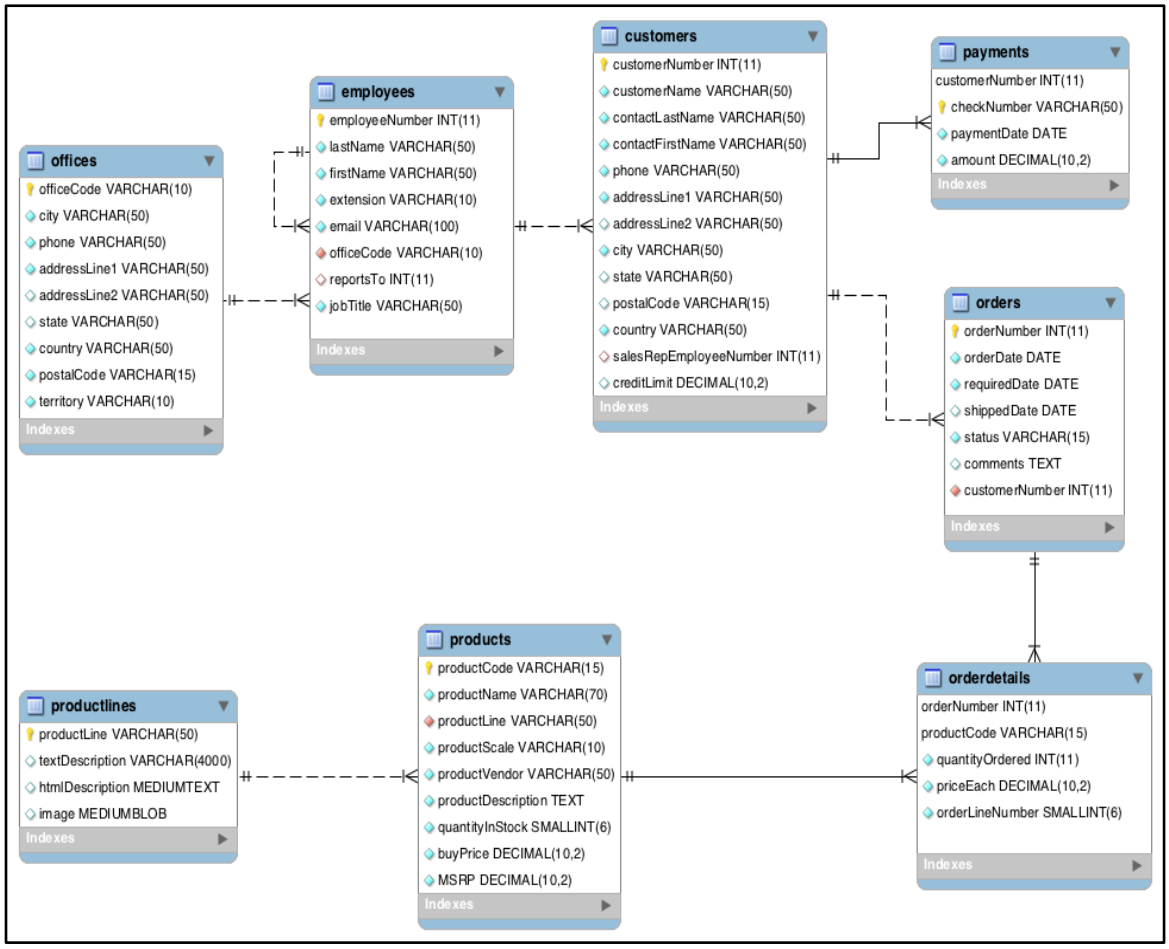
  USE db_name;

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

```
79
80      --  64130500053
81      -- 3.1 List the customer name of all customers who live in the same country of customer named "Mini Classi
82      select customerName
83      from customers
84      where country in ( select country from customers
85      where customerName = "Mini Classics")
86      order by customerName;
87
```

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

customerName
American Souvenirs Inc
Auto-Moto Classics Inc.
Boards & Toys Co.
Cambridge Collectables Co.
Classic Gift Ideas, Inc

customers 7 | Read Only

20 20:39:08 select customerName from customers where country in (select country from custom... 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”.

```
88      -- 64130500053
89      -- 3.2 List the customer name of all customers who live in the same country of customer named "Mini Classi
90      select customerName
91      from customers
92      where country in
93      ( select country from customers
94      where customerName = ("Mini Classics") )
95      and upper( customerName )like 'Mini%'
96      ;
```

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

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

customers 8 | Read Only

22 20:40:38 select customerName from customers where country in (select country from custo... 4 row(s) returned

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

```
97    -- 64130500053
98    -- 3.3 List the product name and quantity in stock of the product that has the maximum quantities in stock
99    select productname,quantityInStock
100   from products
101   where quantityInStock in ( select max(quantityInStock) from products)
102   ;
103   -- 64130500053
104   -- 3.4 List the order number and the total amount of sales of all orders that their total amount of sales
105   select orderNumber, sum(quantityOrdered * priceEach) as total_amount
```

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

productname	quantityInStock
2002 Suzuki XREO	9997

products 10 | Read Only

23 20:41:09 select productname,quantityInStock from products where quantityInStock in (selec... 1 row(s) returned

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”.

```
103   -- 64130500053
104   -- 3.4 List the order number and the total amount of sales of all orders that their total amount of sales
105   select orderNumber, sum(quantityOrdered * priceEach) as total_amount
106   from orderdetails
107   group by orderNumber
108   having sum(quantityOrdered * priceEach) > (select sum(quantityOrdered * priceEach)
109                                              from orderdetails
110                                              where orderNumber = 10103 );
111
```

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

Result 11 | Read Only

24 20:42:42 select orderNumber, sum(quantityOrdered * priceEach) as total_amount from orderd... 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]

```
112     -- 64130500053
113     -- 3.5 List the customer name and the employee last name of all customers that their sales rep employee wc
114     -- type 1
115     select c.customerName ,e.lastname
116     from customers c join employees e
117     on c.salesRepEmployeeNumber = e.employeeNumber
118     where e.officeCode IN ( select officeCode
119                             from offices
120                             where city = 'London');
121     -- type 2
```

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

Result 25 | Read Only

28 20:45:09 select c.customerName ,e.lastname from customers c join employees e on c.salesR... 17 row(s) returned

```
121     -- type 2
122     select c.customername ,e.lastname
123     from customers c join employees e
124     on c.salesRepEmployeeNumber = e.employeeNumber
125     where exists ( select officecode
126                    from offices
127                    where city = 'London'
128                    and e.officecode = officecode);
129
```

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

Result 14 | Read Only

27 20:44:09 select c.customername ,e.lastname from customers c join employees e on c.salesR... 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]

```
129    -- 64130500053
130    -- 3.6 List all cities that are both an office location and a customer location. Remove the duplicate answ
131    -- type1
132    select distinct c.city
133    from customers c inner join employees e on c.salesRepEmployeeNumber = e.employeeNumber
134    where c.city in (select city from offices);
135    -- type2
136    select distinct c.city
137    from customers c inner join employees e on c.salesRepEmployeeNumber = e.employeeNumber
138    where exists (select city from offices where c.city = city);
```

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

city
San Francisco
NYC
Paris
London
Boston

Result 27 | Read Only

29 20:46:35 select distinct c.city from customers c inner join employees e on c.salesRepEmploye... 5 row(s) returned

```
135    -- type2
136    select distinct c.city
137    from customers c inner join employees e on c.salesRepEmployeeNumber = e.employeeNumber
138    where exists (select city from offices where c.city = city);
```

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

city
San Francisco
NYC
Paris
London
Boston

Result 17 | Read Only

32 20:50:18 select distinct c.city from customers c inner join employees e on c.salesRepEmploye... 5 row(s) returned

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]

```
140    -- 64130500053
141    -- 3.7 List all cities where customers who do not live in the same city of their sales rep employee's offi
142    -- type1
143    select distinct c.city
144    from customers c inner join employees e on c.salesRepEmployeeNumber = e.employeeNumber
145    where c.city not in (select city from offices);
146    -- type2
147    select distinct c.city
148    from customers c inner join employees e on c.salesRepEmployeeNumber = e.employeeNumber
```

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

city
Singapore
Kita-ku
Central Hong Kong
Makati City
Minato-ku

Result 20 | Read Only

33 20:54:32 select distinct c.city from customers c inner join employees e on c.salesRepEmploye... 73 row(s) returned

```
146     -- type2
147 •   select distinct c.city
148     from customers c inner join employees e on c.salesRepEmployeeNumber = e.employeeNumber
149     where not exists (select city from offices where c.city = city);
150
151     -- 64130500053
```

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

city
Singapore
Kita-ku
Central Hong Kong
Makati City
Minato-ku

Result 21

34 20:55:37 select distinct c.city from customers c inner join employees e on c.salesRepEmploye... 73 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.

```
151     -- 64130500053
152     -- 3.8 List the customer name of all customers who have a credit limit greater than all average credit lim
153 •   select customername
154     from customers
155     where creditlimit >all (select avg(creditlimit)
156                             from customers
157                             group by city);
158
159     -- 64130500053
160     -- 3.9 List the customer name of all customers who have a credit limit greater than their average credit l
```

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

customername
Euro+ Shopping Channel

Result Grid

customers 22 Read Only

35 20:56:16 select customername from customers where creditlimit >all (select avg(creditlimit) ... 1 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.

```
159     -- 64130500053
160     -- 3.9 List the customer name of all customers who have a credit limit greater than their average credit l
161     select customername
162     from customers c
163     where c.creditlimit > (select avg(creditlimit)
164                            from customers
165                            where city = c.city);
166
167     -- 64130500053
```

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

customername
La Rochelle Gifts
Blauer See Auto, Co.
Land of Toys Inc.
Euro+ Shopping Channel
Dragon Souveniers, Ltd.

customers 23 | Result Grid | Read Only

36 20:57:10 select customername from customers c where c.creditlimit > (select avg(creditlimit) ... 21 row(s) returned

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

```
167     -- 64130500053
168     -- 3.10 List the products that were never ordered by any customers.
169     select productcode, productname
170     from products
171     where productcode not in (select productCode
172                               from orderdetails);
173
174
175
```

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

productcode	productname
S18_3233	1985 Toyota Supra

products 24

37 20:58:08 select productcode, productname from products where productcode not in (select p... 1 row(s) returned