

**LAB01: SQL Review (Solution)**

---

**Submission:**

- Submit a lab file named “int205\_lab01\_xxxxxxxxxxx.docx/.pdf” into the LEB2 system. xxxxxxxxxxxx = your student id

**Due Date & Time:**

- Lecturer will inform the LAB01 due date and time in lab class.
- 

**Task 2: Using the “classicmodels” schema and write SQL statements to answer the following questions.****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;  
    **USE classicmodels;**

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.

2.1 List the product name and quantity in stock of all products that classified in the “Classic Cars” product line (กลุ่มของสินค้า/ผลิตภัณฑ์) and their buy prices are more than 80.

```
select productname, quantityinstock
from products
where productline = 'Classic Cars' and buyprice > 80;
```

The screenshot shows a database management system interface. On the left, there is a 'SCHEMAS' panel with a tree view showing 'classicmodels' and 'Tables'. The 'products' table is selected. The main area displays a query window with the following SQL code:

```
1 -- 63199999999
2 • use classicmodels;
3 -- 2.1 List the product name and quantity in stock of all products that
4 -- classified in the "Classic Cars" product line (กลุ่มของสินค้า/ผลิตภัณฑ์) and
5 -- their buy prices are more than 80.
6 • select productname, quantityinstock
7 from products
8 where productline = 'Classic Cars' and buyprice > 80;
9
```

Below the query window, the 'Result Grid' shows the results of the query. The columns are 'productname' and 'quantityinstock'. The results are as follows:

productname	quantityinstock
1952 Alpine Renault 1300	7305
1972 Alfa Romeo GTA	3252
1962 LanciaA Delta 16V	6791
1968 Ford Mustang	68
2001 Ferrari Enzo	3619
1969 Corvair Monza	6906
1969 Ford Falcon	1049
1993 Mazda RX-7	3975
1995 Honda Civic	9772

At the bottom, the 'Action Output' panel shows the execution details: 46 rows affected, 13:04:41, and 12 row(s) returned.

2.2 List the customer name, city and country of all customers who live in the country named: Japan, Germany or Canada. Sort the results in descending order by country and ascending order by customer name.

-- Ans1

```
select customername, city, country
from customers
where country IN ('Japan','Germany','Canada')
order by country DESC, customername ;
```

-- Ans2

```
select customername, city, country
from customers
where (country = 'Japan' or country = 'Germany' or country = 'Canada')
order by 3 DESC, 1 ;
```

The screenshot shows a database management interface with a query editor and a result grid. The query is as follows:

```

3  -- 2.2 List the customer name, city and country of all customers who live in the country named:
4  -- Japan, Germany or Canada. Sort the results in descending order by country and ascending order by
5  -- customer name.
6  -- Ans1
7  • select customername, city, country
8    from customers
9    where country IN ('Japan','Germany','Canada')
10   order by country DESC, customername ;
11  -- Ans2
12  • select customername, city, country
13    from customers
14    where (country = 'Japan' or country = 'Germany' or country = 'Canada')
15   order by 3 DESC, 1 ;

```

The result grid displays the following data:

customername	city	country
Osaka Souvenirs Co.	Kita-ku	Japan
Tokyo Collectables, Ltd	Minato-ku	Japan
Bavarian Collectables Imports,...	Munich	Germany
Blauer See Auto, Co.	Frankfurt	Germany

The interface also shows a table schema for 'customers' with columns: customerNumber (int PK), customerName (varchar(50)), contactLastName (varchar(50)), contactFirstName (varchar(50)), phone (varchar(50)), addressLine1 (varchar(50)), addressLine2 (varchar(50)), city (varchar(50)), state (varchar(50)), postalCode (varchar(15)), country (varchar(50)), salesRepEmployeeNumber (int), and creditLimit (decimal(10, 2)).

2.3 List the order number and the total amount of sales for all orders. Name the total amount of sales column to “total\_amount”.

```

select ordernumber, sum(quantityordered*priceeach) as total_amount
from orderdetails
group by ordernumber;

```

The screenshot shows a database management interface with a query editor and a result grid. The query is as follows:

```

21  -- 63199999999
22  /*2.3 List the order number and the total amount of sales for all orders.
23  Name the total amount of sales column to “total_amount”.*/
24  • select ordernumber, sum(quantityordered*priceeach) as total_amount
25    from orderdetails
26    group by ordernumber;
27

```

The result grid displays the following data:

ordernumber	total_amount
10100	10223.83
10101	10549.01
10102	5494.78
10103	50218.95
10104	40206.20
10105	53959.21
10106	52151.81
10107	22292.62
10108	51001.22
10109	25833.14
10110	18125.60

The interface also shows a table schema for 'customers' with columns: customerNumber (int PK), customerName (varchar(50)), contactLastName (varchar(50)), contactFirstName (varchar(50)), phone (varchar(50)), addressLine1 (varchar(50)), addressLine2 (varchar(50)), city (varchar(50)), state (varchar(50)), postalCode (varchar(15)), country (varchar(50)), salesRepEmployeeNumber (int), and creditLimit (decimal(10, 2)).

2.4 List the order number and the total amount of sales of all orders that their total amount of sales is more than 55000. Name the total amount of sales column to “total\_amount”.

```
select ordernumber, sum(quantityordered*priceeach) as total_amount
from orderdetails
group by ordernumber
having sum(quantityordered*priceeach) > 55000;
```

The screenshot shows a database management interface with a query editor and a results grid. The query editor contains the following SQL code:

```
-- 6319999999
/*2.4 List the order number and the total amount of sales of all orders that
their total amount of sales is more than 55000. Name the total amount of sales column
to "total_amount".*/
select ordernumber, sum(quantityordered*priceeach) as total_amount
from orderdetails
group by ordernumber
having sum(quantityordered*priceeach) > 55000;
```

The results grid displays the following data:

ordernumber	total_amount
10126	57131.92
10127	58841.35
10135	55601.84
10142	56052.56
10165	67392.85
10181	55069.55
10192	55425.77
10204	58793.53

The interface also shows a schema tree on the left with 'customers' selected, and a table definition for 'customers' on the bottom left.

2.5 List the customer name and the number of sales orders of all customers whose name start with the letter 'D'. Name the number of sales orders column to “num\_orders”.

-- Ans1

```
select customername, count(o.ordernumber) as num_orders
from customers c join orders o
on c.customernumber = o.customernumber
where customername like 'D%'
group by customername;
```

-- Ans2

```
select customername, count(*) as num_orders
from customers c join orders o
on c.customernumber = o.customernumber
where substr(customername,1,1)='D'
group by customername;
```

The screenshot shows the SQL Developer interface with a query window titled 'INT205\_Lab01\_Solution\_01\_2022'. The query is as follows:

```

39 -- 63199999999
40 /* 2.5 List the customer name and the number of sales orders of all customers whose name start
41 with the letter 'D'. Name the number of sales orders column to "num_orders".*/
42 -- Ans1
43 • select customername, count(o.ordernumber) as num_orders
44 from customers c join orders o
45 on c.customernumber = o.customernumber
46 where customername like 'D%'
47 group by customername;
48 -- Ans2
49 • select customername, count(*) as num_orders
50 from customers c join orders o
51 on c.customernumber = o.customernumber
52 where substr(customername,1,1)='D'
53 group by customername;

```

The left pane shows the 'SCHEMAS' tree with 'customers' selected. The 'Table: customers' details are shown below:

Column Name	Data Type	Constraints
customerNumber	int PK	
customerName	varchar(50)	
contactLastName	varchar(50)	
contactFirstName	varchar(50)	
phone	varchar(50)	
addressLine1	varchar(50)	
addressLine2	varchar(50)	
city	varchar(50)	
state	varchar(50)	
postalCode	varchar(15)	
country	varchar(50)	
salesRepEmployeeNumber	int	
creditLimit	decimal(10, 2)	

The 'Result Grid' shows the following data:

customername	num_orders
Danish Wholesale Imports	5
Dragon Souvenirs, Ltd.	5
Diecast Classics Inc.	4
Daedalus Designs Imports	2
Down Under Souvenirs, Inc	5
Double Decker Gift Stores Ltd	2

The 'Action Output' pane shows the execution of the query:

Time	Action	Response	Duration
59	13:29:08	select customername, count(o.ordernumber) as num_orders from customers c join orders o on...	7 row(s) returned 0.00070

2.6 List the customer name, the sales rep employee last name, and the check number of all customers who made the payment in year 2005. Name the sales rep employee last name column to "salesempname".

-- Ans1

```

select customername, lastname as salesempname, checknumber
from employees e join customers c
on e.employeeNumber = c.salesrepemployeenumber
join payments p
on p.customernumber = c.customernumber
where paymentdate like '%2005%';

```

-- Ans2

```

select customername, lastname as salesempname, checknumber
from employees e, customers c, payments p
where e.employeeNumber = c.salesrepemployeenumber
and p.customernumber = c.customernumber
and extract(year from paymentdate) = 2005;
-- and substr(paymentdate,1,4)=2005 ;

```

The screenshot shows a database management system interface with a query editor and a results grid. The query editor contains the following SQL code:

```

48 /*2.6 List the customer name, the sales rep employee last name, and the check number of all customers
49 who made the payment in year 2005. Name the sales rep employee last name column to "salesempname".*/
50 -- Ans1
51 • select customername, lastname as salesempname, checknumber
52 from employees e join customers c
53 on e.employeeenumber = c.salesrepemployeeenumber
54 join payments p
55 on p.customernumber = c.customernumber
56 where paymentdate like '%2005%';
57 -- Ans2
58 • select customername, lastname as salesempname, checknumber
59 from employees e, customers c, payments p
60 where e.employeeenumber = c.salesrepemployeeenumber
61 and p.customernumber = c.customernumber
62 and extract(year from paymentdate) = 2005;
63 -- and substr(paymentdate,1,4)=2005 ;

```

The results grid shows the following data:

customername	salesempname	checknumber
La Rochelle Gifts	Hernandez	NG94694
Mini Gifts Distributors Ltd.	Jennings	AE215433
Mini Gifts Distributors Ltd.	Jennings	ET64396
Euro+ Shopping Channel	Hernandez	DL460618
Euro+ Shopping Channel	Hernandez	IN446258

The interface also shows a schema tree on the left with tables like customers, employees, offices, orderdetails, orders, payments, productlines, products, Views, Stored Procedures, and Functions. The bottom status bar indicates that the query was completed successfully, returning 37 rows in 0.0025 seconds.

2.7 List the manager last name (the employee who were reported to) and the number of employees of all managers. Name the manager last name and the number of employees columns to “mgrname” and “num\_emp”, respectively.

-- Ans1

```

select m.lastname as mgrname, count(e.employeeenumber) as num_emp
from employees e join employees m
on e.reportsto = m.employeeenumber
group by m.lastname;

```

-- Ans2

```

select m.lastname as mgrname, count(*) as num_emp
from employees e join employees m
on e.reportsto = m.employeeenumber
group by m.lastname;

```

The screenshot shows the SQL Server Enterprise Manager interface. The left pane displays the 'SCHEMAS' tree with 'customers' and 'employees' tables selected. The central query window contains two SQL queries:

```

67 /*2.7 List the manager last name (the employee who were reported to) and the number of employees of
68 all managers. Name the manager last name and the number of employees columns to "mgrname" and "num_emp"
69 respectively.*/
70 -- Ans1
71 • select m.lastname as mgrname, count(e.employeeid) as num_emp
72   from employees e join employees m
73   on e.reportsto = m.employeeid
74  group by m.lastname;
75 -- Ans2
76 • select m.lastname as mgrname, count(*) as num_emp
77   from employees e join employees m
78   on e.reportsto = m.employeeid
79  group by m.lastname;

```

The bottom pane shows the 'Result Grid' for the first query (Ans1). It displays a table with two columns: 'mgrname' and 'num\_emp'. The data rows are:

mgrname	num_emp
Murphy	2
Patterson	7
Bondur	6
Bow	6
Nishi	1

The bottom status bar shows the execution time for the query: 13:26:14, 5 row(s) returned, 0.00057 sec / 0.0000...

2.8 Create your own question and the answered SQL statement should have SELECT, FROM, WHERE, GROUP BY and HAVING clauses.

-- Please write your question in English or Thai.

List the customer name and the number of sales orders of all customers who live in the country named USA and the number of sales orders is greater than 10. Name the number of sales orders column to "num\_orders".

```

select customername, count(ordernumber) as num_orders
from customers c join orders o
on c.customerid = o.customerid
where country = 'USA'
group by customername
having count(ordernumber) > 10;

```

The screenshot shows a database management system interface with a left sidebar containing a tree view of database objects (Schemas, Tables, Views, Functions, etc.). The main area displays a SQL query editor with the following code:

```

87  /* 2.8 Create your own question and the answered SQL statement should have SELECT,
88  FROM, WHERE, GROUP BY and HAVING clauses.
89  -- Please write your question in English or Thai. */
90
91  /*List the customer name and the number of sales orders of all customers who live in
92  the country named USA and the number of sales orders is greater than 10.
93  Name the number of sales orders column to "num_orders".*/
94
95  • select customername, count(ordernumber) as num_orders
96      from customers c join orders o
97      on c.customernumber = o.customernumber
98      where country = 'USA'
99      group by customername
100     having count(ordernumber) > 10;
101

```

Below the query editor, the 'Result Grid' shows the results of the query. The first row is highlighted, showing 'Mini Gifts Distributors Ltd.' with a value of 17 in the 'num\_orders' column.

customername	num_orders
Mini Gifts Distributors Ltd.	17

The 'Action Output' section at the bottom shows the execution details of the query, including the time taken and the number of rows returned.

Time	Action	Response	Duration / Fetch
59 13:29:08	select customername, count(o.ordernumber) as num_orders from customers c join orders o on...	7 row(s) returned	0.00070 sec / 0.
60 13:32:37	select customername, count(ordernumber) as num_orders from customers c join orders o on...	1 row(s) returned	0.0017 sec / 0.0

2.9 Create your own question and the answered SQL statement should have SELECT, FROM 3 tables, WHERE and ORDER BY clauses.

-- Please write your question in English or Thai.

List the product name, price and ordered quantity of the order number 10124.  
Sort the results in ascending order by product name.

-- Ans1

```

select productname, priceeach, quantityordered
from products p join orderdetails od
on p.productcode = od.productcode
join orders o
on od.ordernumber = o.ordernumber
where o.ordernumber = 10124
order by productname;

```

-- Ans2

```

select productname, priceeach, quantityordered
from products p, orderdetails od, orders o
where p.productcode = od.productcode
and od.ordernumber = o.ordernumber
and o.ordernumber = 10124
order by 1;

```



The screenshot shows the SQL Developer interface with the following components:

- Top Bar:** Home icon, session ID 63199999999, and a toolbar with various icons.
- Navigation Panel (Left):**
  - SCHEMAS:** Filter objects search bar.
  - classicmodels:** Expanded to show Tables (customers, employees, offices, orderdetails, orders, payments, productlines, products), Views, Stored Procedures, and Functions.
  - Object Info:** Table: customers.
  - Columns:**
    - customerNumber: int PK
    - customerName: varchar(50)
    - contactLastName: varchar(50)
    - contactFirstName: varchar(50)
    - phone: varchar(50)
    - addressLine1: varchar(50)
    - addressLine2: varchar(50)
    - city: varchar(50)
    - state: varchar(50)
    - postalCode: varchar(15)
    - country: varchar(50)
    - salesRepEmployeeNumber: int
    - creditLimit: decimal(10, 2)
- SQL Editor (Center):**

```

101
102 /* 2.9 Create your own question and the answered SQL statement should have SELECT,
103 FROM 3 tables, WHERE and ORDER BY clauses.
104 -- Please write your question in English or Thai. */
105
106 /*List the product name, price and ordered quantity of the order number 10124.
107 Sort the results in ascending order by product name.*/
108 -- Ans1
109 • select productname, priceeach, quantityordered
110 from products p join orderdetails od
111 on p.productcode = od.productcode
112 join orders o
113 on od.ordernumber = o.ordernumber
114 where o.ordernumber = 10124
115 order by productname;
116 -- Ans2

```
- Result Grid (Bottom):**
  - Filter Rows: Search
  - Export: [Icon]
  - Columns: productname, priceeach, quantityordered
  - Results:

productname	priceeach	quantityordered
1911 Ford Town Car	58.12	42
1917 Grand Touring Sedan	153.00	21
1932 Alfa Romeo 8C2300 Spider Sport	75.46	36
1932 Model A Ford J-Coupe	111.87	42
1936 Mercedes Benz 500k Roadster	36.11	46
- Action Output (Bottom):**
  - Time: 13:37:35
  - Action: select productname, priceeach, quantityordered from products p join orderdetails od on p.pro...
  - Response: 13 row(s) returned

2.10 Create your own question and the answered SQL statement should have SELECT, FROM 3 tables, WHERE, GROUP BY, HAVING and ORDER BY clauses.  
 -- Please write your question in English or Thai.

List the product name and the total ordered quantity of all products that were ordered in January and their total ordered quantities are greater than 100. Sort the results in descending order by the total ordered quantity.

```

select productname, sum(quantityordered) as total_quantity_ordered
from products p, orderdetails od, orders o
where p.productcode = od.productcode
and od.ordernumber = o.ordernumber
and extract(month from o.orderdate) = 01
group by productname
having sum(quantityordered) > 100
order by 2 desc;

```

63199999999

Administration Schemas INT205\_Lab01\_Solution\_01\_2022\* orders

SCHEMAS

Filter objects

classicmodels

Tables

- customers
- employees
- offices
- orderdetails
- orders
- payments
- productlines
- products
- Views
- Stored Procedures
- Functions

Object Info Session

Table: **orders**

Columns:

orderNumber	int PK
orderDate	date
requiredDate	date
shippedDate	date
status	varchar(15)
comments	text
customerNumber	int

```

124 -- 2.10 Create your own question and the answered SQL statement should have SELECT,
125 -- FROM 3 tables, WHERE, GROUP BY, HAVING and ORDER BY clauses.
126 -- Please write your question in English or Thai.
127
128 /*List the product name and the total ordered quantity of all products that were ordered in January
129 and their total ordered quantities are greater than 100.
130 Sort the results in descending order by the total ordered quantity.*/
131
132 • select productname, sum(quantityordered) as total_quantity_ordered
133 from products p, orderdetails od, orders o
134 where p.productcode = od.productcode
135 and od.ordernumber = o.ordernumber
136 and extract(month from o.orderdate) = 01
137 group by productname
138 having sum(quantityordered) > 100
139 order by 2 desc;

```

100% 2:132

Result Grid Filter Rows: Search Export:

productname	total_quantity_ordered
1936 Mercedes Benz 500k Roadster	139
1917 Maxwell Touring Car	132
1992 Ferrari 360 Spider red	124
1938 Cadillac V-16 Presidential Limousine	123
1913 Ford Model T Speedster	118

Result 24

Action Output

Time	Action	Response	Duration
69 13:48:51	select productname, sum(quantityordered) as total_quantity_ordered from products p, orde...	14 row(s) returned	0.0019 s