

SQL Lab-4

1. Create a view that represents total sales per order from the ordersdetails table

Ans:

create view totalsales

as

select ordernumber, sum(quantityordered*priceeach) as total

from orderdetails

group by ordernumber;

```
mysql> create view totalsales
-> as
-> select ordernumber, sum(quantityordered*priceeach) as total
-> from orderdetails
-> group by ordernumber;
Query OK, 0 rows affected (0.01 sec)

mysql> select * from totalsales;
```

ordernumber	total
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	48425.69
10111	16537.85
10112	7674.94
10113	11044.30
10114	33383.14
10115	21665.98
10116	1627.56
10117	44380.15
10118	3101.40

2. Create a view that contains products whose buy prices are higher than the average price of all products

Ans:

create view product_prices

as

select productcode, productname from products

where buyprice > (Select avg(buyprice) from products);

```
mysql> create view product_prices
-> as
-> select productcode, productname from products
-> where buyprice > (Select avg(buyprice) from products);
Query OK, 0 rows affected (0.01 sec)

mysql> select * from productprices;
ERROR 1146 (42S02): Table 'classicmodels.productprices' doesn't exist
mysql> select * from product_prices;
```

productcode	productname
S10_1949	1952 Alpine Renault 1300
S10_2016	1996 Moto Guzzi 1100i
S10_4698	2003 Harley-Davidson Eagle Drag Bike
S10_4757	1972 Alfa Romeo GTA
S10_4962	1962 LanciaA Delta 16V
S12_1099	1968 Ford Mustang
S12_1108	2001 Ferrari Enzo
S12_1666	1958 Setra Bus
S12_2823	2002 Suzuki XREO
S12_3148	1969 Corvair Monza
S12_3380	1968 Dodge Charger
S12_3891	1969 Ford Falcon
S12_4473	1957 Chevy Pickup
S12_4675	1969 Dodge Charger
S18_1097	1940 Ford Pickup Truck
S18_1129	1993 Mazda RX-7
S18_1342	1937 Lincoln Berline
S18_1589	1965 Aston Martin DB5

3. create a procedure to select the name, city, state, postcode and country from the customers table in the alphabetical order of name

Ans:

```
create procedure alphaorder()
begin
select customername, city, state, postcode, country
from customers
order by customername;
end/
```

```
mysql> create procedure alphaorder()
-> begin
-> select customername, city, state, postcode, country
-> from customers
-> order by customername;
-> end/
Query OK, 0 rows affected (0.03 sec)

mysql> call alphaorder();
```

customername	city	state	postcode	country
Alpha Cognac	Toulouse	NULL	31000	France
American Souvenirs Inc	New Haven	CT	97823	USA
Amica Models & Co.	Torino	NULL	10100	Italy
ANG Resellers	Madrid	NULL	28001	Spain
Anna's Decorations, Ltd	North Sydney	NSW	2060	Australia
Anton Designs, Ltd.	Madrid	NULL	28023	Spain
Asian Shopping Network, Co	Singapore	NULL	038988	Singapore
Asian Treasures, Inc.	Cork	Co. Cork	NULL	Ireland
Atelier graphique	Nantes	NULL	44000	France
Australian Collectables, Ltd	Glen Waverly	Victoria	3150	Australia
Australian Collectors, Co.	Melbourne	Victoria	3004	Australia
Australian Gift Network, Co	South Brisbane	Queensland	4101	Australia
Auto Associés & Cie.	Versailles	NULL	78000	France
Auto Canal+ Petit	Paris	NULL	75016	France
Auto-Moto Classics Inc.	Brickhaven	MA	58339	USA
AV Stores, Co.	Manchester	NULL	EC2 5NT	UK
Baane Mini Imports	Stavern	NULL	4110	Norway
Bavarian Collectables Imports, Co.	Munich	NULL	80686	Germany
BG&E Collectables	Fribourg	NULL	1700	Switzerland
Blauer See Auto, Co.	Frankfurt	NULL	60528	Germany

4. Create a stored procedure that finds all offices that locate in a country specified by the input parameter countryName

Ans:

```
create procedure locoffice(countryName varchar(20))
begin
select officecode from offices where country=countryName;
```

end/

```
mysql> create procedure locoffice(countryName varchar(20))
-> begin
-> select officecode from offices where country=countryName;
-> end/
Query OK, 0 rows affected (0.01 sec)

mysql> call locoffice("USA")
-> /
+-----+
| officecode |
+-----+
| 1          |
| 2          |
| 3          |
+-----+
3 rows in set (0.00 sec)

Query OK, 0 rows affected (0.01 sec)
```

5. Create a stored procedure to find the number of orders that already shipped by passing the orderstatus into the procedure

Ans:

```
create procedure ordernum(orderstatus varchar(20))
begin
select count(ordernumber) from orders
where status= orderstatus;
end/
```

```

mysql> create procedure ordernum(orderstatus varchar(20))
-> begin
-> select count(ordernumber) from orders
-> where status= orderstatus;
-> end/
Query OK, 0 rows affected (0.00 sec)

mysql> call oredrnum("shipped")/
ERROR 1305 (42000): PROCEDURE classicmodels.oredrnum does not exist
mysql> call ordernum("shipped")/
+-----+
| count(ordernumber) |
+-----+
|                303 |
+-----+
1 row in set (0.00 sec)

Query OK, 0 rows affected (0.00 sec)

```

6. Create a stored procedure using if statement which inputs the customernumber and selects the creditlimit and displays the customerlevel based on the following condition
- If the credit is greater than 50,000, the level of the customer is PLATINUM.
 - If the credit is less than or equal 50,000 and greater than 10,000, then the level of customer is GOLD.
 - Otherwise, the level of the customer is SILVER.

Ans:

```

create procedure customerlevel(in custnum int, out custlevel varchar(20))
begin
declare cred int;
select creditlimit into cred from customers
where customernumber=custnum;
if cred>50000 then
set custlevel=" PLATINUM";
elseif cred<=50000 and cred>10000 then
set custlevel="GOLD";
else

```

```

set custlevel="SILVER";
end if;
end/

```

```

mysql> create procedure customerlevel(in custnum int, out custlevel varchar(20))
-> begin
-> declare cred int;
-> select creditlimit into cred from customers
-> where customernumber=custnum;
-> if cred>50000 then
-> set custlevel=" PLATINUM";
-> elseif cred<=50000 and cred>10000 then
-> set custlevel="GOLD";
-> else
-> set custlevel="SILVER";
-> end if;
-> end/
Query OK, 0 rows affected (0.00 sec)

mysql> call customerlevel(125,@1)/
Query OK, 1 row affected (0.00 sec)

mysql> select @1/
+-----+
| @1    |
+-----+
| SILVER |
+-----+
1 row in set (0.00 sec)

```

7. Create a stored procedure using case which inputs the customernumber and selects the country and displays the shipping time based on the following condition
- If the customer locates in USA , the shipping time is 2-day shipping .
 - If the customer locates in Canada , the shipping time is 3-day shipping .
 - The customers from other countries have 5-day shipping .

Ans:

```

create procedure duration(in custnumber int, out shippingdays
varchar(50))

```

```
begin
declare custcountry varchar(20);
select country into custcountry from customers
where customernumber=custnumber;
case custcountry
when "USA" then set shippingdays="2-day shipping";
when "Canada" then set shippingdays="3-day shipping";

else
set shippingdays="5-day shipping";
end case;
end/
```

```

mysql> create procedure duration(in custnumber int, out shippingdays varchar(50))
-> begin
-> declare custcountry varchar(20);
-> select country into custcountry from customers
-> where customernumber=custnumber;
-> case custcountry
-> when "USA" then set shippingdays="2-day shipping";
-> when "Canada" then set shippingdays="3-day shipping";
->
-> else
-> set shippingdays="5-day shipping";
-> end case;
-> end/
Query OK, 0 rows affected (0.00 sec)

mysql> call duration(125,@s)/
Query OK, 1 row affected (0.00 sec)

mysql> select @s/
+-----+
| @s    |
+-----+
| 5-day shipping |
+-----+
1 row in set (0.00 sec)

```

8. Create a stored function using if statement which inputs a credit and returns the customerlevel based on the following condition
- If the credit is greater than 50,000, the level of the customer is PLATINUM.
 - If the credit is less than or equal 50,000 and greater than 10,000, then the level of customer is GOLD.
 - Otherwise, the level of the customer is SILVER.

Display the customer name and customerlevel of all customers

Ans:

```

create procedure level(in credit int, out custlevel varchar(20))
begin
declare cred int;
select creditlimit into cred from customers
where creditlimit = credit;

```



```

if cred>50000 then
set custlevel=" PLATINUM";
elseif cred<=50000 and cred>10000 then
set custlevel="GOLD";
else
set custlevel="SILVER";
end if;
end/

```

```

mysql> create procedure level(in credit int, out custlevel varchar(20))
-> begin
-> declare cred int;
-> select creditlimit into cred from customers
-> where creditlimit = credit;
-> if cred>50000 then
-> set custlevel=" PLATINUM";
-> elseif cred<=50000 and cred>10000 then
-> set custlevel="GOLD";
-> else
-> set custlevel="SILVER";
-> end if;
-> end/
Query OK, 0 rows affected (0.00 sec)

mysql> call level(21000,@cl)/
Query OK, 1 row affected (0.00 sec)

mysql> select @cl/
+-----+
| @cl |
+-----+
| GOLD |
+-----+
1 row in set (0.00 sec)

```

9. Create a table employees_audit with the following data

Column	Datatype	Constraint
id	int	Primary key,autoincrement
employeenumber	int	Not null

lastname	Varchar(50)	Not null
changedat	datetime	
action	Varchar(50)	

Ans:

```
create table employees_audit (
id int primary key auto_increment,
employeenumber int not null,
lastname varchar(50) not null,
changedat datetime,
action varchar(50)
);
```

```
mysql> create table employees_audit (
-> id int primary key auto_increment,
-> employeenumber int not null,
-> lastname varchar(50) not null,
-> changedat datetime,
-> action varchar(50)
-> );
```

Query OK, 0 rows affected (0.01 sec)

```
mysql> desc employees_audit;
```

Field	Type	Null	Key	Default	Extra
id	int	NO	PRI	NULL	auto_increment
employeenumber	int	NO		NULL	
lastname	varchar(50)	NO		NULL	
changedat	datetime	YES		NULL	
action	varchar(50)	YES		NULL	

5 rows in set (0.00 sec)

10. Create a trigger which will insert into the employees_audit table before updating the employees table.action should be set as "update",employeeenumber and lastname should be set with the old value and changedat should be set with the current date and time. Update rows in the employees table and check the employees_audit table

Ans:

Trigger-

```
create trigger emp_trigger before update
on employees
for each row
begin
insert into employees_audit (employeeenumber, lastname, changedat,
action)
values (old.employeeenumber, old.lastname, now(),'update');
end/
```

update-

```
update employees
set lastname = 'peterson'
where employeeenumber = 1056;
```

```
mysql> create trigger emp_trigger before update
-> on employees
-> for each row
-> begin
-> insert into employees_audit (employeenumber, lastname, changedat, action)
-> values (old.employeenumber, old.lastname, now(),'update');
-> end/
Query OK, 0 rows affected (0.01 sec)

mysql> select lastname, employeenumber from employees limit 3/
+-----+-----+
| lastname | employeenumber |
+-----+-----+
| Murphy   |          1002 |
| Patterson |          1056 |
| Firrelli |          1076 |
+-----+-----+
3 rows in set (0.00 sec)

mysql> update employees
-> set lastname = 'peterson'
-> where employeenumber = 1056;
-> /
Query OK, 1 row affected (0.00 sec)
Rows matched: 1  Changed: 1  Warnings: 0

mysql> select lastname, employeenumber from employees limit 3/
+-----+-----+
| lastname | employeenumber |
+-----+-----+
| Murphy   |          1002 |
| peterson  |          1056 |
| Firrelli |          1076 |
+-----+-----+
```

Employees_audit table-

```
mysql> select * from employees_audit/
+-----+-----+-----+-----+-----+
| id | employeenumber | lastname | changedat          | action |
+-----+-----+-----+-----+-----+
| 1 |          1056 | Patterson | 2023-09-27 15:41:20 | update |
+-----+-----+-----+-----+-----+
1 row in set (0.00 sec)
```