SQL Lab-4

1. **Create a view that represents total sales per order from the orders table.**
   mysql>create or replace view total_sales
   -> as
   -> select ordernumber, sum(quantityordered * priceeach) totalprice from orderdetails group by ordernumber;
   Query OK, 0 rows affected (0.01 sec)

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

mysql> select * from total_sales;
+-------------+------------+
| ordernumber | totalprice |
+-------------+------------+
|       10100 |   10223.83 |
|       10101 |   10549.01 |
|       10102 |    5494.78 |
|       10103 |   50218.95 |
|       10104 |   40206.20 |
|       10105 |   53959.21 |
```

326 rows in set (0.00 sec)

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

   mysql> create or replace view avg_buy
   -> as
   -> select * from products where buyprice > (select avg(buyprice) from products);
   Query OK, 0 rows affected (0.01 sec)

```
mysql> create or replace view avg_buy
    -> as
    -> select * from products where buyprice > (select avg(buyprice) from products);
Query OK, 0 rows affected (0.01 sec)

mysql> select * from avg_buy;
| productCode | productName                          | productLine      | productScale | productVendor              | productDescription

                    | quantityInStock | buyPrice | MSRP   |
| S10_1949    | 1952 Alpine Renault 1300             | Classic Cars     | 1:10         | Classic Metal Creations    | Turnable front wheels; steering function; detailed interior; detailed engine; ope
ning hood; opening trunk; opening doors; and detailed chassis.

                    |            7305 |    98.58 | 214.30 |
| S10_2016    | 1996 Moto Guzzi 1100i                | Motorcycles      | 1:10         | Highway 66 Mini Classics   | Official Moto Guzzi logos and insignias, saddle bags located on side of motorcycl
e, detailed engine, working steering, working suspension, two leather seats, luggage rack, dual exhaust pipes, small saddle bag located on handle bars, two-tone paint with chrome accents, superior die-
cast detail , rotating wheels , working kick stand, diecast metal with plastic parts and baked enamel finish.
                    |            6625 |    68.99 | 118.94 |
| S10_4698    | 2003 Harley-Davidson Eagle Drag Bike | Motorcycles      | 1:10         | Red Start Diecast          | Model features, official Harley Davidson logos and insignias, detachable rear whe
elie bar, heavy diecast metal with resin parts, authentic multi-color tampo-printed graphics, separate engine drive belts, free-turning front fork, rotating tires and rear racing slick, certificate of
authenticity, detailed engine, display stand
, precision diecast replica, baked enamel finish, 1:10 scale model, removable fender, seat and tank cover piece for displaying the superior detail of the v-twin engine |            5582 |    91.02 | 19
3.66 |
| S10_4757    | 1972 Alfa Romeo GTA                  | Classic Cars     | 1:10         | Motor City Art Classics    | Features include: Turnable front wheels; steering function; detailed interior; de
tailed engine; opening hood; opening trunk; opening doors; and detailed chassis.
```

54 rows in set (0.00 sec)

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

mysql> create procedure cust_details()
-> begin
-> select customername, city,state,postalcode,country from customers order by customername;
-> end/
Query OK, 0 rows affected (0.01 sec)
mysql> call cust_details();

```
mysql> create procedure cust_details()
    -> begin
    -> select customername, city,state,postalcode,country from customers order by customername;
    -> end/
Query OK, 0 rows affected (0.01 sec)

mysql> call cust_details();
    -> /
+----------------------------------+------------------+--------------+------------+--------------+
| customername                     | city             | state        | postalcode | 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       |
```

122 rows in set (0.00 sec)

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

mysql> create procedure getoffice(countryname varchar(50))
-> begin
-> select officecode,city from offices where country = countryname;
-> end/
Query OK, 0 rows affected (0.11 sec)
mysql> call getoffice('USA')/

```
mysql> create procedure getoffice(countryname varchar(50))
    -> begin
    -> select officecode,city from offices where country = countryname;
    -> end/
Query OK, 0 rows affected (0.11 sec)

mysql> call getoffice('USA')/
+------------+---------------+
| officecode | city          |
+------------+---------------+
| 1          | San Francisco |
| 2          | Boston        |
| 3          | NYC           |
+------------+---------------+
3 rows in set (0.00 sec)

Query OK, 0 rows affected (0.00 sec)
```

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

mysql> create procedure countorders(in orderstatus varchar(25),out total int)
-> begin
-> select count(ordernumber) into total from orders where status=orderstatus;
-> select total;
-> end/
Query OK, 0 rows affected (0.01 sec)

```
mysql> delimiter /
mysql> create procedure countorders(in orderstatus varchar(25),out total int)
    -> begin
    -> select count(ordernumber) into total from orders where status=orderstatus;
    -> select total;
    -> end/
Query OK, 0 rows affected (0.01 sec)

mysql> call countorders('shipped',@t)/
+-------+
| total |
+-------+
|   303 |
+-------+
1 row in set (0.01 sec)

Query OK, 0 rows affected (0.01 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.**

```
mysql> create procedure getcustlevel(in custnum int,out customerlevel varchar(20))
    -> begin
    -> declare credit decimal default 0;
    -> select creditlimit into credit from customers where customernumber = custnum;
    -> IF credit>50000 then
    -> set customerlevel = 'platinum';
    -> elseif credit<=50000 and credit > 10000 then
    -> set customerlevel = 'GOLD';
    -> else
    -> set customerlevel = 'SILVER';
    -> END IF;
    -> END/
Query OK, 0 rows affected (0.01 sec)
```

```
mysql> call getcustlevel(103,@level)/
Query OK, 1 row affected (0.00 sec)

mysql> select @level/
+--------+
| @level |
+--------+
| GOLD   |
+--------+
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 .**

```
mysql> create procedure getshipping (in custno int, out ship varchar(50))
    -> begin
    -> declare custcountry varchar(50);
    -> select country into custcountry from customers where customernumber = custno;
    -> case custcountry
    -> when 'USA' then set ship = 'the shipping time is 2-day shipping.';
    -> when 'Canada' then set ship = 'the shipping time is 3-day shipping.';
    -> else
    -> set ship = '5-day shipping.';
    -> end case;
    -> end/
Query OK, 0 rows affected (0.01 sec)

mysql> call getshipping(112,@shp)/
Query OK, 1 row affected (0.00 sec)

mysql> select @shp/
+-------------------------------------+
| @shp                                |
+-------------------------------------+
| the shipping time is 2-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

```
mysql> create function custlevel(cred decimal(10,2))
    -> returns varchar(50)
    -> DETERMINISTIC
    -> begin
    -> declare custlevel varchar(50);
    -> if cred >50000 then
    -> set custlevel = 'platinum';
    -> elseif (cred >= 50000 AND cred <= 10000) THEN
    -> set custlevel = 'gold';
    -> else
    -> set custlevel = 'silver';
    -> end if;
    -> return (custlevel);
    -> end/
Query OK, 0 rows affected (0.01 sec)
```

```
mysql> create function custlevel(cred decimal(10,2))
    -> returns varchar(50)
    -> DETERMINISTIC
    -> begin
    -> declare custlevel varchar(50);
    -> if cred >50000 then
    -> set custlevel = 'platinum';
    -> elseif (cred >= 50000 AND cred <= 10000) THEN
    -> set custlevel = 'gold';
    -> else
    -> set custlevel = 'silver';
    -> end if;
    -> return (custlevel);
    -> end/
Query OK, 0 rows affected (0.01 sec)

mysql> select customername, custlevel(creditlimit) from customers/
+----------------------------------+-------------------------+
| customername                     | custlevel(creditlimit) |
+----------------------------------+-------------------------+
| Atelier graphique                | silver                  |
| Signal Gift Stores               | platinum                |
| Australian Collectors, Co.       | platinum                |
| La Rochelle Gifts                | platinum                |
| Baane Mini Imports               | platinum                |
| Mini Gifts Distributors Ltd.     | platinum                |
| Havel & Zbyszek Co               | silver                  |
| Blauer See Auto, Co.             | platinum                |
| Mini Wheels Co.                  | platinum                |
| Land of Toys Inc.                | platinum                |
```

122 rows 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)

```
mysql> desc employee_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",employeenumber 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

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

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

```
mysql> update employees set lastname ='Murfy' where employeeNumber =1002/
Query OK, 1 row affected (0.01 sec)
Rows matched: 1  Changed: 1  Warnings: 0

mysql> select * from employee_audit;
    -> /
+----+----------------+----------+---------------------+--------+
| id | employeenumber | lastname | changedat           | action |
+----+----------------+----------+---------------------+--------+
|  1 |           1002 | Murphy   | 2023-09-27 16:58:20 | update |
+----+----------------+----------+---------------------+--------+
1 row in set (0.00 sec)
```