

Assignment 4

Name-Harshayee Jadhav

PRN No-03

Subject-DBMS (SQL Lab 4)

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

```
mysql> create view salesperorder as select ordernumber, sum(quantityordered * priceeach) total from orderdetails group by ordernumber;  
Query OK, 0 rows affected (0.00 sec)
```

```
mysql> select * from salesperorder;  
+-----+-----+  
| 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 |  
| 10119 | 35826.33 |  
| 10120 | 45864.03 |  
| 10121 | 16700.47 |
```

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

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

```
mysql> select * from highest_buyprice;
```

productname	highest_price
1952 Alpine Renault 1300	98.58
1996 Moto Guzzi 1100i	68.99
2003 Harley-Davidson Eagle Drag Bike	91.02
1972 Alfa Romeo GTA	85.68
1962 LanciaA Delta 16V	103.42
1968 Ford Mustang	95.34
2001 Ferrari Enzo	95.59
1958 Setra Bus	77.90
2002 Suzuki XREO	66.27
1969 Corvair Monza	89.14
1968 Dodge Charger	75.16
1969 Ford Falcon	83.05
1957 Chevy Pickup	55.70
1969 Dodge Charger	58.73
1940 Ford Pickup Truck	58.33
1993 Mazda RX-7	83.51
1937 Lincoln Berline	60.62
1965 Aston Martin DB5	65.96
1980s Black Hawk Helicopter	77.27
1917 Grand Touring Sedan	86.70
1995 Honda Civic	93.89
1998 Chrysler Plymouth Prowler	101.51
1964 Mercedes Tour Bus	74.86
1932 Model A Ford J-Coupe	58.48
1928 Mercedes-Benz SSK	72.56
1999 Indy 500 Monte Carlo SS	56.76
1913 Ford Model T Speedster	60.78
18th Century Vintage Horse Carriage	60.74

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

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

mysql> call cust_details()/
```

customername	city	postalcode	country
Alpha Cognac	Toulouse	31000	France
American Souvenirs Inc	New Haven	97823	USA
Amica Models & Co.	Torino	10100	Italy
ANG Resellers	Madrid	28001	Spain
Anna's Decorations, Ltd	North Sydney	2060	Australia
Anton Designs, Ltd.	Madrid	28023	Spain
Asian Shopping Network, Co	Singapore	038988	Singapore
Asian Treasures, Inc.	Cork	NULL	Ireland
Atelier graphique	Nantes	44000	France
Australian Collectables, Ltd	Glen Waverly	3150	Australia
Australian Collectors, Co.	Melbourne	3004	Australia

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

```
mysql> delimiter /
mysql> create procedure office_country( IN countryname varchar(50))
  -> begin
    -> select * from offices where country=countryname;
    -> end/
Query OK, 0 rows affected (0.00 sec)
```

```
mysql> call office_country('usa')/
+-----+-----+-----+-----+-----+
| officeCode | city          | phone          | addressLine1 | addressLine2 | state | country | postalCode | territory |
+-----+-----+-----+-----+-----+
| 1          | San Francisco | +1 650 219 4782 | 100 Market Street | Suite 300 | CA    | USA     | 94080      | NA        |
| 2          | Boston        | +1 215 837 0825 | 1550 Court Place  | Suite 102 | MA    | USA     | 02107      | NA        |
| 3          | NYC           | +1 212 555 3000 | 523 East 53rd Street | apt. 5A | NY    | USA     | 10022      | NA        |
+-----+-----+-----+-----+-----+
3 rows in set (0.01 sec)
```

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

```
mysql> delimiter /
mysql> create procedure count_orderstatus (IN orderstatus varchar (50), OUT total int)
    -> begin
    -> select count(ordernumber) into total from orders where status=orderstatus;
    -> end/
Query OK, 0 rows affected (0.00 sec)

mysql> call count_orderstatus/
ERROR 1318 (42000): Incorrect number of arguments for PROCEDURE classicmodels.count_orderstatus; expected 2, got 0
mysql> call count_orderstatus ('shipped', @total)/
Query OK, 1 row affected (0.00 sec)

mysql> select @total/
+-----+
| @total |
+-----+
|    303 |
+-----+
1 row in set (0.00 sec)

mysql> |
```

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> delimiter /
mysql> create procedure getcustomerlevel(IN pcustomernumber int,OUT pcustomerlevel varchar (25))
    -> begin
    -> declare credit decimal (10,2) default 0;
    -> select creditlimit
    -> into credit
    -> where customernumber = pcustomernumber;
    -> if credit > 50000 then
    -> set pcustomerlevel = 'PLATINUM';
    -> elseif credit <= 50000 and credit > 10000 then
    -> set pcustomerlevel = 'GOLD';
    -> else
    -> set pcustomerlevel = 'SILVER';
    -> end if;
    -> end/
Query OK, 0 rows affected (0.00 sec)
```

Platinum

```
mysql> call cust_credit (141,@c);
    -> /
Query OK, 1 row affected (0.00 sec)

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

Gold

```
mysql> call getcustomerlevel(447,@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 getcustomershipping( in pcustomernumber int,out pshipping varchar(50))
-> begin
-> declare customercountry varchar(50);
-> select country
-> into customercountry
-> from customers
-> where customernumber=pcustomernumber;
-> case customercountry
-> when 'usa' then
-> set pshipping ='2 day shipping';
-> when 'canada' then
-> set pshipping ='3 day shipping';
-> else
-> set pshipping ='5 day shipping';
-> end case;
-> end/
Query OK, 0 rows affected (0.01 sec)

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

mysql> select @shipping/
+-----+
| @shipping |
+-----+
| 2 day shipping |
+-----+
1 row in set (0.00 sec)
```

9. Create a table employees_audit with the following data

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

```
-> /
```

```
Query OK, 0 rows affected (0.05 sec)
```

```
mysql> desc employees_audit/
```

Field	Type	Null	Key	Default	Extra
id	int	NO	PRI	NULL	auto_increment
employeeenumber	int	NO		NULL	
lastname	varchar(50)	NO		NULL	
changedate	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

```
mysql> create trigger before_employee_update before update on employees
-> for each row
-> insert into employees_audit
-> set action='update',
-> employeeenumber=old.employeeenumber, lastname=old.lastname, changedate=now();
-> /
```

```
Query OK, 0 rows affected (0.02 sec)
```

```
mysql> update employees set lastname='donalds' where employeeenumber=1056;
```

```
Query OK, 1 row affected (0.01 sec)
```

```
Rows matched: 1 Changed: 1 Warnings: 0
```

```
mysql> select * from employees_audit;
```

id	employeeenumber	lastname	changedate	action
1	1056	Patterson	2023-09-27 16:59:01	update

```
1 row in set (0.00 sec)
```