DBMS MINI PROJECT

Title: Hypermarket Chains DBMS

Submitted By:
Name: Roshan Kumar U
SRN: PES1UG20CS582
Semester: 5th
Section: J

SHORT DESCRIPTION AND SCOPE OF THE PROJECT

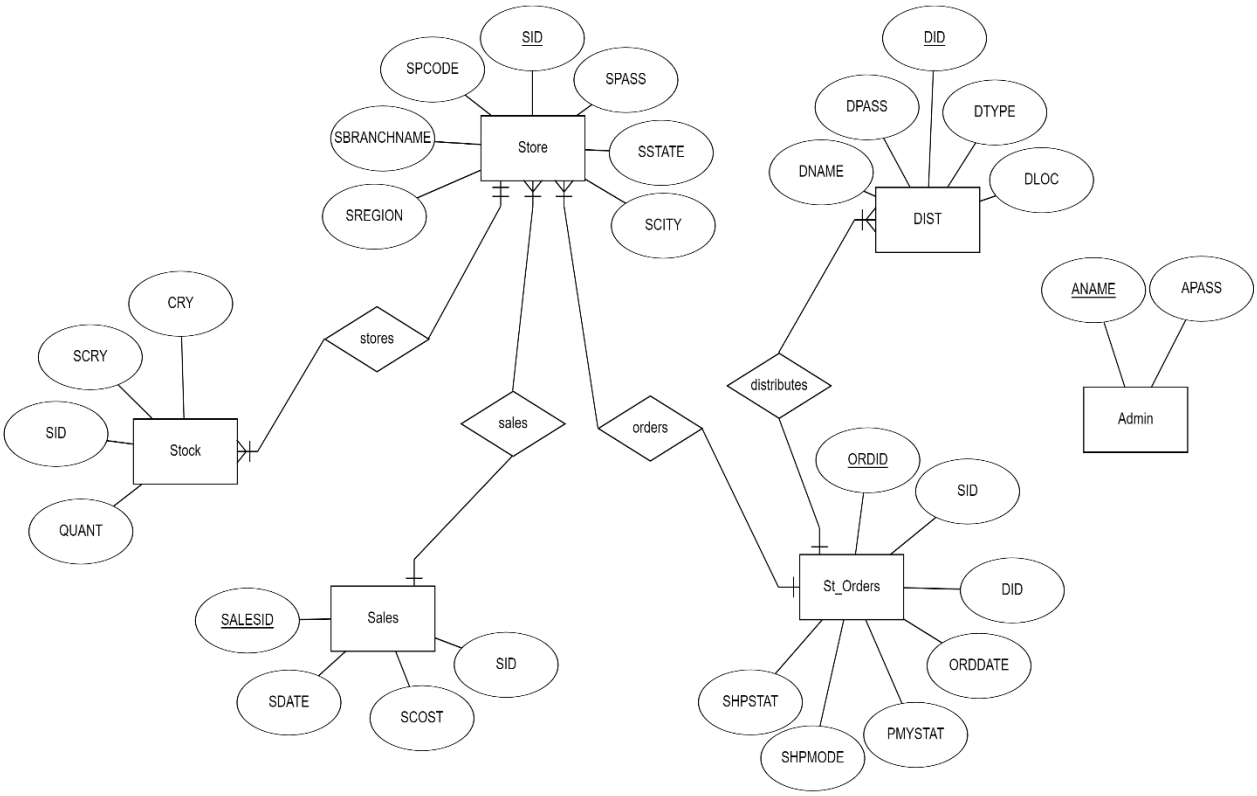
DESCRIPTION:

- The Project describes about the **Hypermarket chains management**.
- This project mainly focuses on how the many stores and distributors can sell various products of different categories. Simply to say, a superstore which is a chain of stores like DMART, BIG Bazaar, etc.
- The database consists of six entities such as **admin, distributor, store, sales, store orders, and stock.**
- The tables are populated with some values and many operations like **join, set, aggregate, functions, procedures, triggers and cursors** are performed on the data.
- A **front end of the database is created with CRUD functionalities** on each table of the database along with **a query box** that can execute and display any SQL query.

SCOPE:

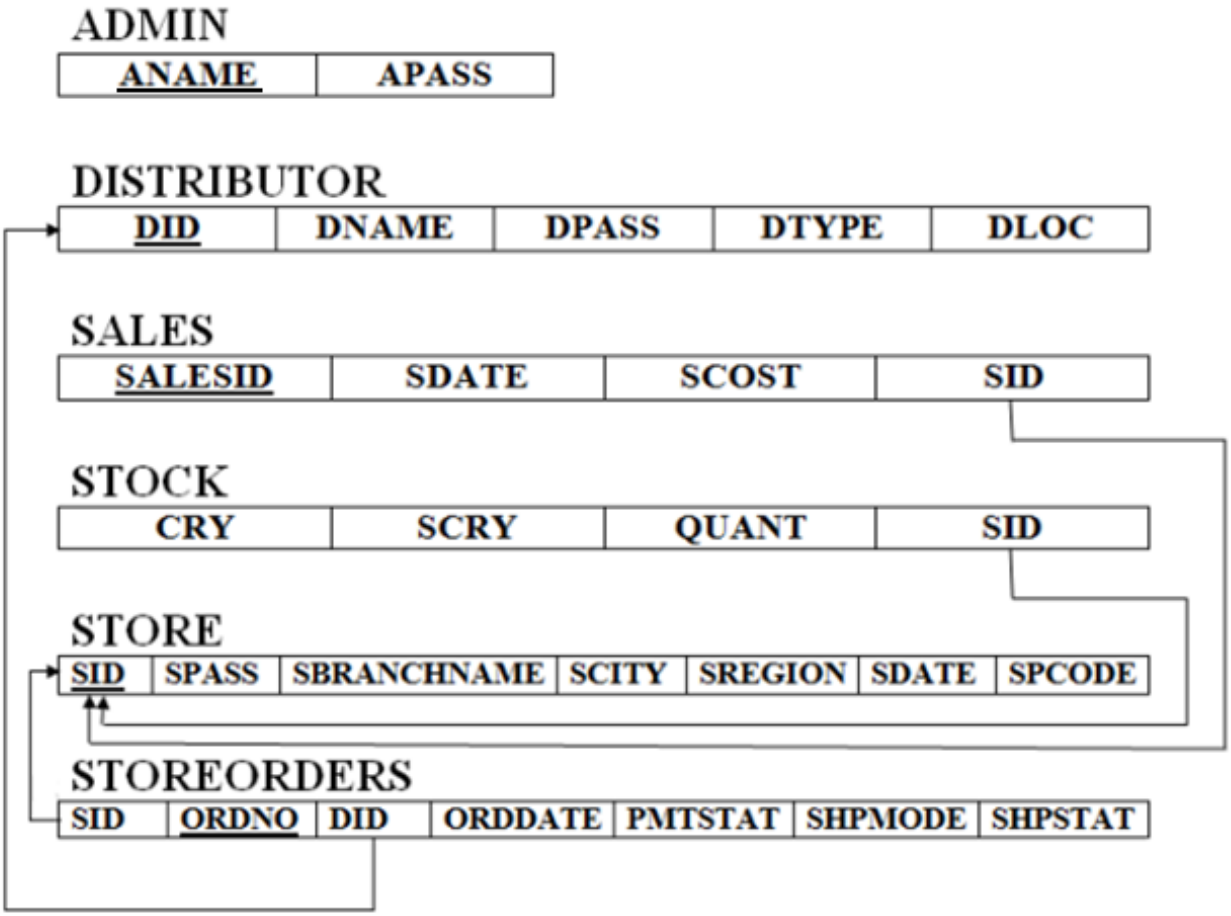
- Hypermarket chains management involves management, creation and development of supermarkets.
- We can deploy it in all kinds of cities and use it for **efficient management** of a hypermarket to ensure all the products are sold at the earliest, with highest discount to the customers, using **deep discounting** models.
- We can closely monitor available stocks and ensure effective stock management, **faster delivery and connectivity** to all cities.
- Many operations can be performed on the database that give us **useful insights** which might go unnoticed to the naked eye. This can help the product user to **prepare, plan, and perfect the art of selling!**

ER Diagram


SID
SPCODE
SPASS
SBRANCHNAME
Store
SSTATE
SREGION
SCITY
DID
DPASS
DTYPE
DNAME
DLOC
DIST
ANAME
APASS
Admin
CRY
SCRY
SID
Stock
QUANT
stores
sales
orders
distributes
ORDID
SID
St_Orders
DID
ORDDATE
SHPSTAT
PMYSTAT
SHPMODE
SALESID
Sales
SID
SDATE
SCOST

Relational Schema

SCHEMA DIAGRAM



DDL statements - Building the database

Table structure for table ‘admin’

QUERY:

```
-- Table structure for table `admin`

CREATE TABLE `admin` (
  `ANAME` varchar(20) NOT NULL,
  `APASS` varchar(20) NOT NULL,
  PRIMARY KEY(ANAME)
) ENGINE=InnoDB DEFAULT CHARSET=latin1;
```

OUTPUT:

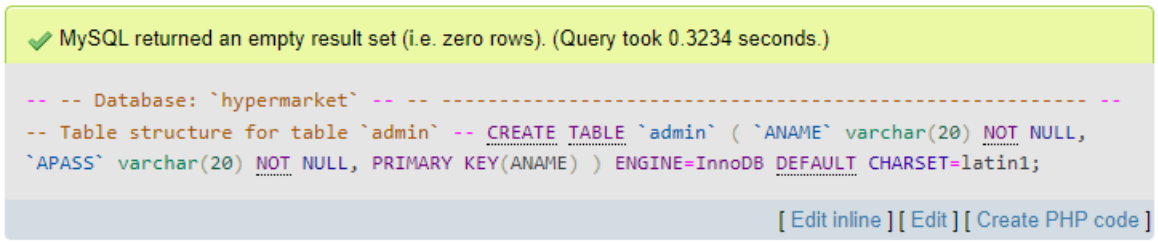


Table structure for table ‘dist’

QUERY:

```
-- Table structure for table `Dist`

CREATE TABLE `Dist` (
  `DID` int(11) NOT NULL,
  `DNAME` varchar(20) NOT NULL,
  `DPASS` varchar(20) NOT NULL DEFAULT 'admin',
  `DTYPE` varchar(20) NOT NULL,
  `DLOC` varchar(30) NOT NULL, PRIMARY KEY(DID)

) ENGINE=InnoDB DEFAULT CHARSET=latin1;
```

OUTPUT:

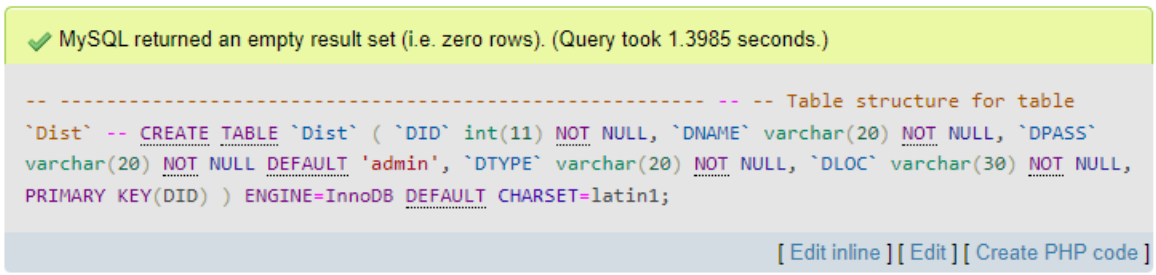


Table structure for table `store`

QUERY:

```
-- Table structure for table `store`


CREATE TABLE `store` (
  `SID` int(11) NOT NULL,
  `SPASS` varchar(20) NOT NULL DEFAULT 'admin',
  `SBRANCHNAME` varchar(20) NOT NULL,
  `SCITY` varchar(20) NOT NULL,
  `SREGION` varchar(20) NOT NULL,
  `SSTATE` varchar(20) NOT NULL,
  `SPCODE` int(11) NOT NULL,PRIMARY KEY(SID)
) ENGINE=InnoDB DEFAULT CHARSET=latin1;
```

OUTPUT:

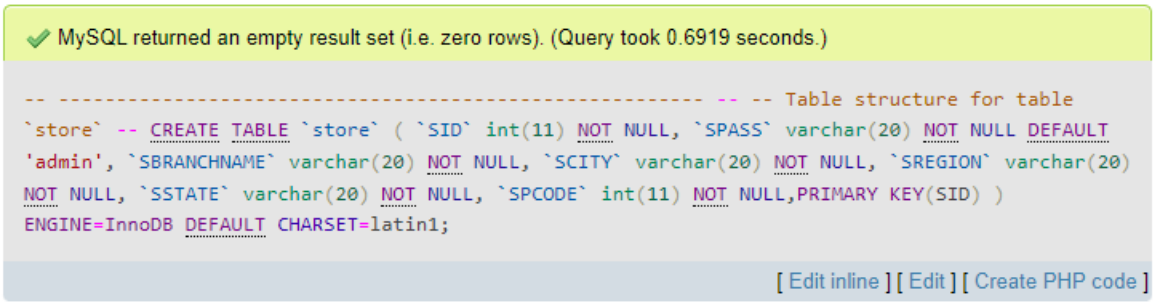


Table structure for table `sales`

QUERY:

```
-- Table structure for table `sales`

CREATE TABLE `sales` (
  `SALESID` int(11) NOT NULL,
  `SDATE` date NOT NULL,
  `SCOST` int(11) NOT NULL,
  `SID` int(11) NOT NULL,PRIMARY KEY(SALESID)
) ENGINE=InnoDB DEFAULT CHARSET=latin1;

alter table sales  add constraint sales_fk foreign key(SID) references store(SID)
on delete CASCADE;
```

OUTPUT:

MySQL returned an empty result set (i.e. zero rows). (Query took 0.3426 seconds.)

```
-- ------------------------------------------------------- -- -- Table structure for table `sales` -- CREATE TABLE `sales` ( `SALESID` int(11) NOT NULL, `SDATE` date NOT NULL, `SCOST` int(11) NOT NULL, `SID` int(11) NOT NULL,PRIMARY KEY(SALESID) ) ENGINE=InnoDB DEFAULT CHARSET=latin1;
```

[Edit inline] [Edit] [Create PHP code]

MySQL returned an empty result set (i.e. zero rows). (Query took 0.8502 seconds.)

```
alter table sales add constraint sales_fk foreign key(SID) references store(SID) on delete CASCADE;
```

[Edit inline] [Edit] [Create PHP code]

Table structure for table `stock`

QUERY:

```
-- Table structure for table `stock`

CREATE TABLE `stock` (
  `CRY` varchar(20) NOT NULL,
  `SCRY` varchar(20) NOT NULL,
  `QUANT` int(11) NOT NULL,
  `SID` int(11) NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=latin1;

alter table stock  add constraint stock_fk foreign key(SID) references store(SID) on delete CASCADE;
```

OUTPUT:

MySQL returned an empty result set (i.e. zero rows). (Query took 0.5035 seconds.)

```
-- ------------------------------------------------------- -- -- Table structure for table `stock` -- CREATE TABLE `stock` ( `CRY` varchar(20) NOT NULL, `SCRY` varchar(20) NOT NULL, `QUANT` int(11) NOT NULL, `SID` int(11) NOT NULL ) ENGINE=InnoDB DEFAULT CHARSET=latin1;
```

[Edit inline] [Edit] [Create PHP code]

MySQL returned an empty result set (i.e. zero rows). (Query took 1.9358 seconds.)

```
alter table stock add constraint stock_fk foreign key(SID) references store(SID) on delete CASCADE;
```

[Edit inline] [Edit] [Create PHP code]

Table structure for table `st_orders`

QUERY:

```
-- Table structure for table `st_orders`

CREATE TABLE `st_orders` (
  `SID` int(11) NOT NULL,
  `ORDID` int(11) NOT NULL,
  `DID` int(11) NOT NULL,
  `ORDDATE` date NOT NULL,
  `PMYSTAT` varchar(20) NOT NULL DEFAULT 'PENDING',
  `SHPMODE` varchar(20) NOT NULL DEFAULT 'Normal',
  `SHPSTAT` varchar(20) NOT NULL DEFAULT 'PENDING',
  PRIMARY KEY(ORDID)
) ENGINE=InnoDB DEFAULT CHARSET=latin1;

alter table st_orders  add constraint strodr1_fk foreign key(SID) references
store(SID) on delete CASCADE;
alter table st_orders  add constraint strodr2_fk foreign key(DID) references
Dist(DID) on delete CASCADE;
```

OUTPUT:

MySQL returned an empty result set (i.e. zero rows). (Query took 0.3772 seconds.)

```
-- -------------------------------------------------------- -- -- Table structure for table
`st_orders` -- CREATE TABLE `st_orders` ( `SID` int(11) NOT NULL, `ORDID` int(11) NOT NULL,
`DID` int(11) NOT NULL, `ORDDATE` date NOT NULL, `PMYSTAT` varchar(20) NOT NULL DEFAULT
'PENDING', `SHPMODE` varchar(20) NOT NULL DEFAULT 'Normal', `SHPSTAT` varchar(20) NOT NULL
DEFAULT 'PENDING', PRIMARY KEY(ORDID) ) ENGINE=InnoDB DEFAULT CHARSET=latin1;
```

[Edit inline] [Edit] [Create PHP code]

MySQL returned an empty result set (i.e. zero rows). (Query took 1.0435 seconds.)

```
alter table st_orders add constraint strodr1_fk foreign key(SID) references store(SID) on delete
CASCADE;
```

MySQL returned an empty result set (i.e. zero rows). (Query took 1.3219 seconds.)

```
alter table st_orders add constraint strodr2_fk foreign key(DID) references Dist(DID) on delete
CASCADE;
```

[Edit inline] [Edit] [Create PHP code]

----------Section Break (Next Page) ----------

Populating the Database

INSERTIONS

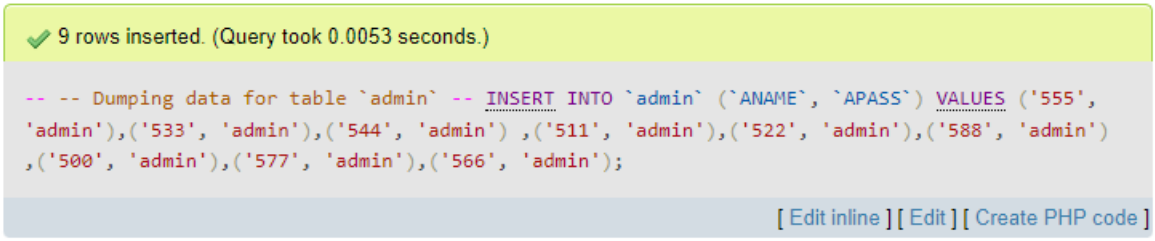
Dumping data for table `admin`

QUERY:

```
-- Dumping data for table `admin`

INSERT INTO `admin` (`ANAME`, `APASS`) VALUES
('555', 'admin'),('533', 'admin'),('544', 'admin')
,('511', 'admin'),('522', 'admin'),('588', 'admin')
,('500', 'admin'),('577', 'admin'),('566', 'admin');
```

OUTPUT:



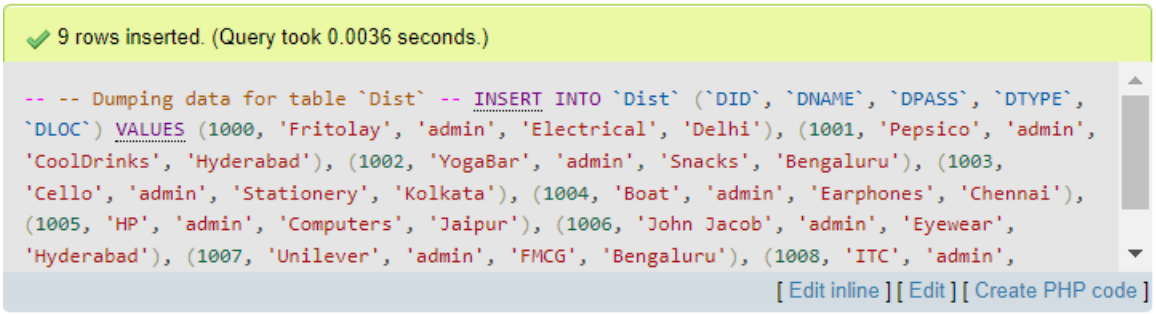
Dumping data for table `dist`

QUERY:

```
-- Dumping data for table `Dist`

INSERT INTO `Dist` (`DID`, `DNAME`, `DPASS`, `DTYPE`, `DLOC`) VALUES
(1000, 'Fritolay', 'admin', 'Electrical', 'Delhi'),
(1001, 'Pepsico', 'admin', 'CoolDrinks', 'Hyderabad'),
(1002, 'YogaBar', 'admin', 'Snacks', 'Bengaluru'),
(1003, 'Cello', 'admin', 'Stationery', 'Kolkata'),
(1004, 'Boat', 'admin', 'Earphones', 'Chennai'),
(1005, 'HP', 'admin', 'Computers', 'Jaipur'),
(1006, 'John Jacob', 'admin', 'Eyewear', 'Hyderabad'),
(1007, 'Unilever', 'admin', 'FMCG', 'Bengaluru'),
(1008, 'ITC', 'admin', 'FMCG', 'Chennai');
```

OUTPUT:



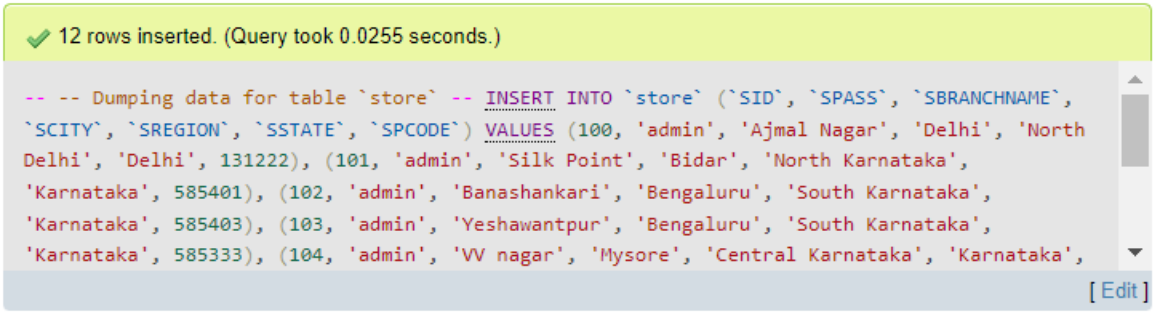
Dumping data for table `store`

QUERY:

```
-- Dumping data for table `store`

INSERT INTO `store` (`SID`, `SPASS`, `SBRANCHNAME`, `SCITY`, `SREGION`, `SSTATE`, `SPCODE`) VALUES
(100, 'admin', 'Ajmal Nagar', 'Delhi', 'North Delhi', 'Delhi', 131222),
(101, 'admin', 'Silk Point', 'Bidar', 'North Karnataka', 'Karnataka', 585401),
(102, 'admin', 'Banashankari', 'Bengaluru', 'South Karnataka', 'Karnataka', 585403),
(103, 'admin', 'Yeshawantpur', 'Bengaluru', 'South Karnataka', 'Karnataka', 585333),
(104, 'admin', 'VV nagar', 'Mysore', 'Central Karnataka', 'Karnataka', 585233),
(105, 'admin', 'Batra Nagar', 'Allahabad', 'North UP', 'UP', 485401),
(106, 'admin', 'Shanti Layout', 'Jaipur', 'South Rajasthan', 'Rajasthan', 385401),
(107, 'admin', 'Amar Colony', 'Kozhikode', 'North Kerala', 'Kerala', 457401),
(108, 'admin', 'Nippani', 'Gulbarga', 'North Karnataka', 'Karnataka', 595401),
(109, 'admin', 'Bhuvana Colony', 'Bengaluru', 'South Karnataka', 'Karnataka', 585567),
(110, 'admin', 'Ganesh Nagar', 'Shimla', 'North Kashmir', 'Kashmir', 591401),
(111, 'admin', 'Tulasi Layout', 'Shillong', 'North Meghalaya', 'Meghalaya', 185234);
```

OUTPUT:



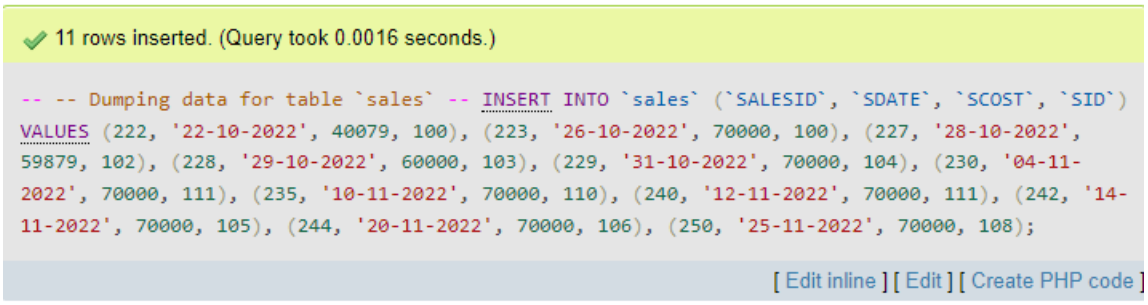
Dumping data for table `sales`

QUERY:

```
-- Dumping data for table `sales`

INSERT INTO `sales` (`SALESID`, `SDATE`, `SCOST`, `SID`) VALUES
(222, '2022-11-22', 40079, 100),
(223, '2022-11-22', 70000, 100),
(227, '2022-11-28', 59879, 102),
(228, '2022-11-25', 60000, 103),
(229, '2022-11-17', 70000, 104),
(230, '2022-11-27', 70000, 111),
(235, '2022-11-25', 70000, 110),
(240, '2022-11-23', 70000, 111),
(242, '2022-11-20', 70000, 105),
(244, '2022-11-24', 70000, 106),
(250, '2022-11-28', 70000, 108);
```

OUTPUT:

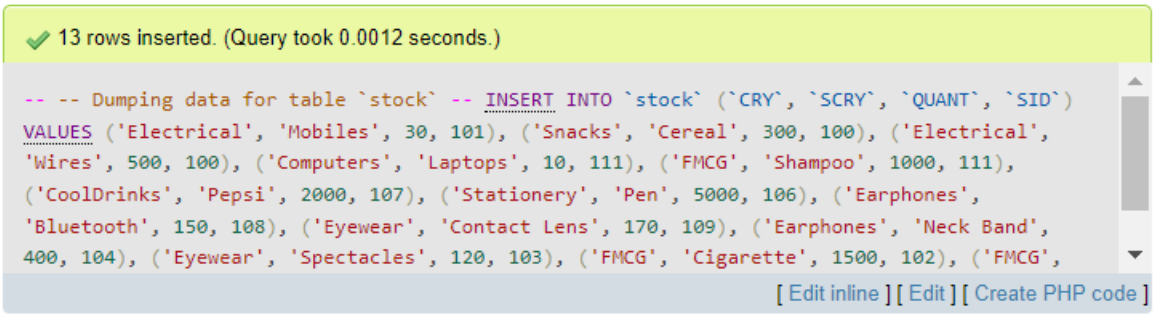


Dumping data for table `stock`
QUERY:

```
-- Dumping data for table `stock`

INSERT INTO `stock` (`CRY`, `SCRY`, `QUANT`, `SID`) VALUES
('Electrical', 'Mobiles', 30, 101),
('Snacks', 'Cereal', 300, 100),
('Electrical', 'Wires', 500, 100),
('Computers', 'Laptops', 10, 111),
('FMCG', 'Shampoo', 1000, 111),
('CoolDrinks', 'Pepsi', 2000, 107),
('Stationery', 'Pen', 5000, 106),
('Earphones', 'Bluetooth', 150, 108),
('Eyewear', 'Contact Lens', 170, 109),
('Earphones', 'Neck Band', 400, 104),
('Eyewear', 'Spectacles', 120, 103),
('FMCG', 'Cigarette', 1500, 102),
('FMCG', 'Soap', 3000, 105);
```

OUTPUT:

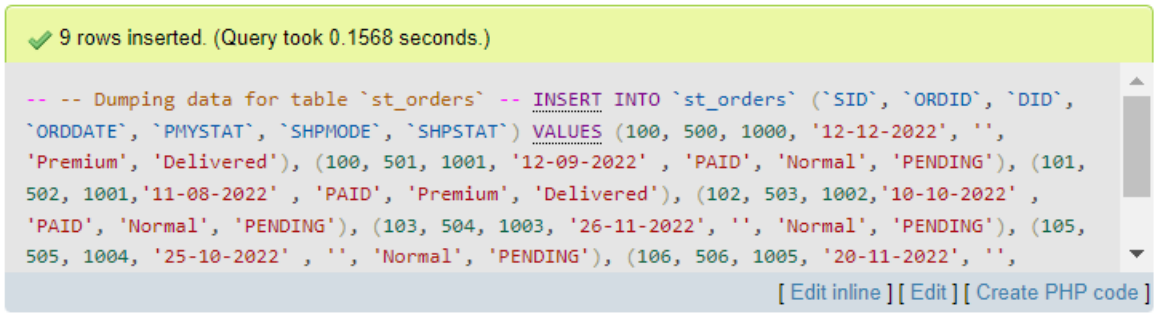


Dumping data for table `st_orders`
QUERY:

```
-- Dumping data for table `st_orders`

INSERT INTO `st_orders` (`SID`, `ORDID`, `DID`, `ORDDATE`, `PMYSTAT`, `SHPMODE`,
`SHPSTAT`) VALUES
(100, 500, 1000, '2022-11-15', '', 'Premium', 'Delivered'),
(100, 501, 1001, '2022-11-28'  , 'PAID', 'Normal', 'PENDING'),
(101, 502, 1001,'2022-11-27' , 'PAID', 'Premium', 'Delivered'),
(102, 503, 1002,'2022-11-18' , 'PAID', 'Normal', 'PENDING'),
(103, 504, 1003, '2022-11-25', '', 'Normal', 'PENDING'),
(105, 505, 1004, '2022-11-22' , '', 'Normal', 'PENDING'),
(106, 506, 1005, '2022-11-26', '', 'Normal', 'PENDING'),
(105, 507, 1006, '2022-11-28', '', 'Normal', 'PENDING'),
(111, 508, 1008, '2022-11-28', '', 'Normal', 'PENDING');
```

OUTPUT:



----------Section Break (Next Page) ----------

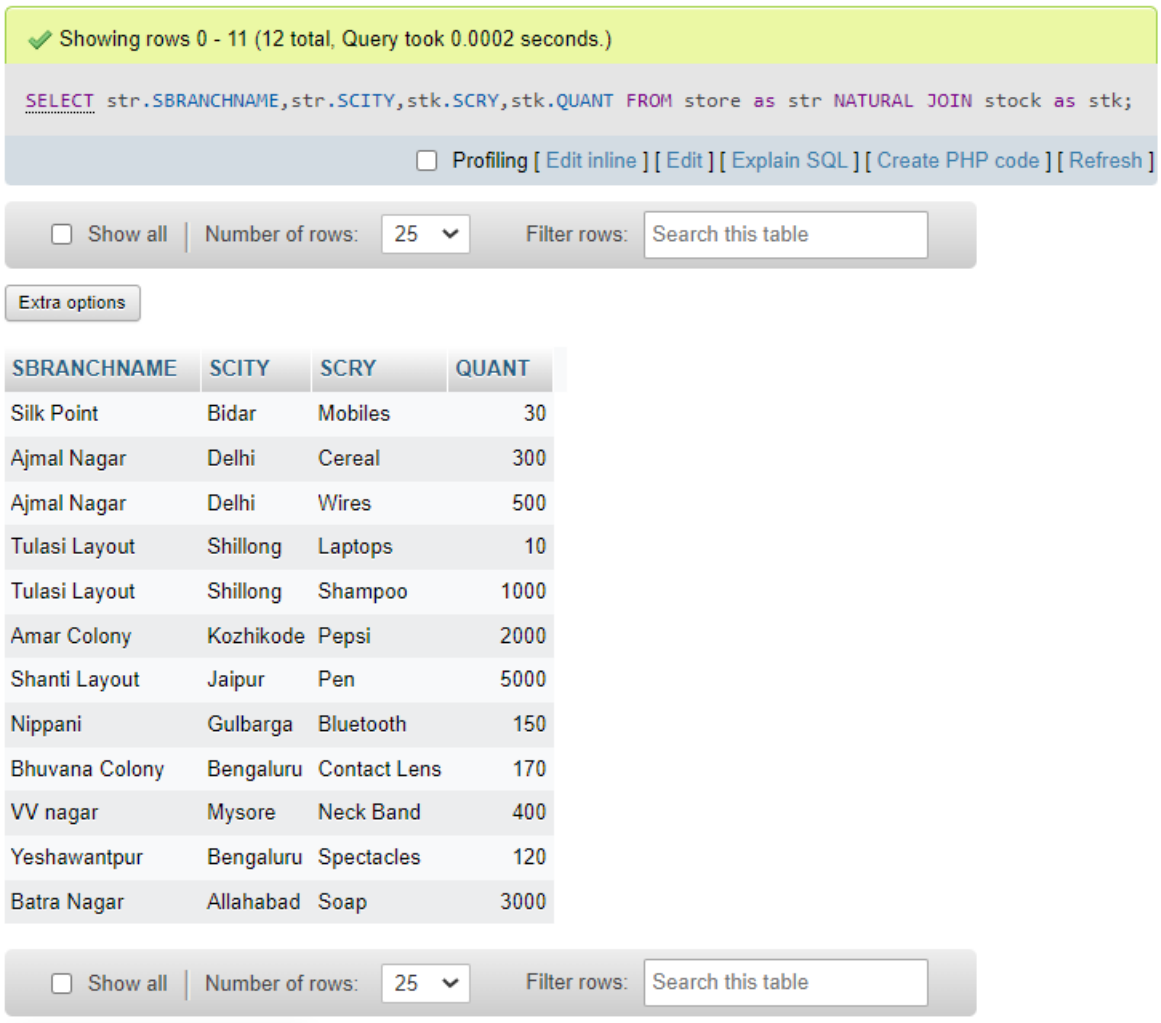
Join Queries

Showcase at least 4 join queries
Write the query in English Language, Show the equivalent SQL statement and also a screenshot of the query and the results.

(Note: Store with SID 102 was deleted which resulted in deletion of a few other related entries. The following operations are performed on remaining data in database.)

1. **Natural Join:**
   Retrieve all the branch name, city, specific category, quantity from store and stock

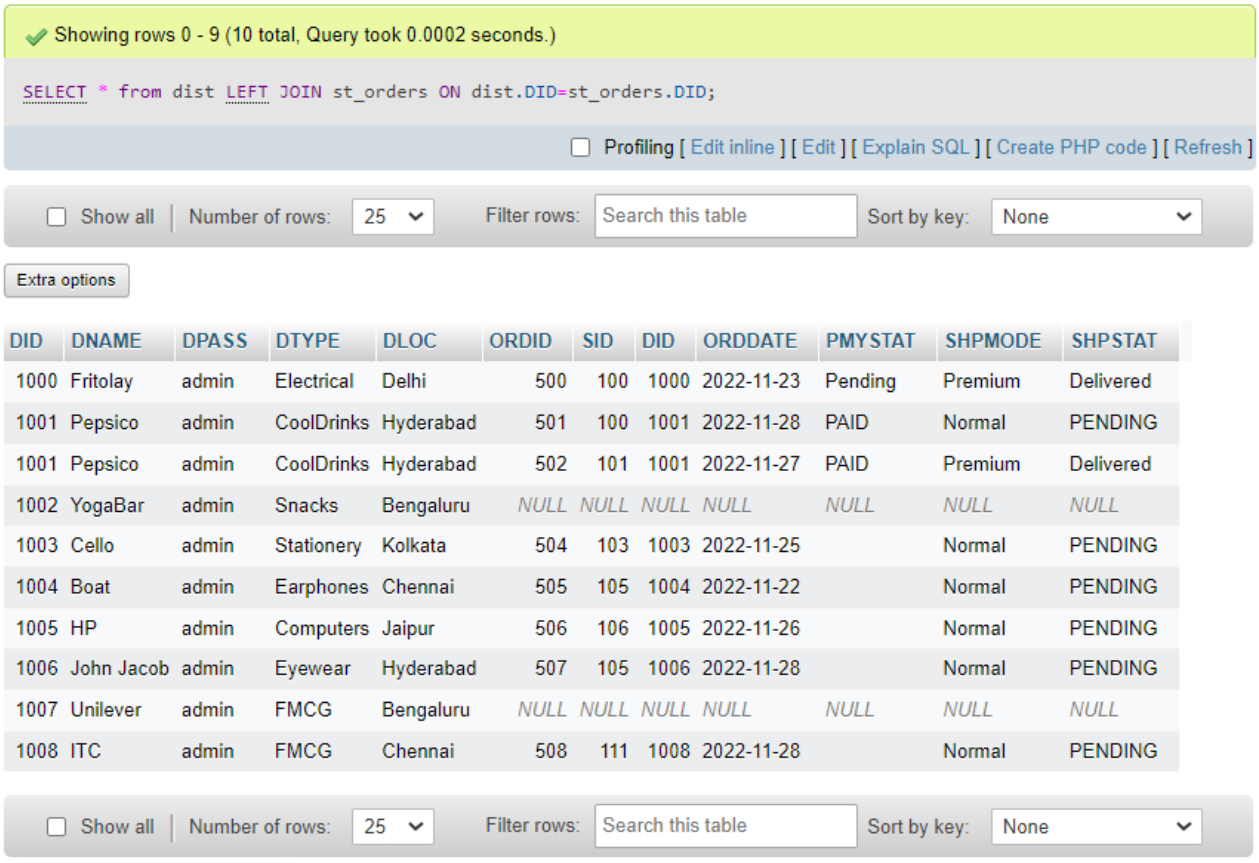
QUERY and OUTPUT:



SBRANCHNAME	SCITY	SCRY	QUANT
Silk Point	Bidar	Mobiles	30
Ajmal Nagar	Delhi	Cereal	300
Ajmal Nagar	Delhi	Wires	500
Tulasi Layout	Shillong	Laptops	10
Tulasi Layout	Shillong	Shampoo	1000
Amar Colony	Kozhikode	Pepsi	2000
Shanti Layout	Jaipur	Pen	5000
Nippani	Gulbarga	Bluetooth	150
Bhuvana Colony	Bengaluru	Contact Lens	170
VV nagar	Mysore	Neck Band	400
Yeshawantpur	Bengaluru	Spectacles	120
Batra Nagar	Allahabad	Soap	3000

2. **Left Join:**
   Retrieve all details from dist and only those which have matching DID from st_orders.

QUERY and OUTPUT:



Showing rows 0 - 9 (10 total, Query took 0.0002 seconds.)

```
SELECT * from dist LEFT JOIN st_orders ON dist.DID=st_orders.DID;
```

Profiling [Edit inline] [Edit] [Explain SQL] [Create PHP code] [Refresh]

Show all | Number of rows: 25 Filter rows: Search this table Sort by key: None

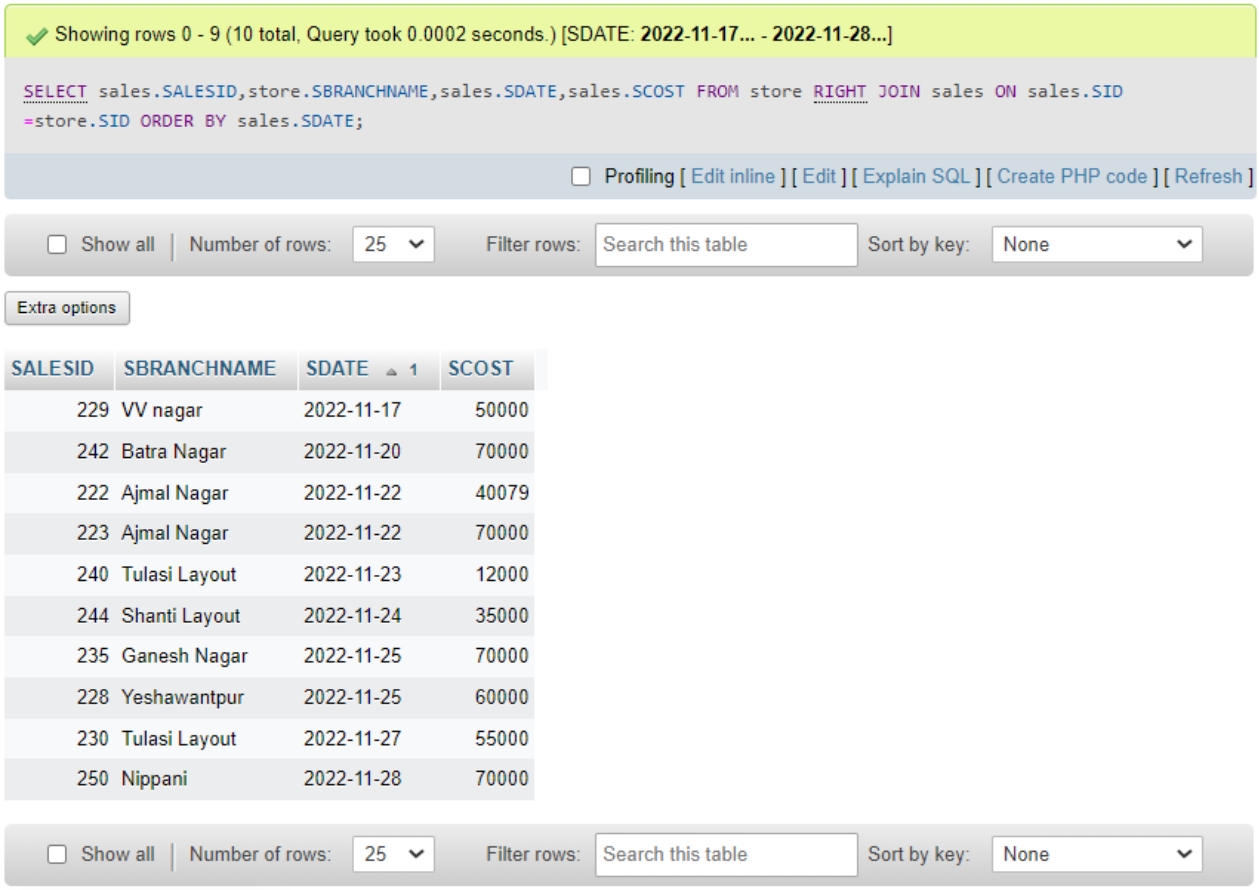
Extra options

DID	DNAME	DPASS	DTYPE	DLOC	ORDID	SID	DID	ORDDATE	PMYSTAT	SHPMODE	SHPSTAT
1000	Fritolay	admin	Electrical	Delhi	500	100	1000	2022-11-23	Pending	Premium	Delivered
1001	Pepsico	admin	CoolDrinks	Hyderabad	501	100	1001	2022-11-28	PAID	Normal	PENDING
1001	Pepsico	admin	CoolDrinks	Hyderabad	502	101	1001	2022-11-27	PAID	Premium	Delivered
1002	YogaBar	admin	Snacks	Bengaluru	*NULL*	*NULL*	*NULL*	*NULL*	*NULL*	*NULL*	*NULL*
1003	Cello	admin	Stationery	Kolkata	504	103	1003	2022-11-25		Normal	PENDING
1004	Boat	admin	Earphones	Chennai	505	105	1004	2022-11-22		Normal	PENDING
1005	HP	admin	Computers	Jaipur	506	106	1005	2022-11-26		Normal	PENDING
1006	John Jacob	admin	Eyewear	Hyderabad	507	105	1006	2022-11-28		Normal	PENDING
1007	Unilever	admin	FMCG	Bengaluru	*NULL*	*NULL*	*NULL*	*NULL*	*NULL*	*NULL*	*NULL*
1008	ITC	admin	FMCG	Chennai	508	111	1008	2022-11-28		Normal	PENDING

Show all | Number of rows: 25 Filter rows: Search this table Sort by key: None

3. **Right Join:**
   Retrieve the sales ID, date, cost from sales table and Branch name from store table while keeping all entries of sales and only matching ones from store, also order by sales date.

QUERY and OUTPUT:



Showing rows 0 - 9 (10 total, Query took 0.0002 seconds.) [SDATE: **2022-11-17...** - **2022-11-28...**]

```
SELECT sales.SALESID,store.SBRANCHNAME,sales.SDATE,sales.SCOST FROM store RIGHT JOIN sales ON sales.SID
=store.SID ORDER BY sales.SDATE;
```

Profiling [Edit inline] [Edit] [Explain SQL] [Create PHP code] [Refresh]

Show all | Number of rows: 25 Filter rows: Search this table Sort by key: None

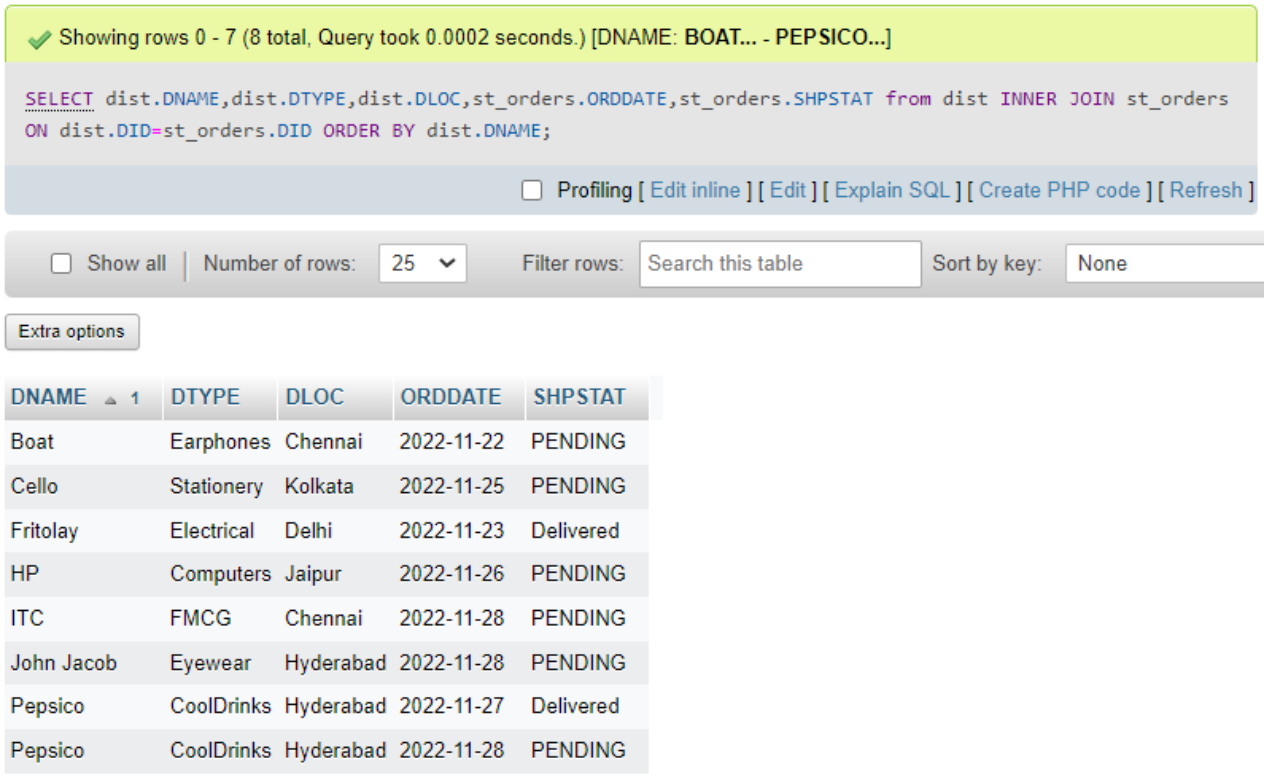
Extra options

SALESID	SBRANCHNAME	SDATE ▲ 1	SCOST
229	VV nagar	2022-11-17	50000
242	Batra Nagar	2022-11-20	70000
222	Ajmal Nagar	2022-11-22	40079
223	Ajmal Nagar	2022-11-22	70000
240	Tulasi Layout	2022-11-23	12000
244	Shanti Layout	2022-11-24	35000
235	Ganesh Nagar	2022-11-25	70000
228	Yeshawantpur	2022-11-25	60000
230	Tulasi Layout	2022-11-27	55000
250	Nippani	2022-11-28	70000

Show all | Number of rows: 25 Filter rows: Search this table Sort by key: None

4. **Inner join:**
   Retrieve dname, dtype, dloc from dist; orddate, shpstat from st_orders by inner join and order by dname.

QUERY and OUTPUT:



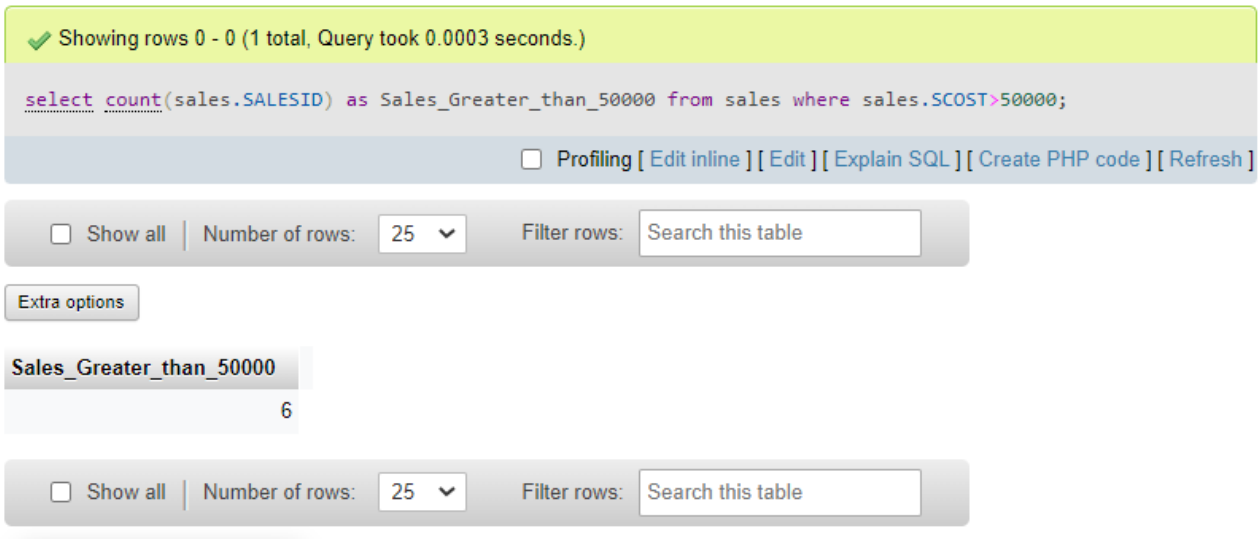
DNAME ▲ 1	DTYPE	DLOC	ORDDATE	SHPSTAT
Boat	Earphones	Chennai	2022-11-22	PENDING
Cello	Stationery	Kolkata	2022-11-25	PENDING
Fritolay	Electrical	Delhi	2022-11-23	Delivered
HP	Computers	Jaipur	2022-11-26	PENDING
ITC	FMCG	Chennai	2022-11-28	PENDING
John Jacob	Eyewear	Hyderabad	2022-11-28	PENDING
Pepsico	CoolDrinks	Hyderabad	2022-11-27	Delivered
Pepsico	CoolDrinks	Hyderabad	2022-11-28	PENDING

----------Section Break (Next Page) ----------

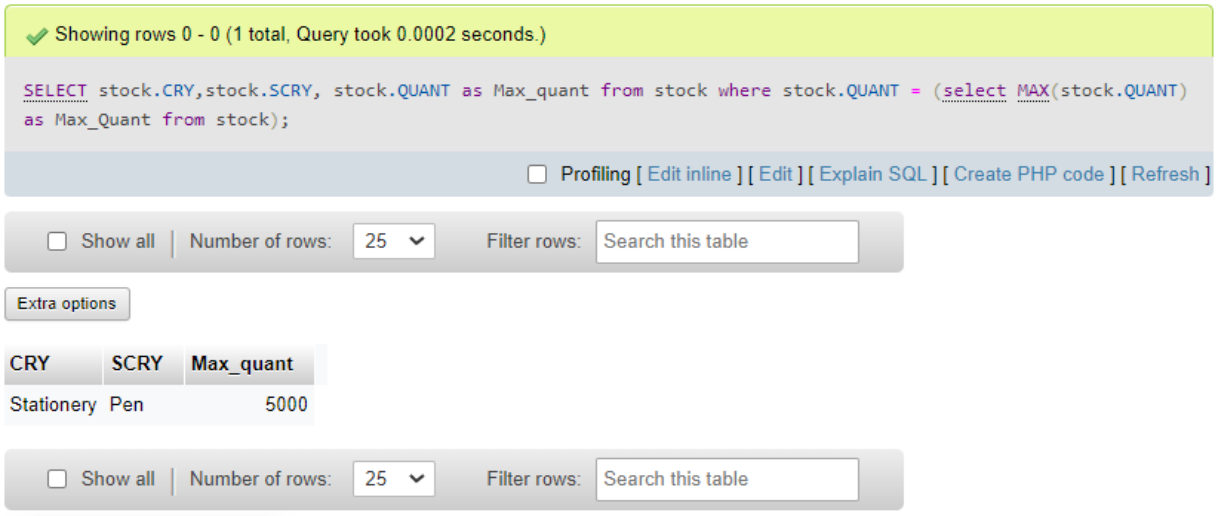
Aggregate Functions

Showcase at least 4 Aggregate function queries
Write the query in English Language, Show the equivalent SQL statement and also a screenshot of the query and the results

1. Retrieve the **count** of number of sales greater than 50,000
   QUERY and OUTPUT:

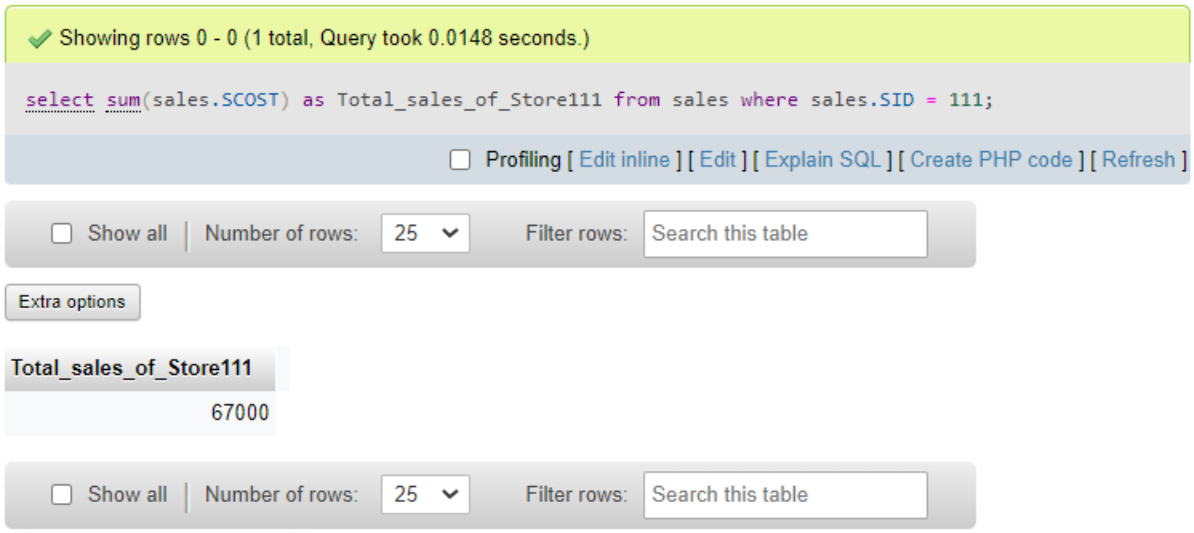


2. Display the Category, specific category and quantity of stock with **maximum** quantity.
   QUERY and OUTPUT:



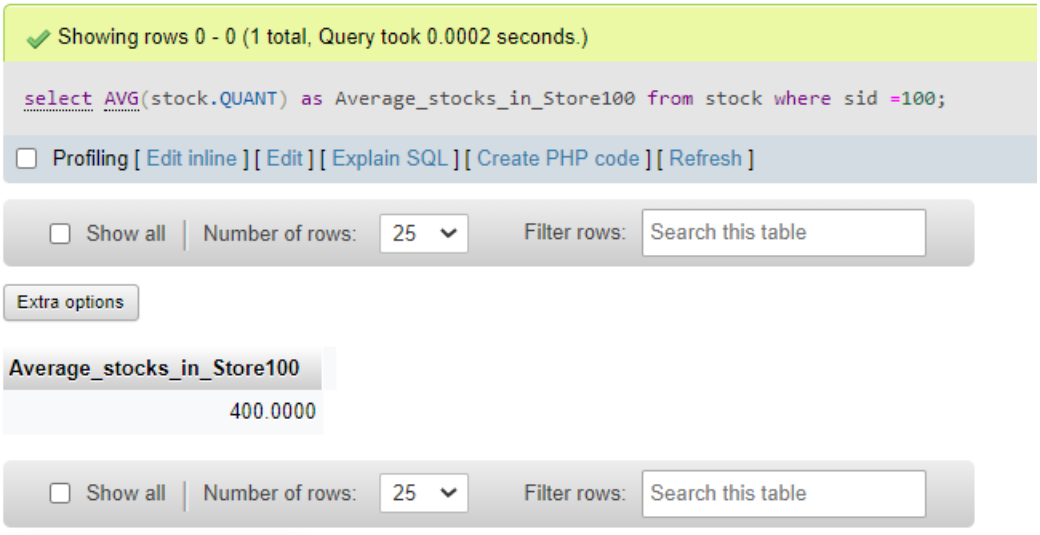
3. Retrieve **Sum** of sales made by store with SID=111.

   QUERY and OUTPUT:



4. Display **Average** stocks in store with SID=100.

   QUERY and OUTPUT:



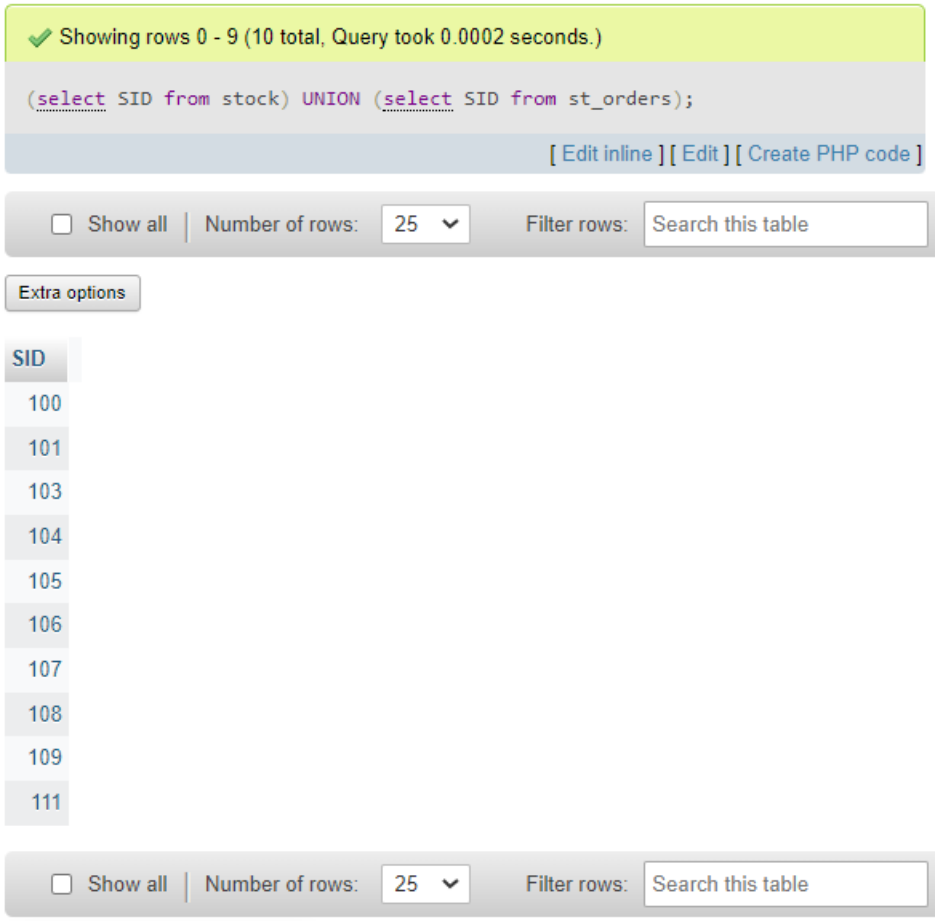
----------Section Break (Next Page) ----------

Set Operations

Showcase at least 4 Set Operations queries
Write the query in English Language, Show the equivalent SQL statement and also a screenshot of the query and the results.

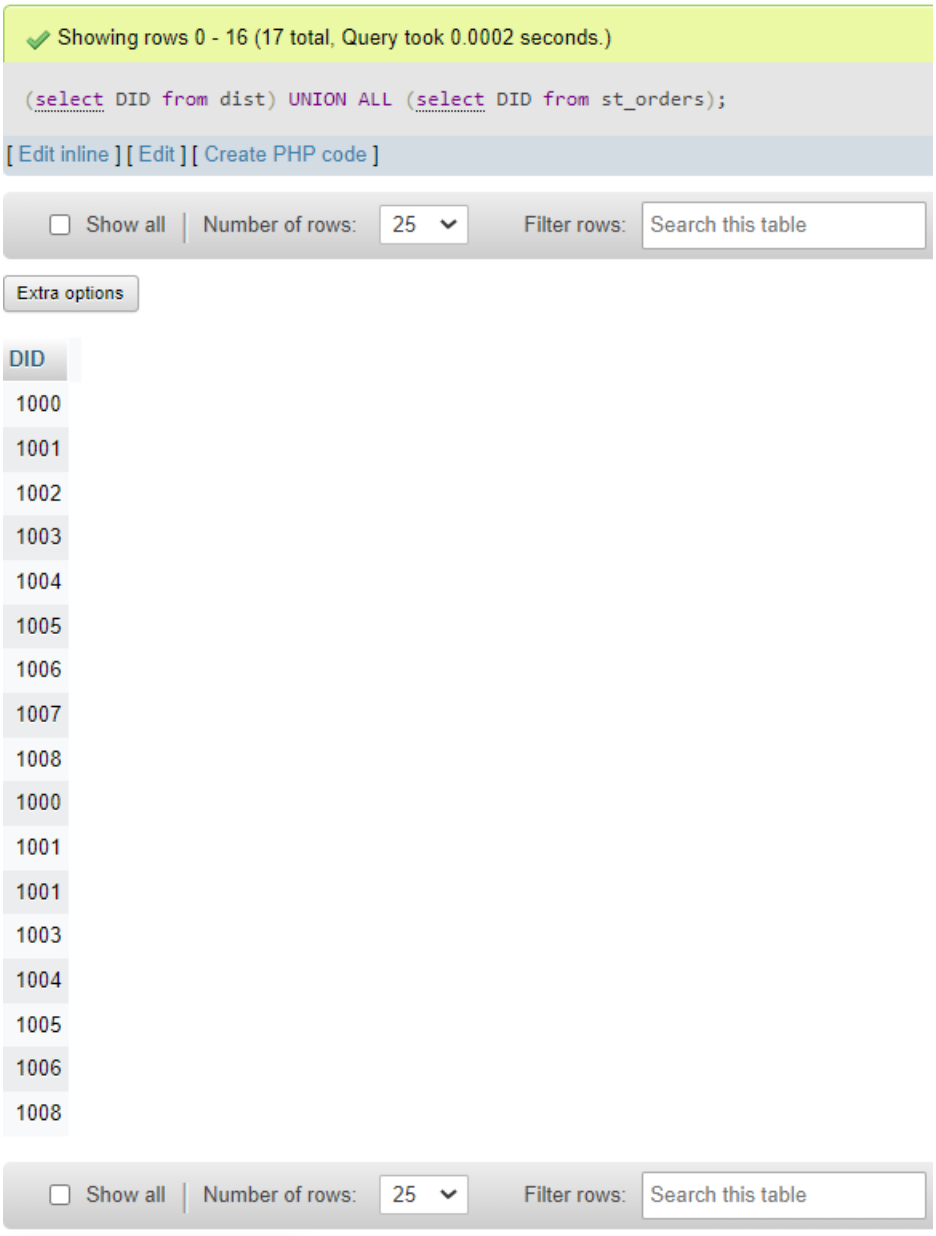
1. Perform **UNION** operation on SID of stock and st_orders.

QUERY and OUTPUT:



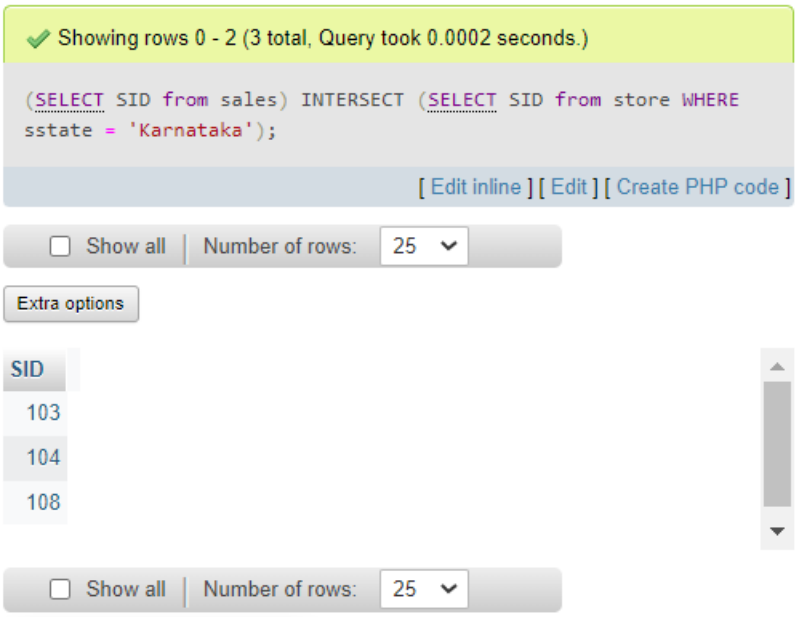
2. Perform **UNION ALL** operation on DID from dist and st_orders.

QUERY and OUTPUT:



3. To find **INTERSECTION** of SIDs from sales table and store table where state is Karnataka.

QUERY and OUTPUT:



4. To find which stores have not had any sales (**MINUS** operation)

QUERY and OUTPUT:

----------Section Break (Next Page) ----------

Functions and Procedures

Create a Function and Procedure. State the objective of the function / Procedure. Run and display the results.

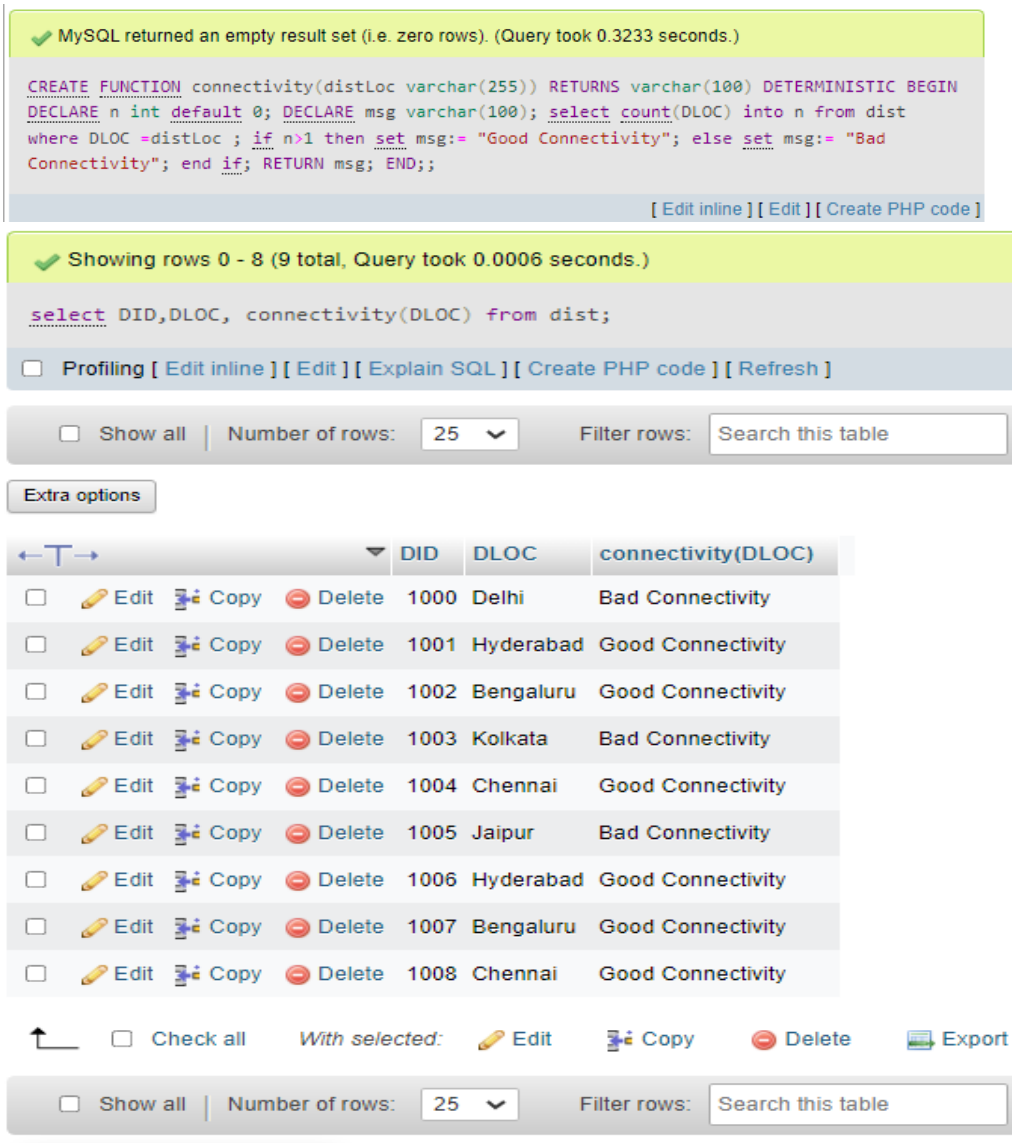
FUNCTION:

Function to show the connectivity of distributors.

OBJECTIVE:

We can say that a distributor has good connectivity if he distributes stocks in more than one warehouse in a single location. This is because he can connect to more stores or customers quickly through multiple warehouses.

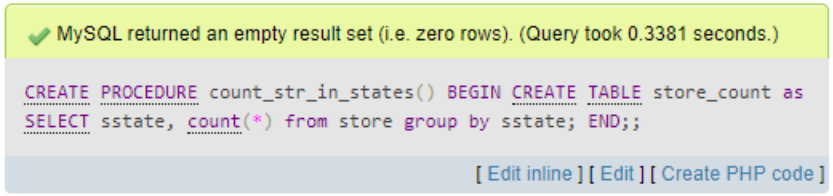
QUERY and OUTPUT:

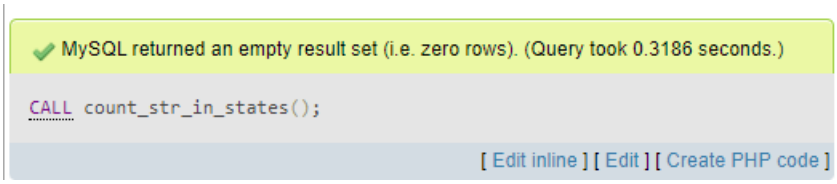


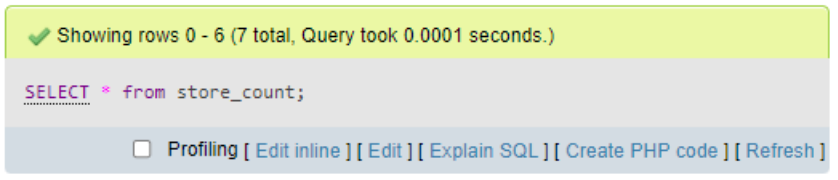
PROCEDURE:

Procedure to count the number of stores in each state.

QUERY and OUTPUT:







sstate	count(*)
Delhi	1
Karnataka	5
Kashmir	1
Kerala	1
Meghalaya	1
Rajasthan	1
UP	1

----------Section Break (Next Page) ----------

Triggers and Cursors

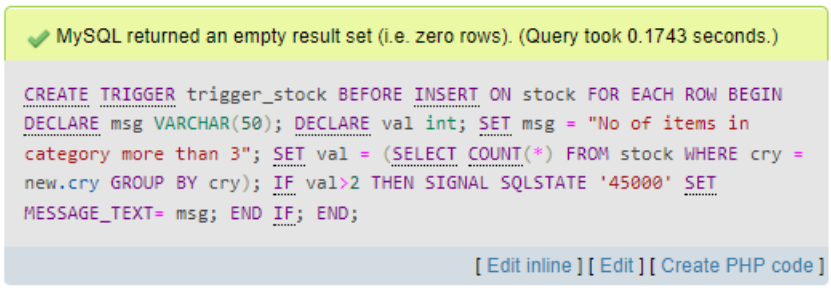
Create a Trigger and a Cursor. State the objective. Run and display the results.

TRIGGER:

Write a trigger on stock for when category of exceeds 3, display an error message.

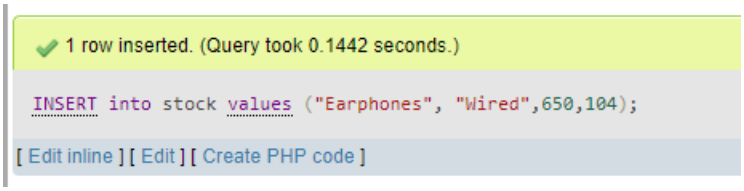
OBJECTIVE:
To show that each category in stock shouldn't exceed 3 as we want to ensure many other categories are included in the stock to get larger customer base.

QUERY and OUTPUT:



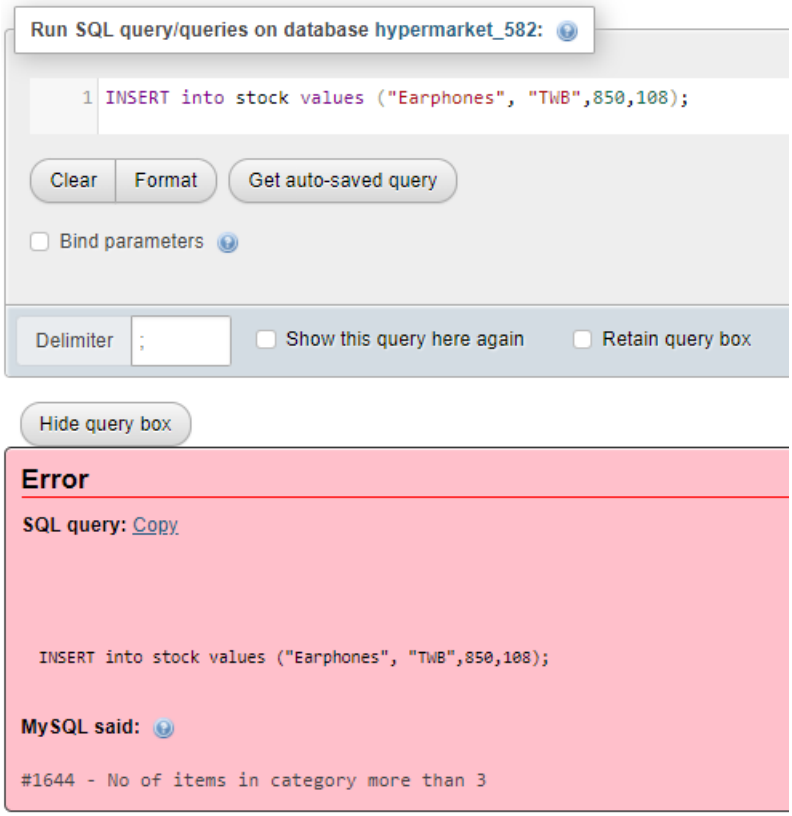
POSITIVE CASE:

QUERY and OUTPUT:



NEGATIVE CASE:

QUERY and OUTPUT:



CURSOR:

Create a Cursor to display the distributors in Hyderabad.

OBJECTIVE

A cursor acts as a pointer, hence the cursor used below tries to display the Details of distributors in Hyderabad.

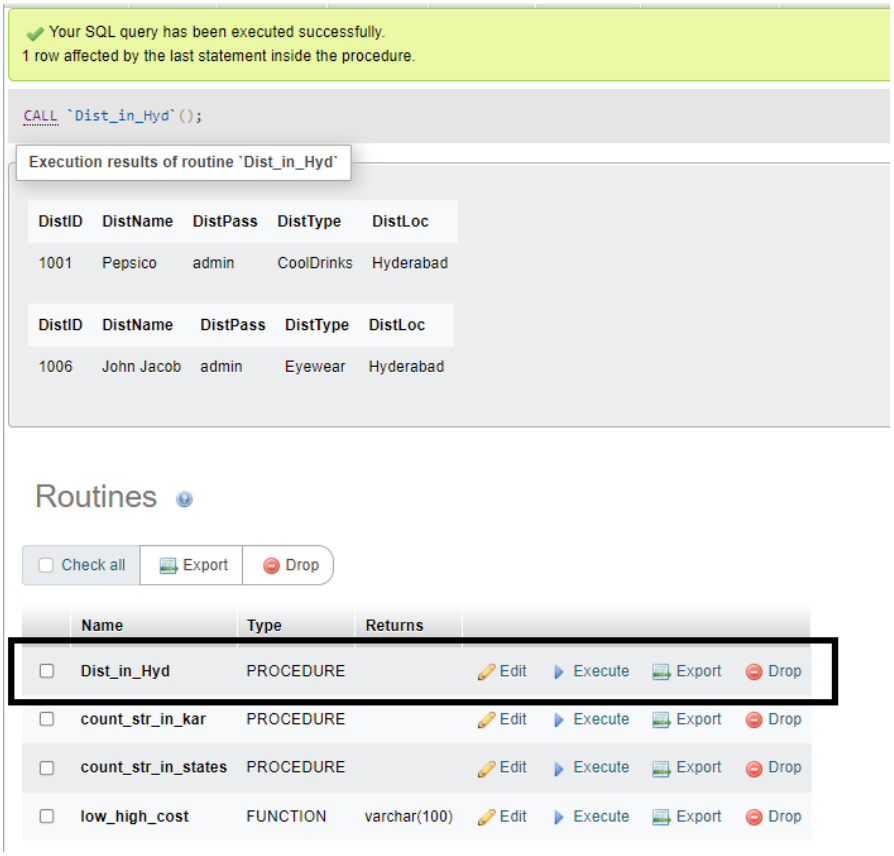
The cursor here is used within a procedure.

QUERY and OUTPUT:

MySQL returned an empty result set (i.e. zero rows). (Query took 0.3264 seconds.)

```
CREATE PROCEDURE Dist_in_Hyd() BEGIN DECLARE DistID int; DECLARE DistName varchar(255); DECLARE DistPass varchar(255); DECLARE DistType varchar(255); DECLARE DistLoc varchar(255); DECLARE exit_loop BOOLEAN DEFAULT FALSE; DECLARE d_cursor CURSOR FOR SELECT DID, DNAME, DPASS, DTYPE, DLOC FROM dist; DECLARE CONTINUE HANDLER FOR NOT FOUND SET exit_loop = TRUE; OPEN d_cursor; d_loop:LOOP FETCH FROM d_cursor INTO DistID,DistName,DistPass,DistType,DistLoc; IF exit_loop THEN LEAVE d_loop; END IF; IF DistLoc = "Hyderabad" THEN SELECT DistID,DistName,DistPass,DistType,DistLoc; END IF; END LOOP d_loop; CLOSE d_cursor; END;
```

[Edit inline] [Edit] [Create PHP code]



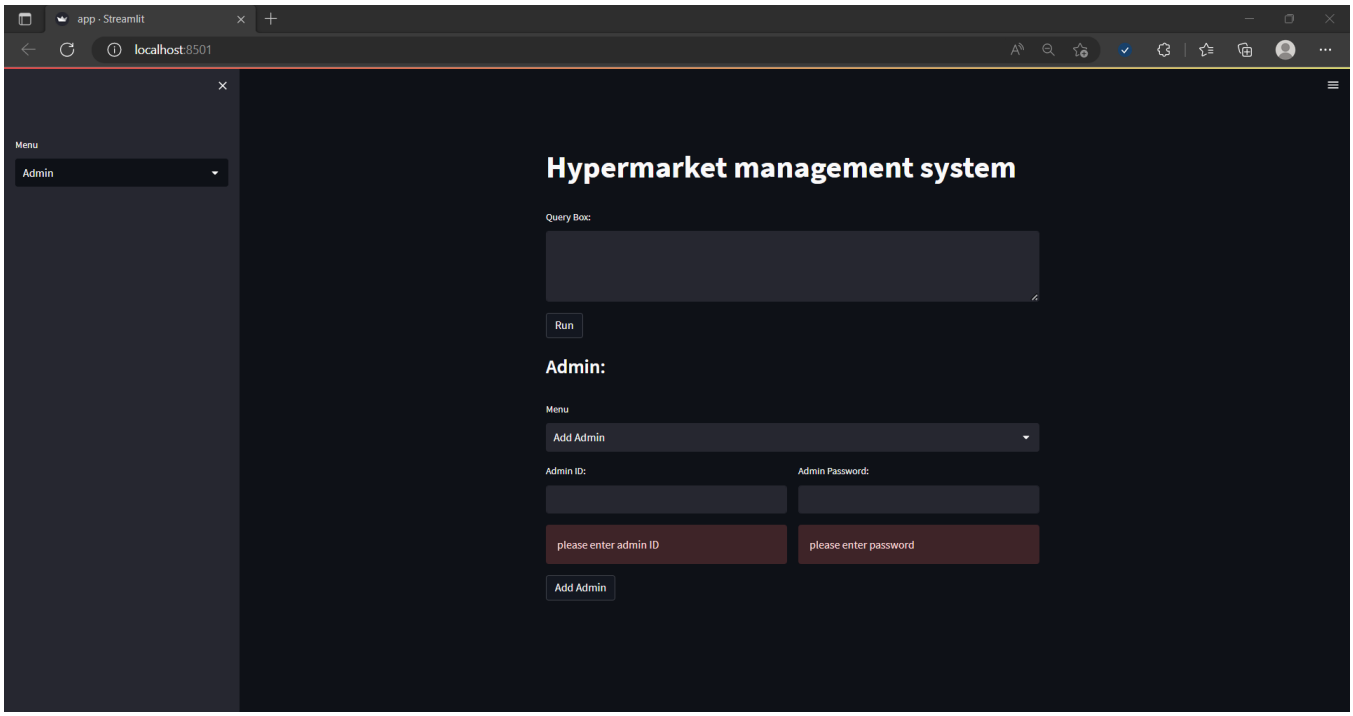
----------Section Break (Next Page) ----------

Developing a Frontend

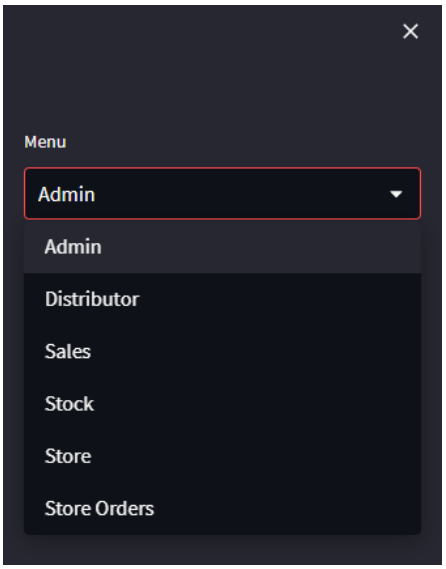
The frontend should support

1. Addition, Modification and Deletion of records from any chosen table
2. There should be a window to accept and run any SQL statement and display the result

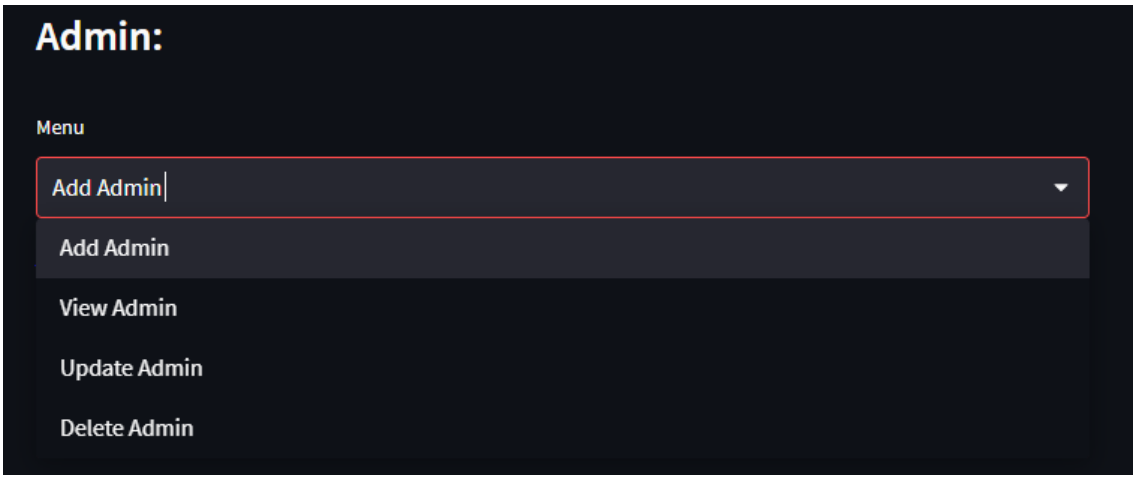
Overview:



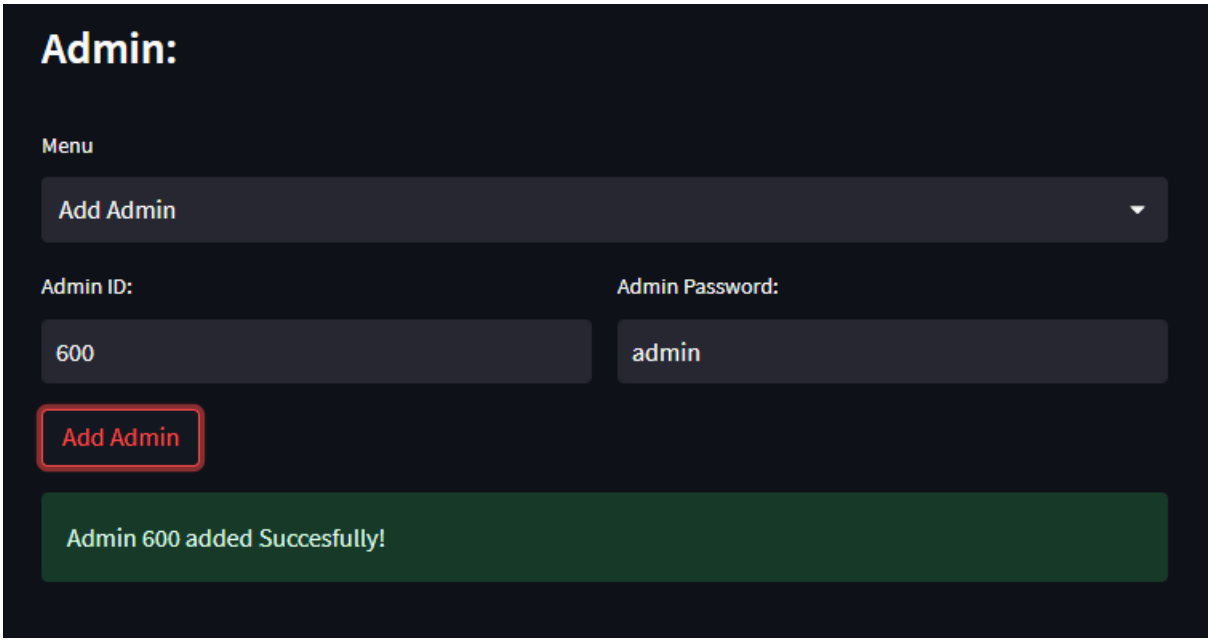
Menu to choose which table to perform CRUD operations:



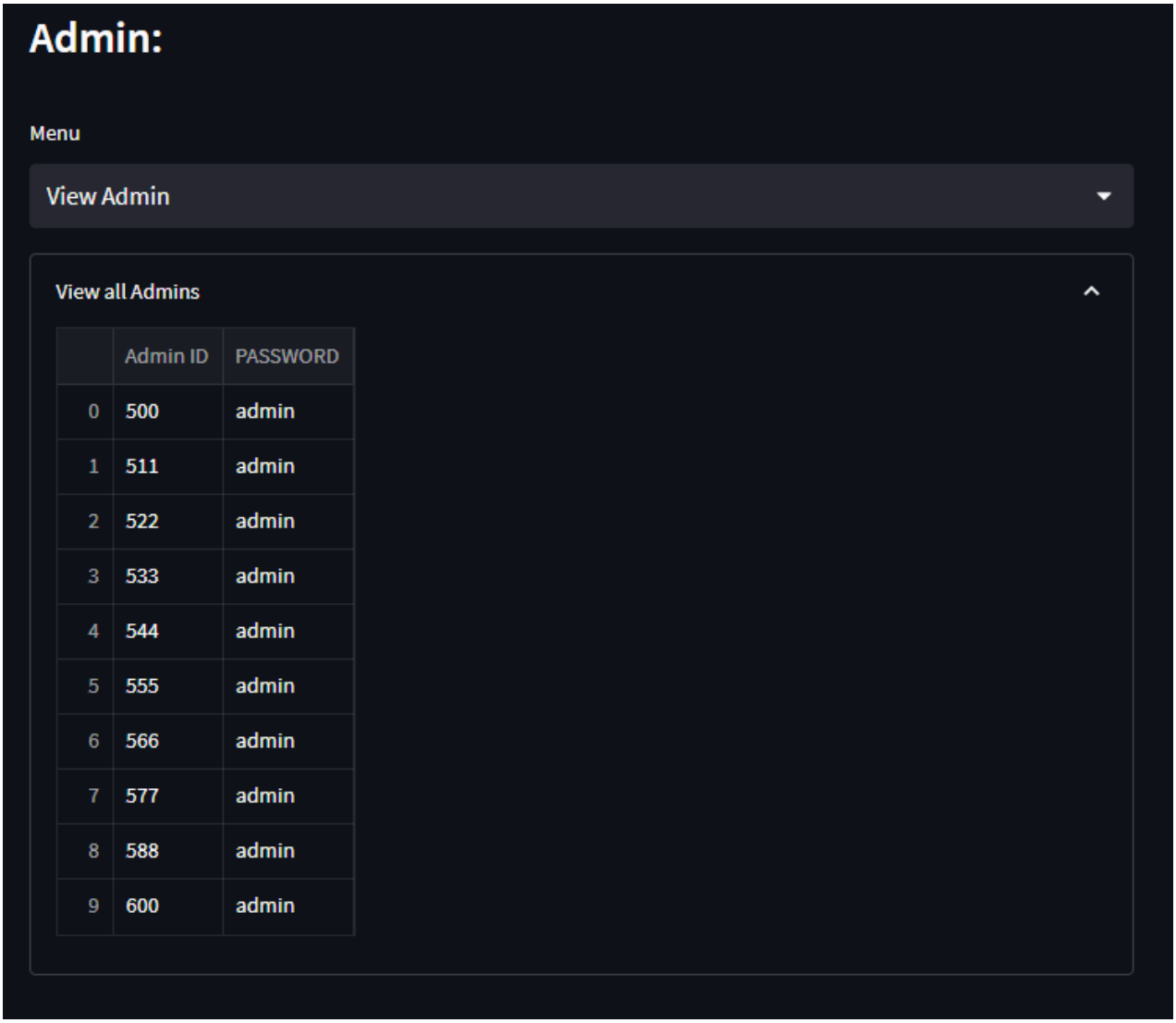
Menu to choose which CRUD operation to perform on selected table:



Adding Admin:

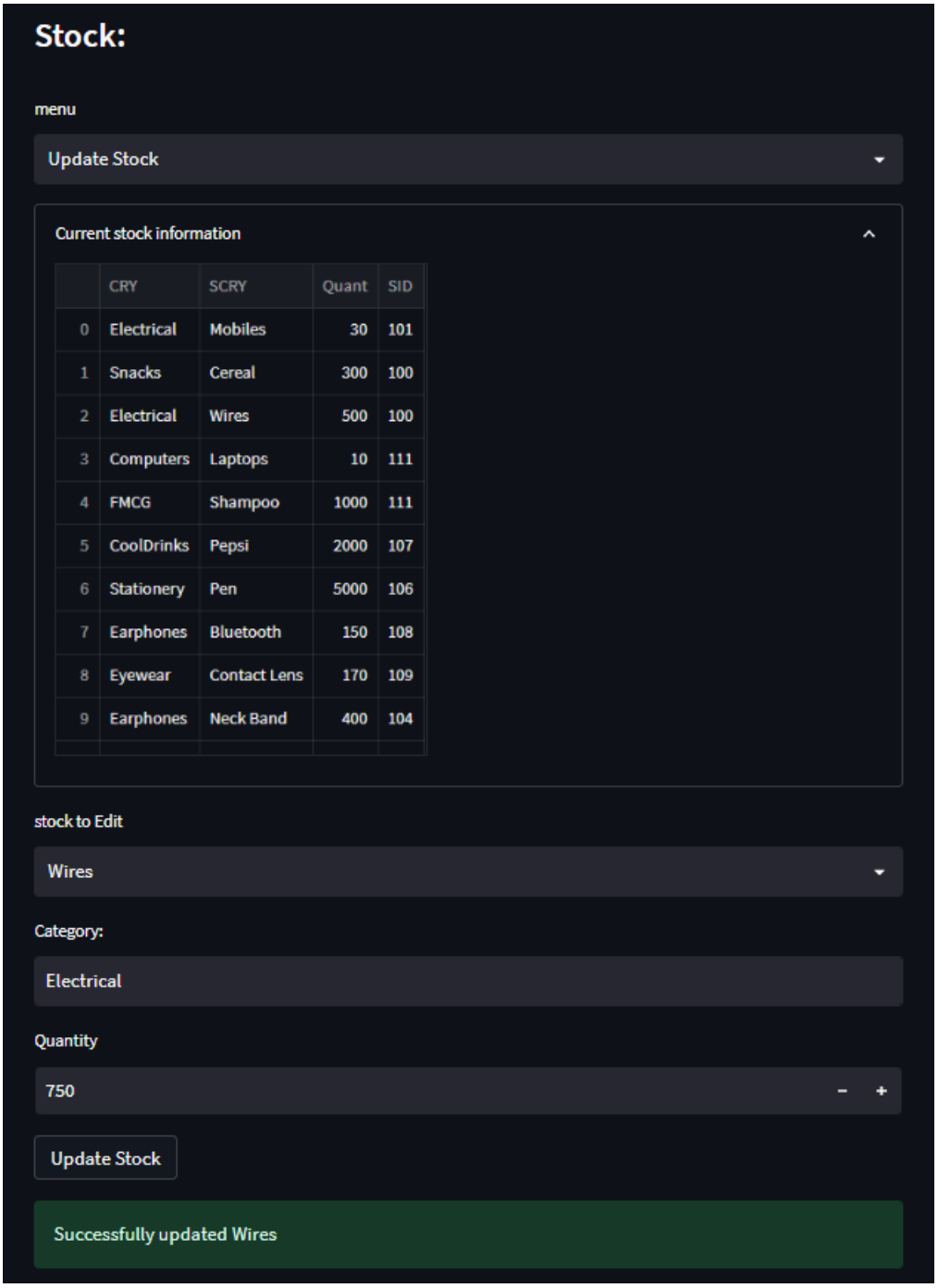


Viewing admin: (Note the newly added admin-600)



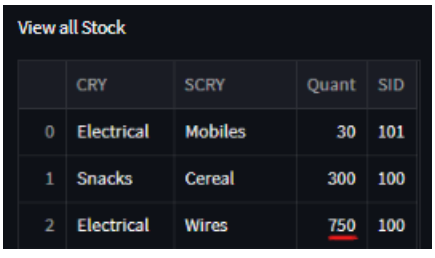
	Admin ID	PASSWORD
0	500	admin
1	511	admin
2	522	admin
3	533	admin
4	544	admin
5	555	admin
6	566	admin
7	577	admin
8	588	admin
9	600	admin

Updating a stock:



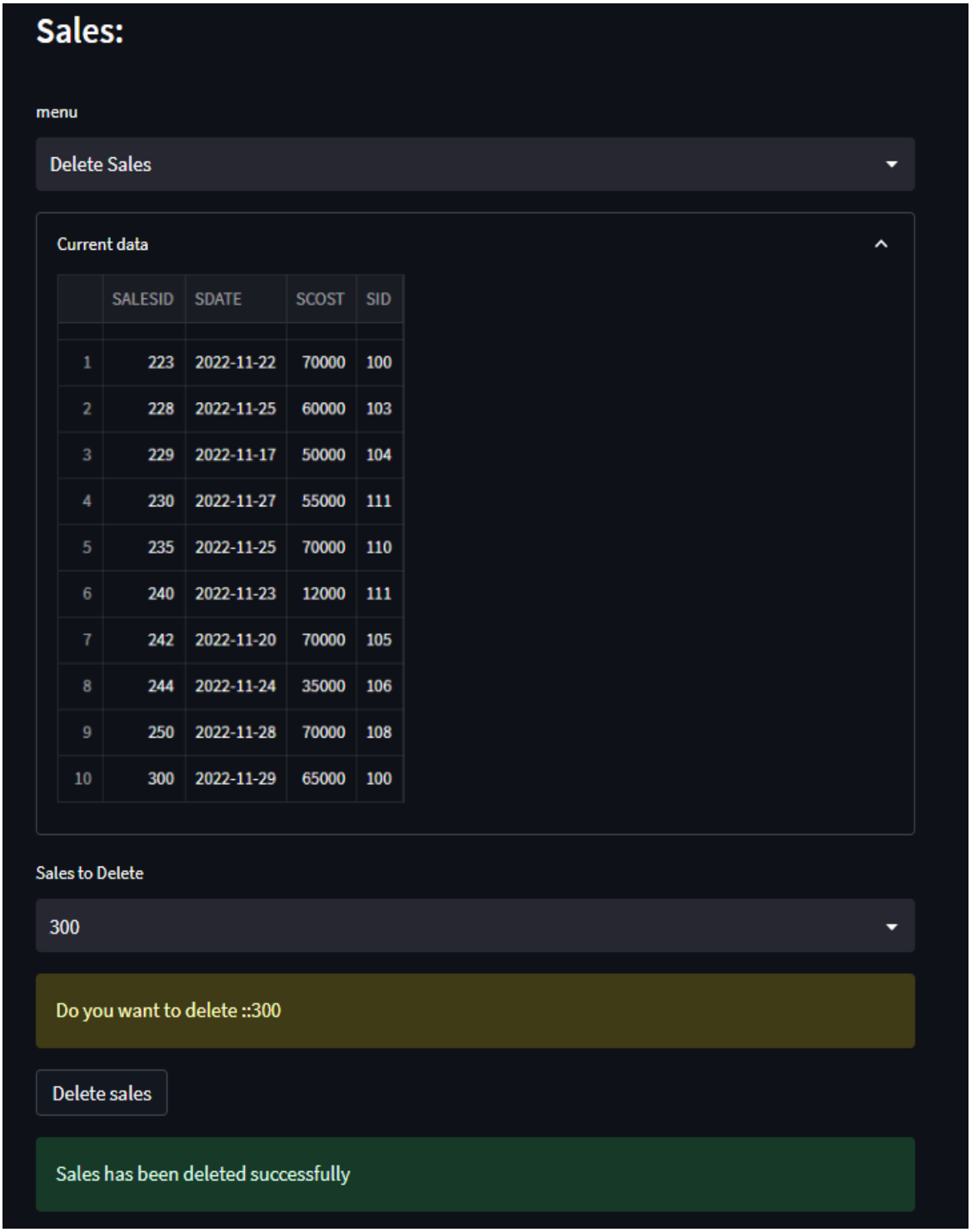
	CRY	SCRY	Quant	SID
0	Electrical	Mobiles	30	101
1	Snacks	Cereal	300	100
2	Electrical	Wires	500	100
3	Computers	Laptops	10	111
4	FMCG	Shampoo	1000	111
5	CoolDrinks	Pepsi	2000	107
6	Stationery	Pen	5000	106
7	Earphones	Bluetooth	150	108
8	Eyewear	Contact Lens	170	109
9	Earphones	Neck Band	400	104

Viewing the change:

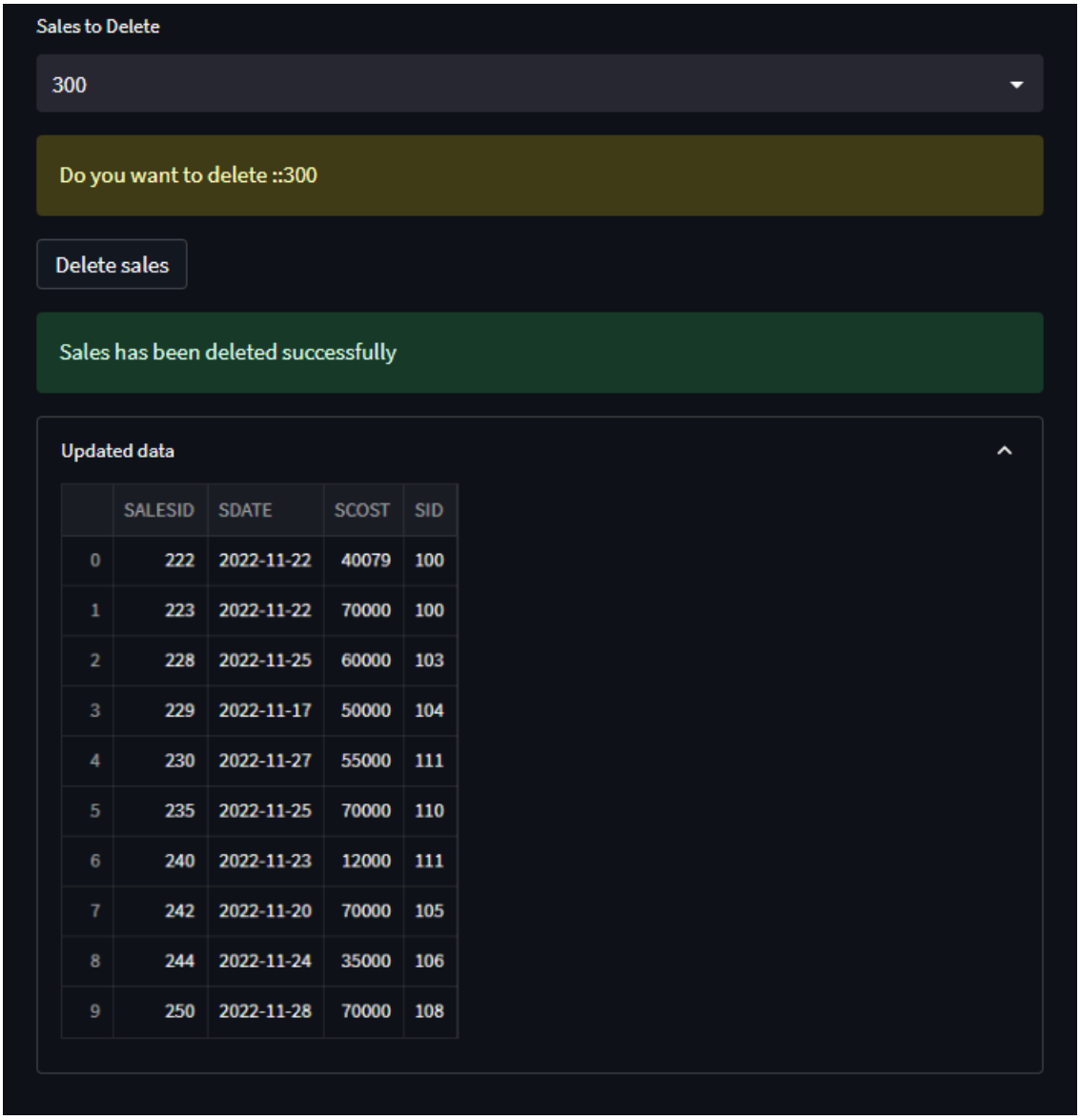


	CRY	SCRY	Quant	SID
0	Electrical	Mobiles	30	101
1	Snacks	Cereal	300	100
2	Electrical	Wires	750	100

Deleting an entry from sales table:



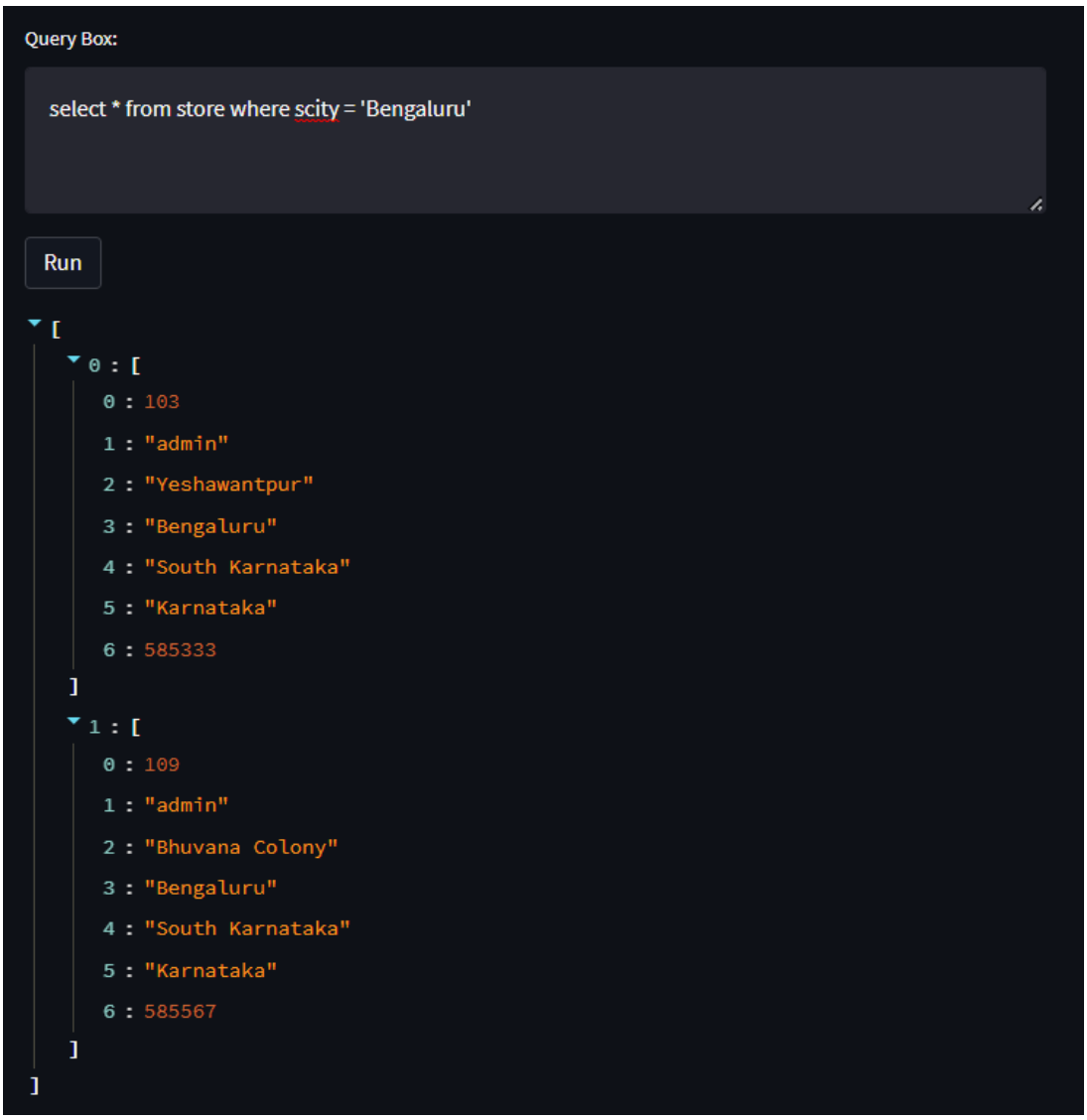
In the same page you have an option to see the changes after deletion:



	SALESID	SDATE	SCOST	SID
0	222	2022-11-22	40079	100
1	223	2022-11-22	70000	100
2	228	2022-11-25	60000	103
3	229	2022-11-17	50000	104
4	230	2022-11-27	55000	111
5	235	2022-11-25	70000	110
6	240	2022-11-23	12000	111
7	242	2022-11-20	70000	105
8	244	2022-11-24	35000	106
9	250	2022-11-28	70000	108

Note: These functionalities are applicable and included in front end for all the tables that have been created in the database.

Query Box:



----------THANK YOU----------