Part1
1
Create a new schema named "SQL_Basics_Practice"
Ans:
```
Create Schema SQL_Basics_Practice;
```

2
Create a table in this new schema named "sales" using SQL "Create" command. It should have all the columns as per "Data_Sales" sheet. Make OrderId as primary key
Ans:
```
use sql_basics_practice;
CREATE TABLE sales (
  `Order ID` INTEGER not null unique,
  `Order Date` DATETIME,
  `Customer ID` INTEGER,
  `Customer Name` VARCHAR(9),
  `Sales Person` VARCHAR(12),
  `Region` VARCHAR(10),
  `Product_Type` VARCHAR(9),
  `Price` INTEGER,
  `Quantity` INTEGER,
  primary key (`Order ID`)
);
```

3
Insert 5 records into this table using INSERT operation

Ans:
```
INSERT INTO sales
  (`Order ID`, `Order Date`, `Customer ID`, `Customer Name`, `Sales Person`,
`Region`, `Product_Type`, `Price`, `Quantity`)
VALUES
  (0009, '2021-05-01', '20', 'Company T', 'Andrew James', 'Arizona', 'Product 1',
399, 5),
  (0012, '2021-05-01', '6', 'Company F', 'Laura Larsen', 'California', 'Product 1',
399, 6),
  (0014, '2021-05-01', '4', 'Company D', 'Anna Weber', 'Texas', 'Product 1', 399,
4),
  (0020, '2021-07-01', '5', 'Company E', 'Anna Weber', 'Texas', 'Product 1', 399,
3),
  (0024, '2021-07-01', '12', 'Company L', 'Michael Fox', 'New Mexico', 'Product 1',
399, 2);
```

4
SELECT all columns of the table sorted by order date first to last and then by quantity highest to lowest
Ans:
```
Select * from sql_basics_practice.sales order by 'order date' asc, quantity desc;
```

5
Create a new table named "sales_bkp" from the existing "sales" table copying all rows & columns
Ans:
Create table sales_bkp (select * from sales);

6
Delete the first row from sales_bkp

Ans:
first not able to delte first row as there is not primary key for sales_bkp table
After adding primary key to order id
Alter table sales_bkp add Primary key(`Order ID`);

Delete from sales_bkp where 'Order ID' = 9;

7
Add a new column named "CreatedBy" that has default value of your name

Ans:
Alter Table sales_bkp add column `Created By` NVARCHAR(20) default 'Pavan';

8
SELECT ORDERID, CREATEDBY, SALESPERSON and PRICE from sales_bkp.

Ans:
select `Order ID`, `Sales Person`, `Created By`, `Price` from sales_bkp;

9
Remove "CreatedBy" column from sales_bkp

Ans:
Alter table sales_bkp drop column `Created By`;

10
Remove sales_bkp table from the schema

Ans:
Truncate table SQL_Basics_Practice.sales_bkp; // Deleteing all rows

Drop table SQL_Basics_Practice.sales_bkp; // Deleting table from schema

Part2
1
Create a new table "sales_data" using data table import functionality. Import the "data_sales" sheet into this table.
Ans:
Table Data Import Wizard

2
Inspect this table "sales_data" especially its columns data types and the SQL that generated this table
Ans:
Describe data_sales;

3
SELECT all columns of this table
Ans:
select * from sql_basics_practice.data_sales;

4
SELECT all columns of this table only for the State of California
Ans:

select * from sql_basics_practice.data_sales where region = 'California';

5
Find the "Minimum", "Maximum", Average and Total values for Price and Quantity
Ans:
select max(quantity),min(quantity),max(price),min(price),avg(quantity),avg(price), sum(Quantity),sum(Price) from sql_basics_practice.data_sales;

6
Count the number of orders occurring for each Product
Ans:
select product_type ,Count(*) from sql_basics_practice.data_sales group by product_type;

7
Create a "Revenue" column which is product of Price and Quantity
Ans:
Alter table sql_basics_practice.data_sales add column Revenue double default (Quantity*price);
select Revenue from sql_basics_practice.data_sales;

8
Calculate the total revenue for each sales person
Ans:
select `Sales Person`,   sum(Revenue) from sql_basics_practice.data_sales group by `Sales Person` order by sum(Revenue) desc ;
9
Get the list of orders for which revenue is between 1000 and 3000, for the state of

TEXAS
Ans:

```
select * from sql_basics_practice.data_sales where region = 'Texas' and revenue
between 1000 and 3000 order by revenue desc ;
```

10
Get the list of Sales Person where total revenue is greater than 250000

Ans:

```
select  `Sales Person`,  sum(Revenue) from  sql_basics_practice.data_sales group by
`Sales Person` having  sum(Revenue)  > 250000    ;
```