Data Base Management System Lab

Shivansh, MCA Section-A

- **Create the tables described below:**

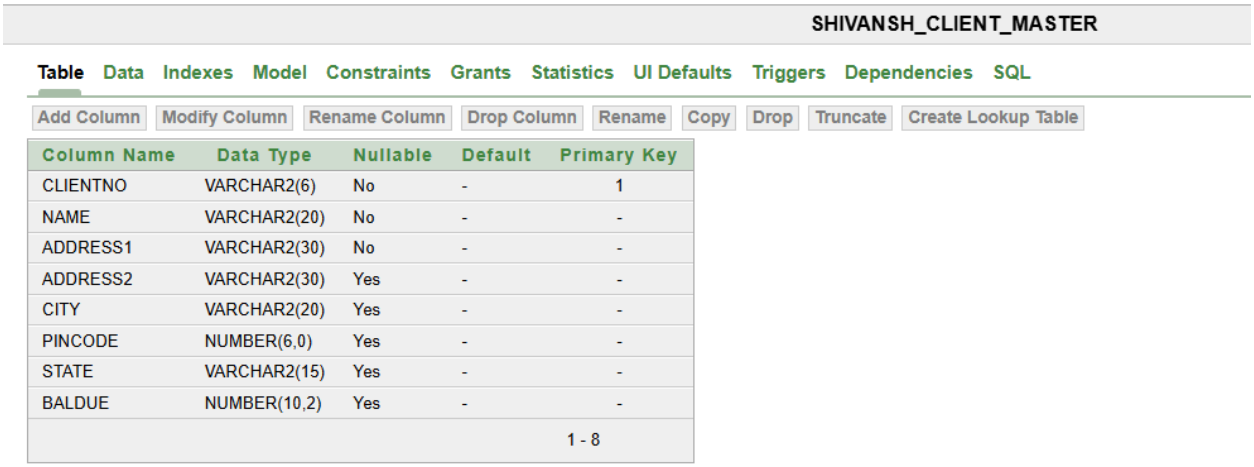
Table Name: SHIVANSH_CLIENT_MASTER

Description: Used to store client information

Command:

```
CREATE TABLE SHIVANSH_CLIENT_MASTER (
CLIENTNO VARCHAR2(6),
NAME VARCHAR2(60),
ADDRESS1 VARCHAR2(30),
ADDRESS2 VARCHAR2(30),
CITY VARCHAR2(15),
PINCODE NUMBER,
STATE VARCHAR2(15),
BALDUE NUMBER (10, 2)
);
```

OUTPUT:



SHIVANSH_CLIENT_MASTER

Table | Data | Indexes | Model | Constraints | Grants | Statistics | UI Defaults | Triggers | Dependencies | SQL

Add Column | Modify Column | Rename Column | Drop Column | Rename | Copy | Drop | Truncate | Create Lookup Table

Column Name	Data Type	Nullable	Default	Primary Key
CLIENTNO	VARCHAR2(6)	No	-	1
NAME	VARCHAR2(20)	No	-	-
ADDRESS1	VARCHAR2(30)	No	-	-
ADDRESS2	VARCHAR2(30)	Yes	-	-
CITY	VARCHAR2(20)	Yes	-	-
PINCODE	NUMBER(6,0)	Yes	-	-
STATE	VARCHAR2(15)	Yes	-	-
BALDUE	NUMBER(10,2)	Yes	-	-

1 - 8

Table Name: SHIVANSH_PRODUCT_MASTER

Description: Used to store client information

Command:

```
CREATE TABLE SHIVANSH_PRODUCT_MASTER (
PRODUCTNO VARCHAR2(6),
DESCRIPTION VARCHAR2(15),
PROFITPERCENT NUMBER (4, 2),
UNITMEASURE VARCHAR2(10),
QTYONHAND NUMBER (8),
REORDERLVL NUMBER (8),
SELLPRICE NUMBER (8, 2),
COSTPRICE NUMBER (8, 2)
);
```

OUTPUT:

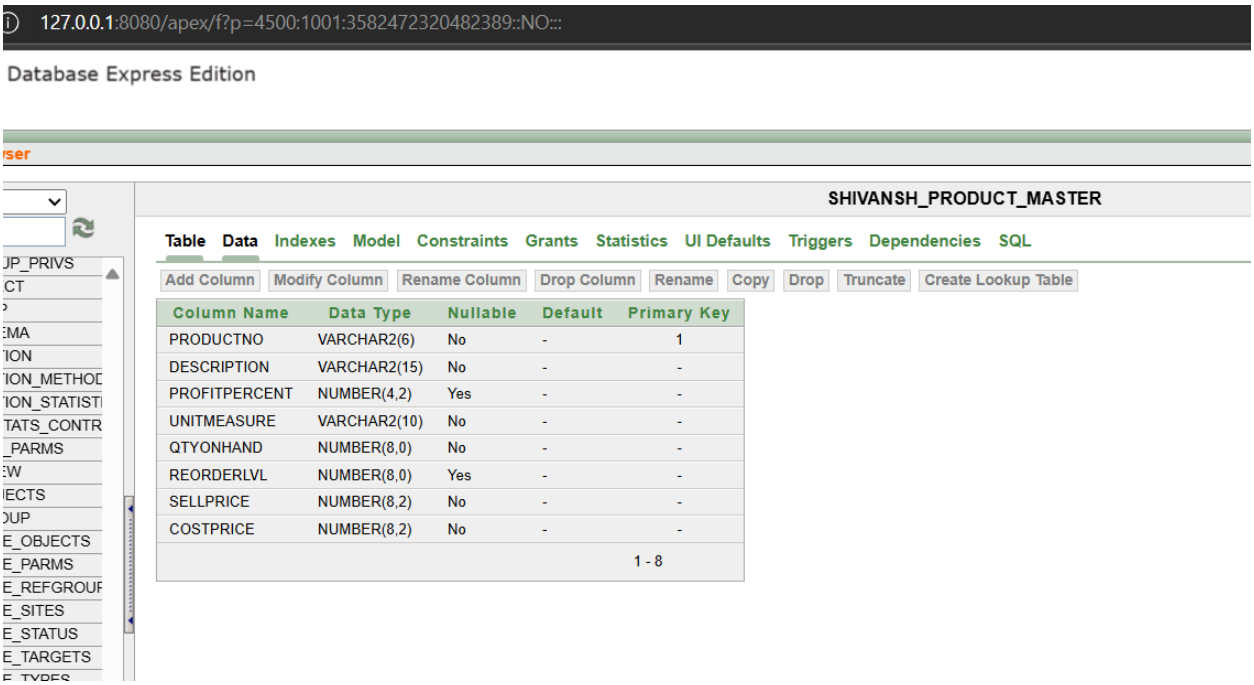


Table Name: SHIVANSH_SALESMAN_MASTER

Description: Used to store client information

Command:

CREATE TABLE SHIVANSH_SALESMAN_MASTER (

SALESMANNO VARCHAR2(6),

SALESMANNAME VARCHAR2(20),

ADDRESS1 VARCHAR2(30),

ADDRESS2 VARCHAR2(30),

CITY VARCHAR2(20),

PINCODE NUMBER (8),

STATE VARCHAR2(20),

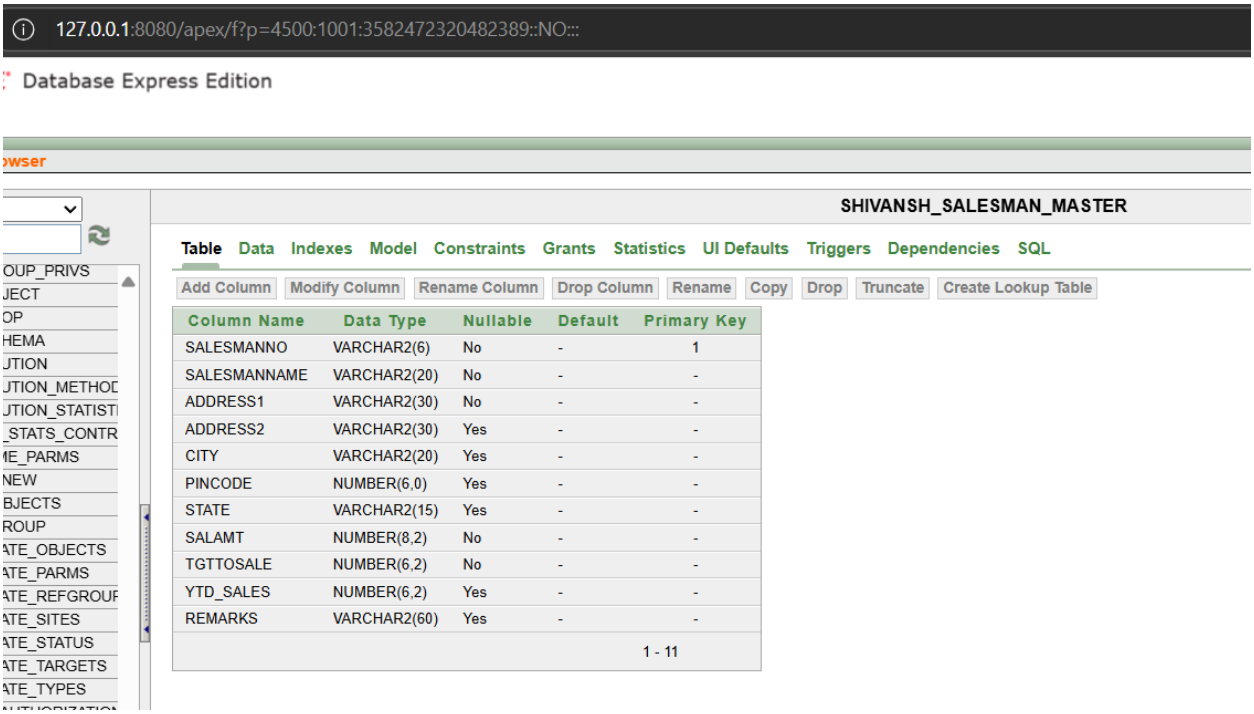
SALAMT NUMBER (8, 2),

TGTTOGET NUMBER (6, 2),

YTD_SALES NUMBER (6, 2),

REMARKS VARCHAR2(60)

); OUTPUT:



1. Insert the following data into their respective table:

a. Data for SHIVANSH_CLIENT_MASTER table:

Command:

INSERT INTO SHIVANSH_CLIENT_MASTER (CLIENTNO, NAME, ADDRESS1, ADDRESS2, CITY, PINCODE, STATE, BALDUE)

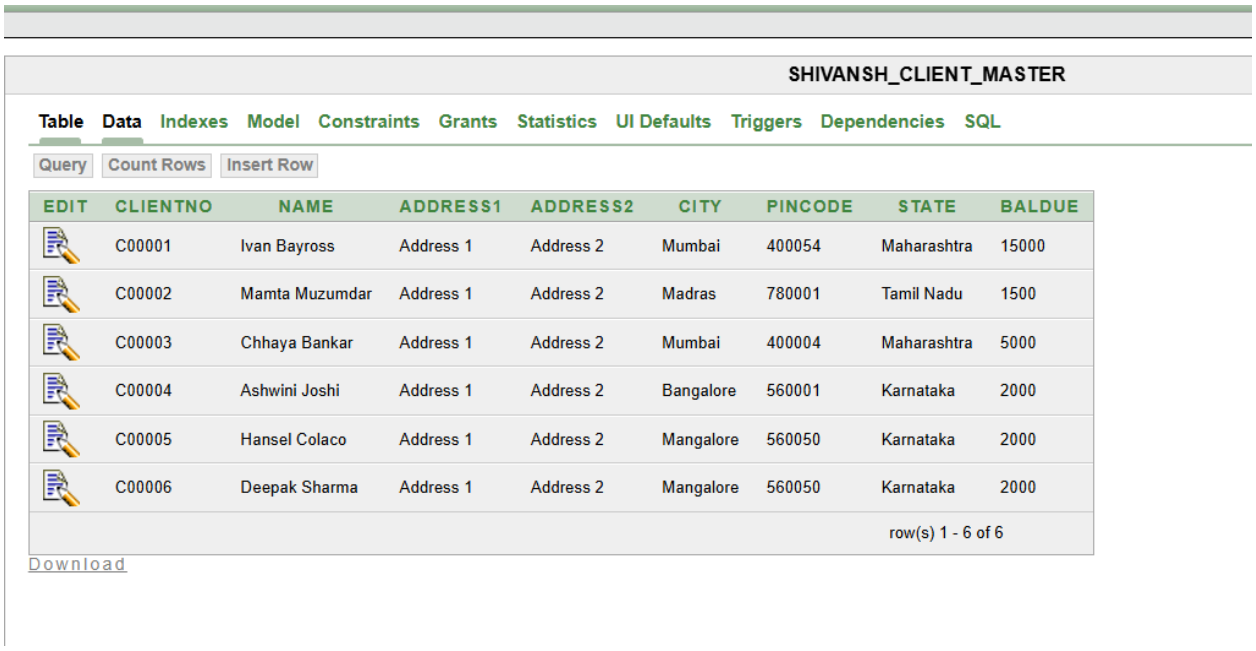
VALUES ('C001', Ivan Bayross, '123 Main St', 'Apt 4B', 'Mumbai', 400054, 'Maharashtra', 15000);

.

.

.

Output:



SHIVANSH_CLIENT_MASTER

Table Data Indexes Model Constraints Grants Statistics UI Defaults Triggers Dependencies SQL

Query Count Rows Insert Row

EDIT	CLIENTNO	NAME	ADDRESS1	ADDRESS2	CITY	PINCODE	STATE	BALDUE
	C00001	Ivan Bayross	Address 1	Address 2	Mumbai	400054	Maharashtra	15000
	C00002	Mamta Muzumdar	Address 1	Address 2	Madras	780001	Tamil Nadu	1500
	C00003	Chhaya Bankar	Address 1	Address 2	Mumbai	400004	Maharashtra	5000
	C00004	Ashwini Joshi	Address 1	Address 2	Bangalore	560001	Karnataka	2000
	C00005	Hansel Colaco	Address 1	Address 2	Mangalore	560050	Karnataka	2000
	C00006	Deepak Sharma	Address 1	Address 2	Mangalore	560050	Karnataka	2000

row(s) 1 - 6 of 6

Download

b. Data for SHIVANSH_PRODUCT_MASTER Table:

Command:

INSERT INTO SHIVANSH_PRODUCT_MASTER (PRODUCTNO, DESCRIPTION, PROFITPERCENT, UNITMEASURE, QTYONHAND, REORDERLVL, SELLPRICE, COSTPRICE)

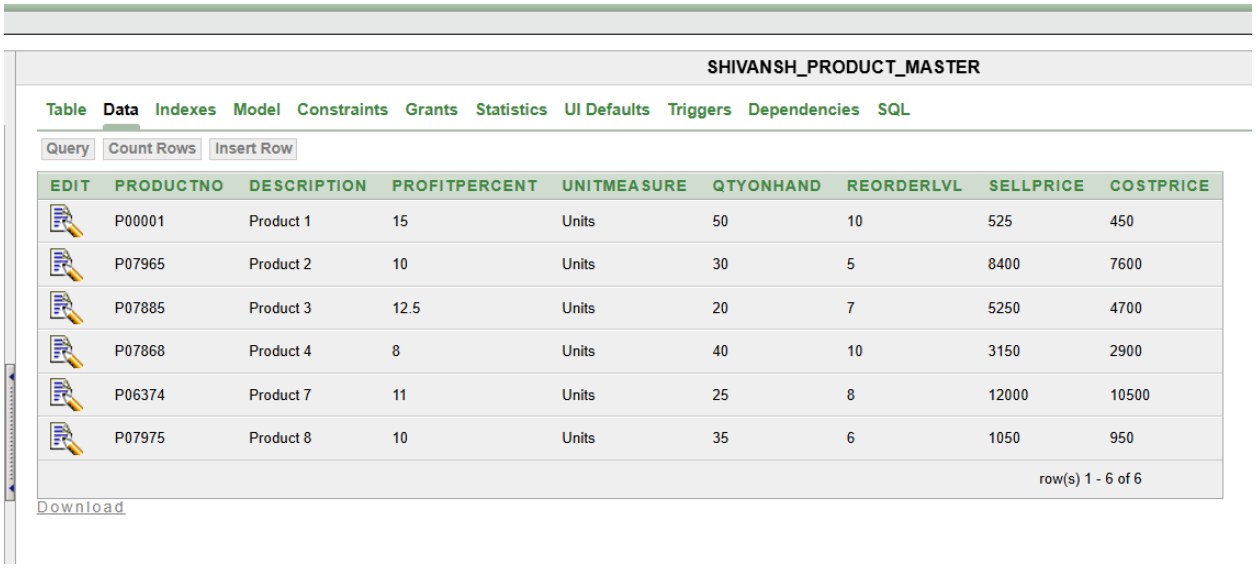
VALUES ('P00001', 'T-Shirts', 5, 'Piece', 200, 50, 350, 250);

.

.

.

Output:



SHIVANSH_PRODUCT_MASTER

Table | Data | Indexes | Model | Constraints | Grants | Statistics | UI Defaults | Triggers | Dependencies | SQL

Query | Count Rows | Insert Row

EDIT	PRODUCTNO	DESCRIPTION	PROFITPERCENT	UNITMEASURE	QTYONHAND	REORDERLVL	SELLPRICE	COSTPRICE
	P00001	Product 1	15	Units	50	10	525	450
	P07965	Product 2	10	Units	30	5	8400	7600
	P07885	Product 3	12.5	Units	20	7	5250	4700
	P07868	Product 4	8	Units	40	10	3150	2900
	P06374	Product 7	11	Units	25	8	12000	10500
	P07975	Product 8	10	Units	35	6	1050	950

row(s) 1 - 6 of 6

Download

c. Data for SHIVANSH_SALESMAN_MASTER Table:

Command:

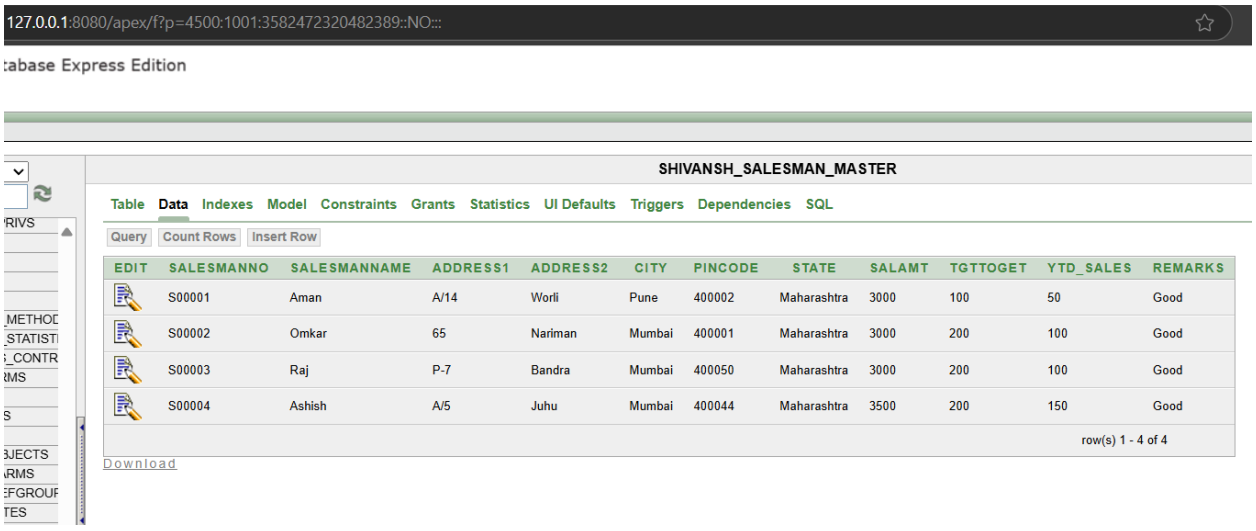
```
INSERT INTO SHIVANSH_SALESMAN_MASTER (SALESMANNO, SALESMANNAME, ADDRESS1,
ADDRESS2, CITY, PINCODE, STATE, SALAMT, TGTTOGET, YTD_SALES, REMARKS)

VALUES ('S00001', 'Aman', 'A/14', 'Worli', 'Mumbai', 400002, 'Maharashtra', 3000, 100, 50, 'Good');
```

.

.

.

Output:



127.0.0.1:8080/apex/f?p=4500:1001:3582472320482389::NO:::

tabase Express Edition

SHIVANSH_SALESMAN_MASTER

Table | Data | Indexes | Model | Constraints | Grants | Statistics | UI Defaults | Triggers | Dependencies | SQL

Query | Count Rows | Insert Row

EDIT	SALESMANNO	SALESMANNAME	ADDRESS1	ADDRESS2	CITY	PINCODE	STATE	SALAMT	TGTTOGET	YTD_SALES	REMARKS
	S00001	Aman	A/14	Worli	Pune	400002	Maharashtra	3000	100	50	Good
	S00002	Omkar	65	Nariman	Mumbai	400001	Maharashtra	3000	200	100	Good
	S00003	Raj	P-7	Bandra	Mumbai	400050	Maharashtra	3000	200	100	Good
	S00004	Ashish	A/5	Juhu	Mumbai	400044	Maharashtra	3500	200	150	Good

row(s) 1 - 4 of 4

Download

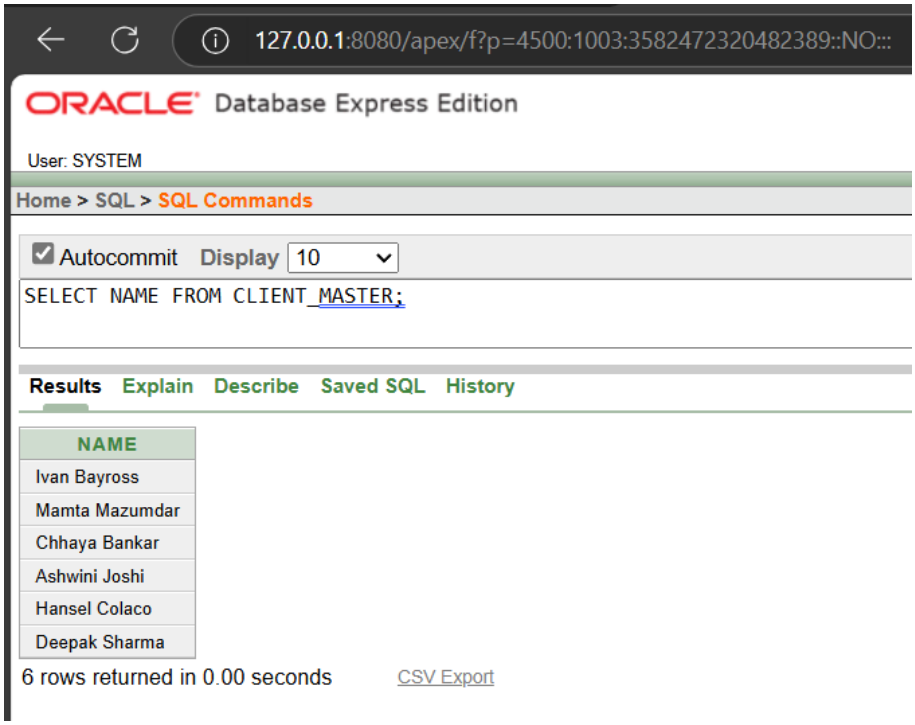
3. Exercise on retrieving records from a table:

1. Find out the names of all the clients:

Command:

```
SELECT NAME FROM SHIVANSH_CLIENT_MASTER;
```

Output:

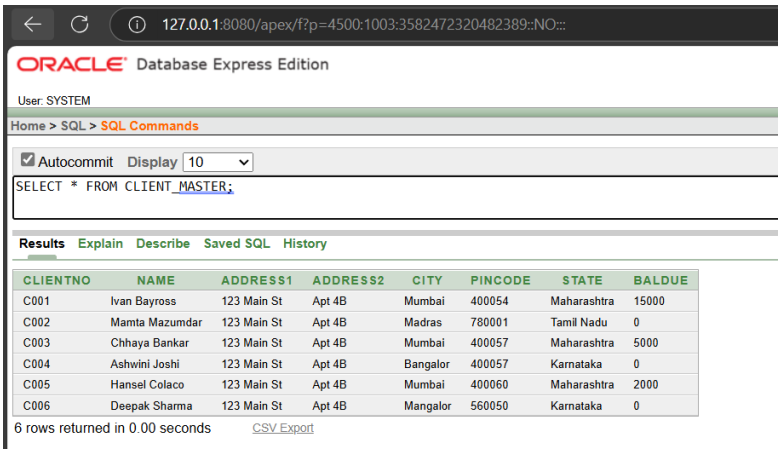


2. Retrieve the entire contents of the SHIVANSH_CLIENT_MASTER table:

Command:

```
SELECT * FROM SHIVANSH_CLIENT_MASTER;
```

Output:

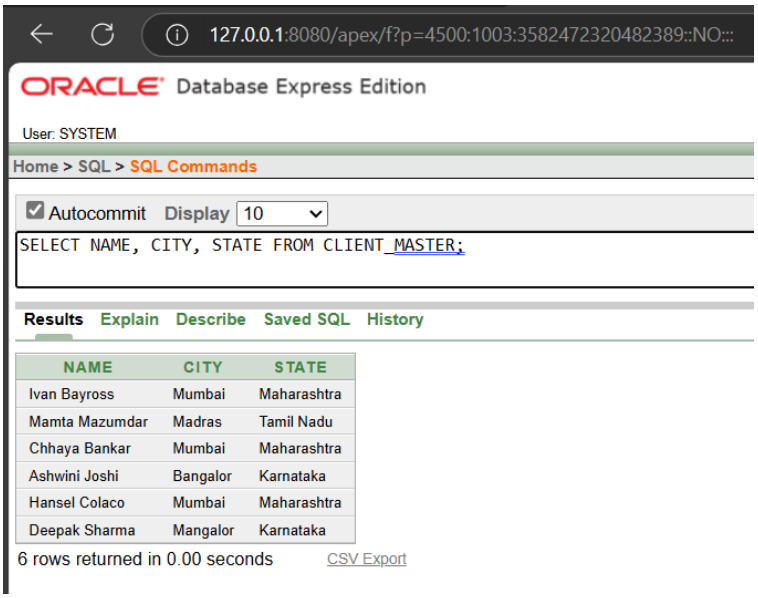


3. Retrieve the list of names, city, and state of all the clients:

Command:

SELECT NAME, CITY, STATE FROM SHIVANSH_CLIENT_MASTER;

Output:

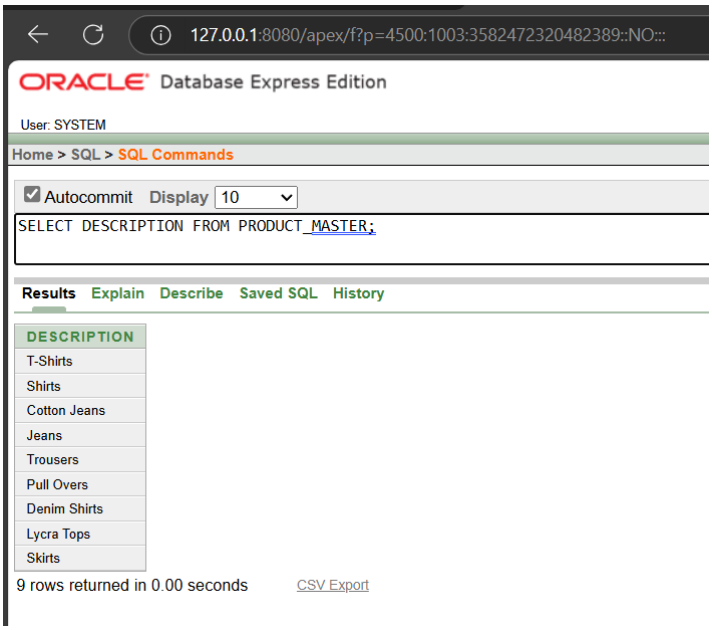


4. List the various products available from the SHIVANSH_PRODUCT_MASTER table:

Command:

SELECT DESCRIPTION FROM SHIVANSH_PRODUCT_MASTER;

Output:

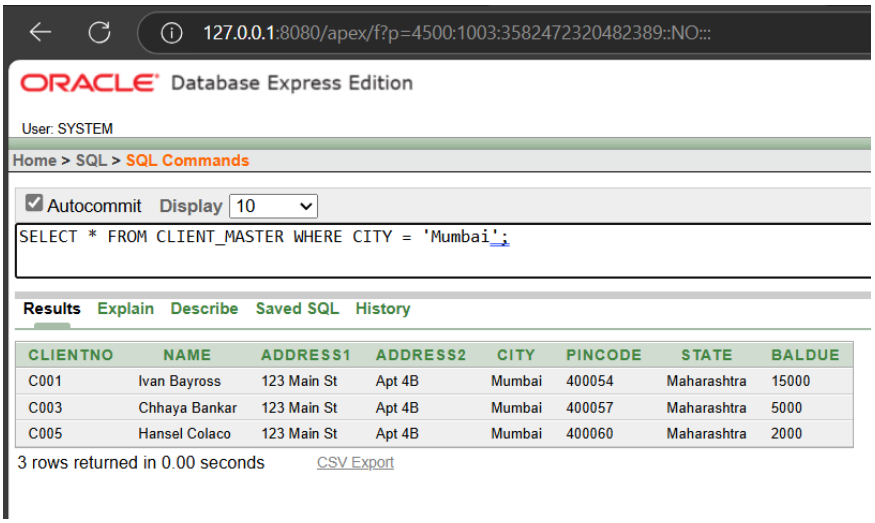


5. List all the clients who are located in Mumbai:

Command:

SELECT * FROM SHIVANSH_CLIENT_MASTER WHERE CITY = 'Mumbai';

Output:

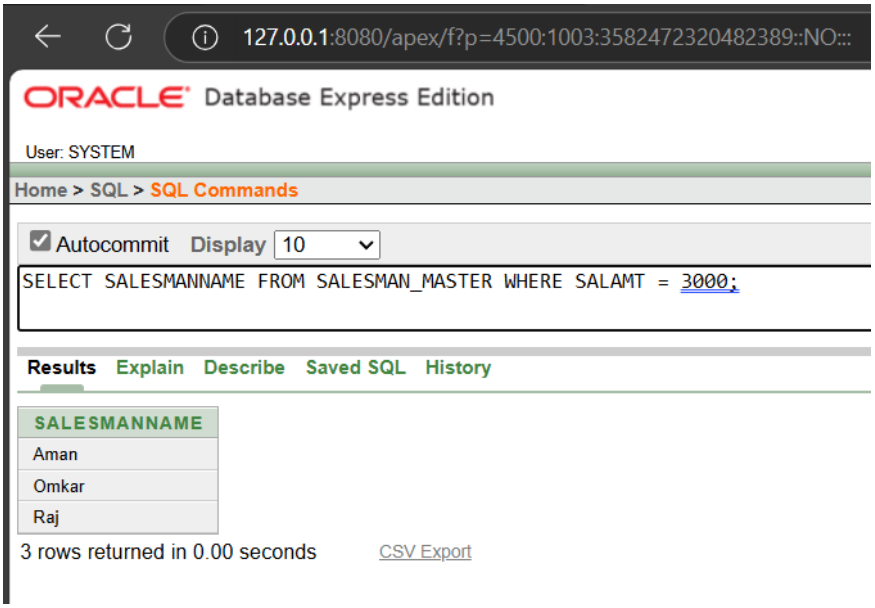


6. Find the names of salesmen who have a salary equal to Rs.3000:

Command:

SELECT SALESMANNAME FROM SHIVANSH_SALESMAN_MASTER WHERE SALAMT = 3000;

Output:



4. Exercise on updating records in a table:

1. Change the city of ClientNo 'C005' to 'Bangalore':

Command:

UPDATE SHIVANSH_CLIENT_MASTER

SET CITY = 'Bangalore'

WHERE CLIENTNO = 'C005';

Output:

Results Explain Describe Saved SQL History

CLIENTNO	NAME	ADDRESS1	ADDRESS2	CITY	PINCODE	STATE	BALDUE
C005	Hansel Colaco	123 Main St	Apt 4B	Bangalore	400060	Maharashtra	2000

1 rows returned in 0.01 seconds CSV Export

2. Change the BalDue of ClientNo 'C001' to Rs. 1000:

Command:

UPDATE SHIVANSH_CLIENT_MASTER

SET BALDUE = 1000

WHERE CLIENTNO = 'C001';

Output:

Results Explain Describe Saved SQL History

CLIENTNO	NAME	ADDRESS1	ADDRESS2	CITY	PINCODE	STATE	BALDUE
C001	Ivan Bayross	123 Main St	Apt 4B	Mumbai	400054	Maharashtra	1000

1 rows returned in 0.00 seconds CSV Export

3. Change the cost price of 'Trousers' to Rs. 950.00:

Command:

UPDATE SHIVANSH_PRODUCT_MASTER

SET COSTPRICE = 950.00

WHERE DESCRIPTION = 'Trousers';

Output:

Results Explain Describe Saved SQL History

PRODUCTNO	DESCRIPTION	PROFITPERCENT	UNITMEASURE	QTYONHAND	REORDERLVL	SELLPRICE	COSTPRICE
P07868	Trousers	2	Piece	150	50	850	950

1 rows returned in 0.00 seconds CSV Export

4. Change the city of the salesman to Pune:

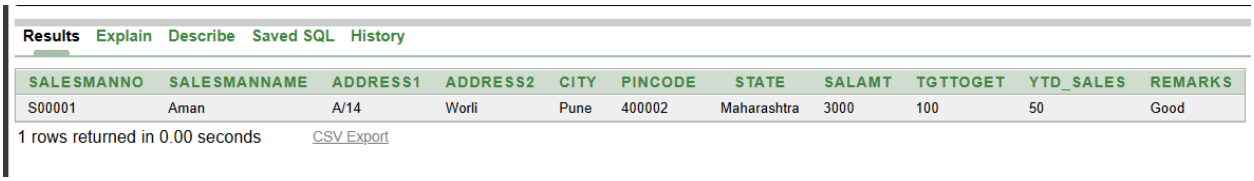
Command:

UPDATE SHIVANSH_SALESMAN_MASTER

SET CITY = 'Pune'

WHERE SALESMANNO = 'S00001';

Output:



Results Explain Describe Saved SQL History

SALESMANNO	SALESMANNAME	ADDRESS1	ADDRESS2	CITY	PINCODE	STATE	SALAMT	TGTTOGET	YTD_SALES	REMARKS
S00001	Aman	A/14	Worli	Pune	400002	Maharashtra	3000	100	50	Good

1 rows returned in 0.00 seconds CSV Export

5. Exercise on deleting records in a table:

a. Delete all salesmen from the SHIVANSH_SALESMAN_MASTER whose salaries are equal to Rs. 3500:

Command :

DELETE FROM SHIVANSH_SALESMAN_MASTER

WHERE salary = 3500;

b. Delete all products from SHIVANSH_PRODUCT_MASTER where the quantity on hand is equal to 100:

Command:

DELETE FROM SHIVANSH_PRODUCT_MASTER WHERE quantity_on_hand = 100;

c. Delete from SHIVANSH_CLIENT_MASTER where the column state holds the value 'Tamil Nadu':

Command:

DELETE FROM SHIVANSH_CLIENT_MASTER

WHERE state = 'Tamil Nadu';

6. Exercise on altering the table structure:

a. Add a column called 'Telephone' of data type 'number' and size ='10' to the SHIVANSH_CLIENT_MASTER table:

Command:

ALTER TABLE SHIVANSH_CLIENT_MASTER

ADD Telephone NUMBER (10);

Output:

Column Name	Data Type	Nullable	Default	Primary Key
CLIENTNO	VARCHAR2(6)	No	-	-
NAME	VARCHAR2(20)	No	-	-
ADDRESS1	VARCHAR2(30)	Yes	-	-
ADDRESS2	VARCHAR2(30)	Yes	-	-
CITY	VARCHAR2(15)	Yes	-	-
PINCODE	VARCHAR2(8)	No	-	-
STATE	VARCHAR2(15)	Yes	-	-
BALDUE	NUMBER(10,2)	Yes	-	-
TELEPHONE	NUMBER(10,0)	Yes	-	-
				1 - 9

b. Change the size of SellPrice column in SHIVANSH_PRODUCT_MASTER to 10,2:

Command:

ALTER TABLE SHIVANSH_PRODUCT_MASTER

MODIFY SellPrice NUMBER (10, 2);

Output:

Column Name	Data Type	Nullable	Default	Primary Key
PRODUCTNO	VARCHAR2(6)	Yes	-	-
DESCRIPTION	VARCHAR2(15)	Yes	-	-
PROFITPERCENT	NUMBER(4,2)	Yes	-	-
UNITMEASURE	VARCHAR2(10)	Yes	-	-
QTYONHAND	NUMBER(8,0)	Yes	-	-
REORDERLVL	NUMBER(8,0)	Yes	-	-
SELLPRICE	NUMBER(10,2)	Yes	-	-
COSTPRICE	NUMBER(8,2)	Yes	-	-
				1 - 8

7. Exercise on deleting the table structure along with the data

a. Destroy the table SHIVANSH_CLIENT_MASTER along with its data:

Command:

DROP TABLE SHIVANSH_CLIENT_MASTER;

8. Exercise on renaming the table

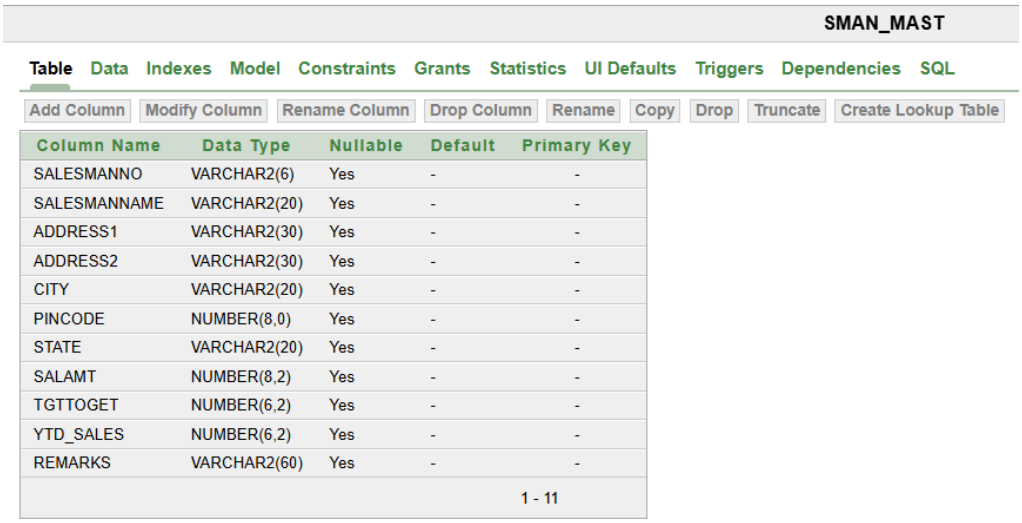
Change the name of the SHIVANSH_SALESMAN_MASTER table to sman_mast:

Command:

ALTER TABLE SHIVANSH_SALESMAN_MASTER

RENAME TO sman_mast;

Output:



SMAN_MAST

Table Data Indexes Model Constraints Grants Statistics UI Defaults Triggers Dependencies SQL

Add Column | Modify Column | Rename Column | Drop Column | Rename | Copy | Drop | Truncate | Create Lookup Table

Column Name	Data Type	Nullable	Default	Primary Key
SALESMANNO	VARCHAR2(6)	Yes	-	-
SALESMANNAME	VARCHAR2(20)	Yes	-	-
ADDRESS1	VARCHAR2(30)	Yes	-	-
ADDRESS2	VARCHAR2(30)	Yes	-	-
CITY	VARCHAR2(20)	Yes	-	-
PINCODE	NUMBER(8,0)	Yes	-	-
STATE	VARCHAR2(20)	Yes	-	-
SALAMT	NUMBER(8,2)	Yes	-	-
TGTTOGET	NUMBER(6,2)	Yes	-	-
YTD_SALES	NUMBER(6,2)	Yes	-	-
REMARKS	VARCHAR2(60)	Yes	-	-

1 - 11

HANDS ON EXERCISES

1. Create the tables described below:

Table Name: SHIVANSH_CLIENT_MASTER

Description: Used to store client's information

Command:

CREATE TABLE SHIVANSH_CLIENT_MASTER (

CLIENTNO VARCHAR2(6) PRIMARY KEY CHECK (CLIENTNO LIKE 'C%'),

NAME VARCHAR2(20) NOT NULL,

ADDRESS1 VARCHAR2(30) NOT NULL,

ADDRESS2 VARCHAR2(30),

CITY VARCHAR2(20),

PINCODE NUMBER (6),

STATE VARCHAR2(15),

BALDUE NUMBER (10, 2)

);

Output:

Column Name	Data Type	Nullable	Default	Primary Key
CLIENTNO	VARCHAR2(6)	No	-	1
NAME	VARCHAR2(20)	No	-	-
ADDRESS1	VARCHAR2(30)	No	-	-
ADDRESS2	VARCHAR2(30)	Yes	-	-
CITY	VARCHAR2(20)	Yes	-	-
PINCODE	NUMBER(6,0)	Yes	-	-
STATE	VARCHAR2(15)	Yes	-	-
BALDUE	NUMBER(10,2)	Yes	-	-
				1 - 8

Table Name: SHIVANSH_PRODUCT_MASTER

Description: Used to store products information

Command:

```
CREATE TABLE SHIVANSH_PRODUCT_MASTER (
PRODUCTNO VARCHAR2(6) PRIMARY KEY CHECK (PRODUCTNO LIKE 'P%'),
DESCRIPTION VARCHAR2(15) NOT NULL,
PROFITPERCENT NUMBER (4, 2),
UNITMEASURE VARCHAR2(10) NOT NULL,
QTYONHAND NUMBER (8) NOT NULL,
REORDERLVL NUMBER (8),
SELLPRICE NUMBER (8, 2) NOT NULL CHECK (SELLPRICE > 0),
COSTPRICE NUMBER (8, 2) NOT NULL
);
```

Output:

Column Name	Data Type	Nullable	Default	Primary Key
PRODUCTNO	VARCHAR2(6)	No	-	1
DESCRIPTION	VARCHAR2(15)	No	-	-
PROFITPERCENT	NUMBER(4,2)	Yes	-	-
UNITMEASURE	VARCHAR2(10)	No	-	-
QTYONHAND	NUMBER(8,0)	No	-	-
REORDERLVL	NUMBER(8,0)	Yes	-	-
SELLPRICE	NUMBER(8,2)	No	-	-
COSTPRICE	NUMBER(8,2)	No	-	-
				1 - 8

Table Name: SHIVANSH_SALESMAN_MASTER

Description: Used to store salesman information working for the company

Command:

```
CREATE TABLE SHIVANSH_SALESMAN_MASTER (
SALESMANNO VARCHAR2(6) PRIMARY KEY CHECK (SALESMANNO LIKE 'S%'),
SALESMANNAME VARCHAR2(20) NOT NULL,
ADDRESS1 VARCHAR2(30) NOT NULL,
ADDRESS2 VARCHAR2(30),
CITY VARCHAR2(20),
PINCODE NUMBER (6),
STATE VARCHAR2(15),
SALAMT NUMBER (8, 2) NOT NULL CHECK (SALAMT > 0),
TGTTOSALE NUMBER (6, 2) NOT NULL CHECK (TGTTOSALE > 0),
YTD_SALES NUMBER (6, 2),
REMARKS VARCHAR2(60)
);
```

Output:

Column Name	Data Type	Nullable	Default	Primary Key
SALESMANNO	VARCHAR2(6)	No	-	1
SALESMANNAME	VARCHAR2(20)	No	-	-
ADDRESS1	VARCHAR2(30)	No	-	-
ADDRESS2	VARCHAR2(30)	Yes	-	-
CITY	VARCHAR2(20)	Yes	-	-
PINCODE	NUMBER(6,0)	Yes	-	-
STATE	VARCHAR2(15)	Yes	-	-
SALAMT	NUMBER(8,2)	No	-	-
TGTTOSALE	NUMBER(6,2)	No	-	-
YTD_SALES	NUMBER(6,2)	Yes	-	-
REMARKS	VARCHAR2(60)	Yes	-	-
				1 - 11

Table Name: SALES _ORDER

Description: Used to store client's order

Command:

```
CREATE TABLE SALES_ORDER (
ORDERNO VARCHAR2(6) PRIMARY KEY CHECK (ORDERNO LIKE 'O%'),
CLIENTNO VARCHAR2(6) REFERENCES SHIVANSH_CLIENT_MASTER(CLIENTNO),
ORDERDATE DATE NOT NULL,
DELYDATE DATE,
SALESMANNO VARCHAR2(6) REFERENCES SHIVANSH_SALESMAN_MASTER(SALESMANNO),
ORDAMT NUMBER (8, 2) NOT NULL,
BILLYN CHAR (1) DEFAULT 'F' CHECK (BILLYN IN ('P', 'F')),
DELYTYPE CHAR (1) DEFAULT 'P' CHECK (DELYTYPE IN ('P', 'F')),
ORDERSTATUS VARCHAR2(10) CHECK (ORDERSTATUS IN ('In Process', 'Fulfilled', 'BackOrder', 'Cancelled'))
);
```

Output:

Column Name	Data Type	Nullable	Default	Primary Key
ORDERNO	VARCHAR2(6)	No	-	1
CLIENTNO	VARCHAR2(6)	Yes	-	-
ORDERDATE	DATE	No	-	-
DELYDATE	DATE	Yes	-	-
SALESMANNO	VARCHAR2(6)	Yes	-	-
ORDAMT	NUMBER(8,2)	No	-	-
BILLYN	CHAR(1)	Yes	'F'	-
DELYTYPE	CHAR(1)	Yes	'P'	-
ORDERSTATUS	VARCHAR2(10)	Yes	-	-
				1 - 9

Table Name: SALES _ORDER_DETAILS

Description: Used to store client's order with details of each product ordered

Command:

```
CREATE TABLE SALES_ORDER_DETAILS (
OrderNo     VARCHAR2(12), -- Foreign Key references OrderNo of Sales_Order table
ProductNo   VARCHAR2(10), -- Foreign Key references ProductNo of SHIVANSH_PRODUCT_MASTER table
QtyOrdered NUMBER (8),
QtyDisp     NUMBER (8),
ProductRate NUMBER (10,2),
CONSTRAINT fk_sales_order FOREIGN KEY (OrderNo) REFERENCES SALES_ORDER(OrderNo),
CONSTRAINT fk_product FOREIGN KEY (ProductNo) REFERENCES
SHIVANSH_PRODUCT_MASTER(ProductNo)
);
```

Output:

Column Name	Data Type	Nullable	Default	Primary Key
ORDERNO	VARCHAR2(12)	Yes	-	-
PRODUCTNO	VARCHAR2(10)	Yes	-	-
QTYORDERED	NUMBER(8,0)	Yes	-	-
QTYDISP	NUMBER(8,0)	Yes	-	-
PRODUCTRATE	NUMBER(10,2)	Yes	-	-
				1 - 5

HANDS ON EXERCISES:

1.Perform the following computations on the table data:

a. List the names of all clients having 'a' as the second letter in their names.

Command:

```
SELECT Name
FROM SHIVANSH_CLIENT_MASTER
WHERE Name LIKE '_a%';
```

Output:

NAME
Mamta Muzumdar
Hansel Colaco

2 rows returned in 0.00 seconds

b. List the clients who stay in a city whose first letter is 'M'.

Command:

```
SELECT Name
FROM SHIVANSH_CLIENT_MASTER
WHERE City LIKE 'M%';
```

Output:

Results Explain Describe Saved SQL History

NAME
Ivan Bayross
Mamta Muzumdar
Chhaya Bankar
Hansel Colaco
Deepak Sharma

5 rows returned in 0.00 seconds CSV Export

c. List all clients who stay in 'Bangalore' or 'Mangalore'.

Command:

```
SELECT Name
FROM SHIVANSH_CLIENT_MASTER
WHERE City IN ('Bangalore', 'Mangalore');
```

Output:

Results Explain Describe Save

NAME
Ashwini Joshi
Hansel Colaco
Deepak Sharma

3 rows returned in 0.00 seconds

d. List all clients whose BalDue is greater than value 10000.

Command:

```
SELECT Name
FROM SHIVANSH_CLIENT_MASTER
WHERE BalDue > 10000;
```

Output:

Results Explain Describe Saved

NAME
Ivan Bayross

1 rows returned in 0.00 seconds

e. List all information from the Sales_Order table for orders placed in the month of June.

Command:

```
SELECT *
FROM SHIVANSH_SALE_ORDER
WHERE EXTRACT (MONTH FROM ORDERDATE) = 6;
```

Output:

Results Explain Describe Saved SQL History

ORDERNO	CLIENTNO	ORDERDATE	DELYDATE	SALESMANNO	ORDAMT	BILLYN	DELYTYPE	ORDERSTATUS
O19001	C00001	12-JUN-04	20-JUL-02	S00001	2000	F	P	In Process
O19002	C00002	12-JUN-04	20-JUL-02	S00001	2000	F	P	Cancelled

2 rows returned in 0.00 seconds CSV Export

f. List all order information for ClientNo 'C00001' and 'C00002'.

Command:

```
SELECT *

FROM SHIVANSH_SALE_ORDER

WHERE ClientNo IN ('C00001', 'C00002');
```

Output:

Results Explain Describe Saved SQL History

ORDERNO	CLIENTNO	ORDERDATE	DELYDATE	SALESMANNO	ORDAMT	BILLYN	DELYTYPE	ORDERSTATUS
O19001	C00001	12-JUN-04	20-JUL-02	S00001	2000	F	P	In Process
O19002	C00002	12-JUN-04	20-JUL-02	S00001	2000	F	P	Cancelled
O19003	C00001	03-APR-04	07-APR-02	S00003	2000	F	P	Fulfilled

3 rows returned in 0.00 seconds CSV Export

g. List products whose selling price is greater than 500 and less than or equal to 750.

Command:

```
SELECT *

FROM SHIVANSH_PRODUCT_MASTER

WHERE SELLPRICE> 500 AND SELLPRICE<= 750;
```

Output:

Results Explain Describe Saved SQL History

PRODUCTNO	DESCRIPTION	PROFITPERCENT	UNITMEASURE	QTYONHAND	REORDERLVL	SELLPRICE	COSTPRICE
P00001	Product 1	15	Units	50	10	525	450

1 rows returned in 0.00 seconds CSV Export

h. List products whose selling price is more than 500. Calculate a new selling price as original selling price * 1.15. Rename the new column in the output as new_price.

Command:

```
SELECT PRODUCTNO, DESCRIPTION, SELLPRICE, SELLPRICE * 1.15 AS new_price

FROM SHIVANSH_PRODUCT_MASTER

WHERE SELLPRICE > 500;
```

Output:

Results Explain Describe Saved SQL History

PRODUCTNO	DESCRIPTION	SELLPRICE	NEW_PRICE
P00001	Product 1	525	603.75
P07965	Product 2	8400	9660
P07885	Product 3	5250	6037.5
P07868	Product 4	3150	3622.5
P06374	Product 7	12000	13800
P07975	Product 8	1050	1207.5

6 rows returned in 0.00 seconds CSV Export

i. List the names, city, and state of clients who are not in the state of 'Maharashtra'.

Command:

```
SELECT NAME, City, State
FROM SHIVANSH_CLIENT_MASTER
WHERE State <> 'Maharashtra';
```

Output:

Results Explain Describe Saved SQL History

NAME	CITY	STATE
Mamta Muzumdar	Madras	Tamil Nadu
Ashwini Joshi	Bangalore	Karnataka
Hansel Colaco	Mangalore	Karnataka
Deepak Sharma	Mangalore	Karnataka

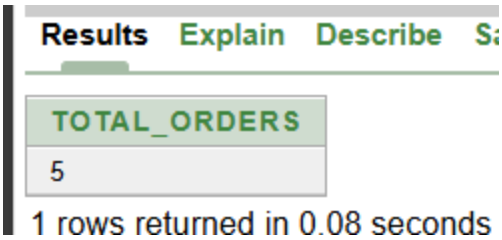
4 rows returned in 0.00 seconds CSV Export

j. Count the total number of orders.

Command:

```
SELECT COUNT (*) AS total_orders
FROM SHIVANSH_SALE_ORDER;
```

Output:



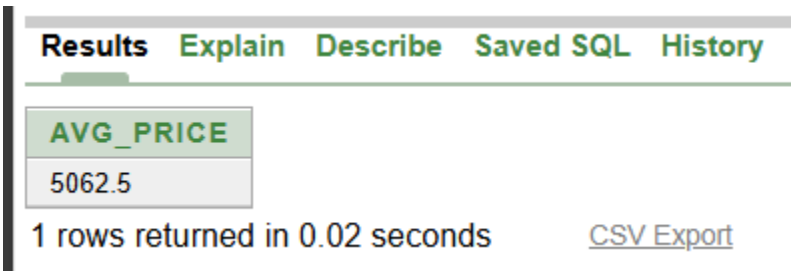
k. Calculate the average price of all products.

Command:

SELECT AVG(SELLPRICE) AS avg_price

FROM SHIVANSH_PRODUCT_MASTER;

Output:



l. Determine the maximum and minimum product prices. Rename the output as max_price and min_price respectively.

Command:

SELECT MAX(SELLPRICE) AS max_price, MIN(SELLPRICE) AS min_price

FROM SHIVANSH_PRODUCT_MASTER;

Output:

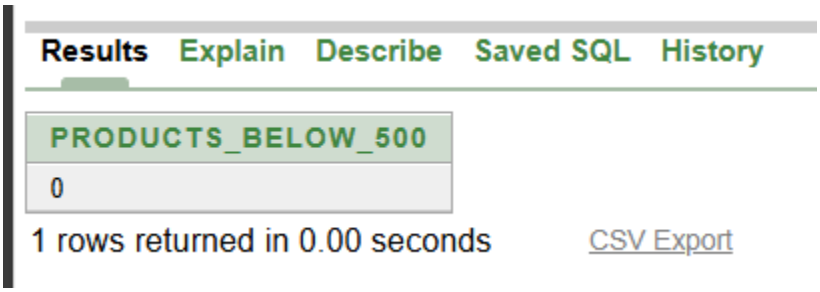
m. Count the number of products having a price less than or equal to 500.
Command:

SELECT COUNT (*) AS products_below_500

FROM SHIVANSH_PRODUCT_MASTER

WHERE SELLPRICE<= 500;

Output:



n. List all products whose QtyOnHand is less than the reorder level.

Command:

SELECT *

FROM SHIVANSH_PRODUCT_MASTER

WHERE QtyOnHand < REORDERLVL;

Output:

No data found.

2. Exercise on Date Manipulation:

a. List the order number and day on which clients placed their order.

Command:

SELECT ORDERNO, TO_CHAR (ORDERDATE, 'DD') AS Order_Day

FROM SHIVANSH_SALE_ORDER;

Output:

ORDERNO	ORDER_DAY
O19001	12
O19002	12
O46865	18
O19003	03
O46866	20

5 rows returned in 0.02 seconds

b. List the month (in alphabets) and date when the orders must be delivered.

Command:

SELECT ORDERNO, TO_CHAR (DELYDATE, 'Month') AS Delivery_Month, TO_CHAR (DELYDATE, 'DD') AS Delivery_Day

FROM SHIVANSH_SALE_ORDER;

Output:

Results Explain Describe Saved SQL History

ORDERNO	DELIVERY_MONTH	DELIVERY_DAY
O19001	July	20
O19002	July	20
O46865	February	20
O19003	April	07
O46866	April	07

5 rows returned in 0.00 seconds CSV Export

c. List the OrderDate in the format 'DD-Month-YY', e.g., 12-February-02.

Command:

SELECT TO_CHAR (ORDERDATE, 'DD-Month-YY') AS Formatted_Order_Date

FROM SHIVANSH_SALE_ORDER;

Output:

Results Explain Describe Saved SQL History

FORMATTED_ORDER_DATE
12-June -04
12-June -04
18-February -04
03-April -04
20-May -04

5 rows returned in 0.00 seconds CSV Export

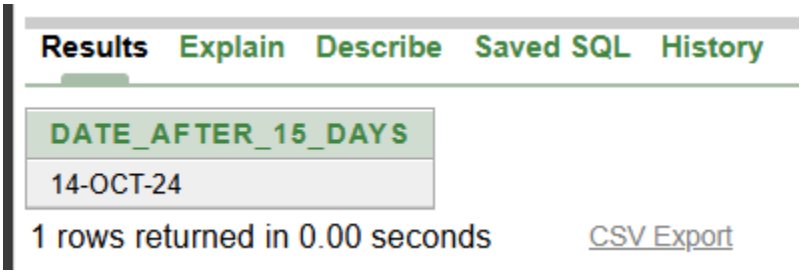
d. List the date 15 days after today's date.

Command:

SELECT SYSDATE + 15 AS Date_After_15_Days

FROM DUAL;

Output:



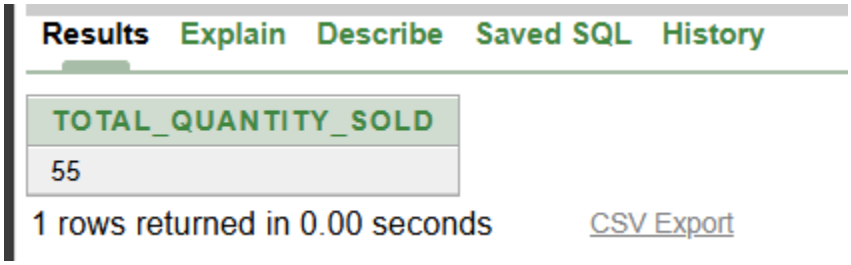
3. Exercise on Having and Group by Clauses:

a. Print the description and total quantity sold for each product.

Command:

```
SELECT SUM(QTYORDERED) AS Total_Quantity_Sold
FROM SHIVANSH_SHIVANSH_SALE_ORDER_DETAILS;
```

Output:



b. Find the value of each product sold.

Command:

```
SELECT SOD.PRODUCTNO, SUM (SOD.QTYORDERED * PM. SELLPRICE) AS Total_Value_Sold
FROM SHIVANSH_SHIVANSH_SALE_ORDER_DETAILS SOD
JOIN SHIVANSH_PRODUCT_MASTER PM ON SOD.PRODUCTNO = PM. PRODUCTNO
GROUP BY SOD.PRODUCTNO;
```

Output:

Results Explain Describe Saved SQL History

PRODUCTNO	TOTAL_VALUE_SOLD
P07965	25200
P07885	26250
P07975	6300
P00001	17850

4 rows returned in 0.00 seconds CSV Export

c. Find out the total of all the billed orders for the month of June.

Command:

```
SELECT SUM(ORDAMT) AS Total_Billed_June
FROM SHIVANSH_SALE_ORDER
WHERE EXTRACT (MONTH FROM ORDERDATE) = 6;
```

Output:

Results Explain Describe Saved SQL History

TOTAL_BILLED_JUNE
4000

1 rows returned in 0.00 seconds CSV Export

4. Exercise on Joins and Correlation:

a. Find the products and their quantities that will have to be delivered in the current month.

Command:

```
SELECT SOD.PRODUCTNO, SOD.QTYORDERED
        FROM SHIVANSH_SALE_ORDER SO
        JOIN SHIVANSH_SHIVANSH_SALE_ORDER_DETAILS SOD ON SO. ORDERNO = SOD.ORDERNO
        WHERE EXTRACT (MONTH FROM SO. DELYDATE) = EXTRACT (MONTH FROM SYSDATE)
        AND EXTRACT (YEAR FROM SO. DELYDATE) = EXTRACT (YEAR FROM SYSDATE);
```

Output:

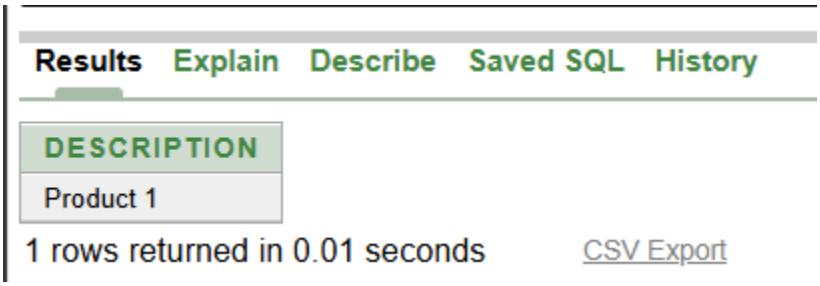
No data found.

b. Find the products and description of constantly sold (rapidly moving) products.

Command:

```
SELECT DESCRIPTION

FROM SHIVANSH_PRODUCT_MASTER

WHERE PRODUCTNO IN (SELECT PRODUCTNO FROM SHIVANSH_SHIVANSH_SALE_ORDER_DETAILS
GROUP BY PRODUCTNO HAVING COUNT (*) > 2);
```

Output:

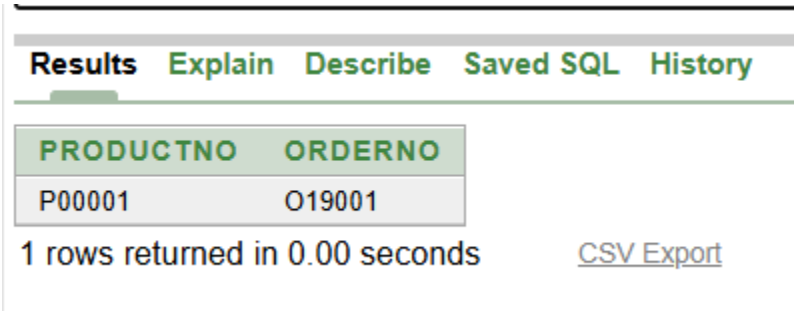


C. Find the names of clients who have purchased 'Trousers'.

Command:

```
SELECT PRODUCTNO, ORDERNO

FROM SHIVANSH_SHIVANSH_SALE_ORDER_DETAILS

WHERE PRODUCTNO = (SELECT PRODUCTNO FROM SHIVANSH_PRODUCT_MASTER WHERE
DESCRIPTION = 'Product 1')

AND QTYORDERED < 5;
```

Output:



d. Find the products and their quantities for the orders placed by 'Ivan Bayross' and 'Mamta Muzumdar'.

Command:

```
SELECT SOD.PRODUCTNO, SOD.QTYORDERED, SO. ORDERNO, CM.NAME
```

FROM SHIVANSH_SHIVANSH_SALE_ORDER_DETAILS SOD

JOIN SHIVANSH_SALE_ORDER SO ON SOD.ORDERNO = SO. ORDERNO

JOIN SHIVANSH_CLIENT_MASTER CM ON SO. CLIENTNO = CM. CLIENTNO

WHERE CM.NAME IN ('Ivan Bayross', 'Mamta Muzumdar');

Output:

Results Explain Describe Saved SQL History

PRODUCTNO	QTYORDERED	ORDERNO	NAME
P07885	2	O19001	Ivan Bayross
P07965	2	O19001	Ivan Bayross
P00001	4	O19001	Ivan Bayross
P00001	10	O19002	Mamta Muzumdar
P06734	1	O19003	Ivan Bayross
P03453	2	O19003	Ivan Bayross

6 rows returned in 0.02 seconds CSV Export

e. Find the products and their quantities for the orders placed by ClientNo 'C00001' and 'C00002'.

Command:

SELECT SOD.PRODUCTNO, SOD.QTYORDERED, SO. ORDERNO, SO. CLIENTNO

FROM SHIVANSH_SHIVANSH_SALE_ORDER_DETAILS SOD

JOIN SHIVANSH_SALE_ORDER SO ON SOD.ORDERNO = SO. ORDERNO

WHERE SO. CLIENTNO IN ('C00001', 'C00002');

Output:

Results Explain Describe Saved SQL History

PRODUCTNO	QTYORDERED	ORDERNO	CLIENTNO
P00001	4	O19001	C00001
P07965	2	O19001	C00001
P07885	2	O19001	C00001
P00001	10	O19002	C00002
P03453	2	O19003	C00001
P06734	1	O19003	C00001

6 rows returned in 0.00 seconds CSV Export

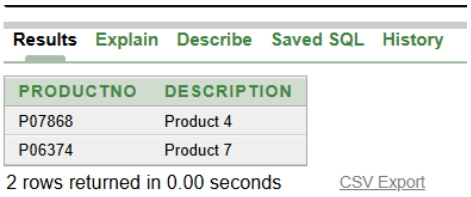
5. **Exercise on Sub-queries:**

a. Find the ProductNo and description of non-moving products i.e. products not being sold.

Command:

```
SELECT PM. PRODUCTNO, PM. DESCRIPTION
FROM SHIVANSH_PRODUCT_MASTER PM
LEFT JOIN SHIVANSH_SHIVANSH_SALE_ORDER_DETAILS SOD ON PM. PRODUCTNO = SOD.PRODUCTNO
WHERE SOD.PRODUCTNO IS NULL;
```

Output:



Results Explain Describe Saved SQL History

PRODUCTNO	DESCRIPTION
P07868	Product 4
P06374	Product 7

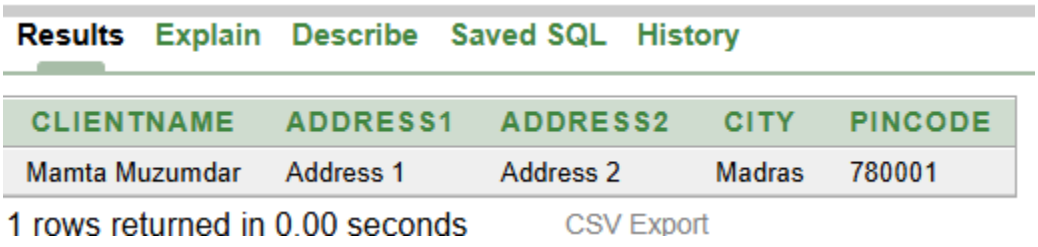
2 rows returned in 0.00 seconds CSV Export

b. Find the Client Name, Address1, Address2, City, and Pin Code for the client who has placed order number 'O19002'.

Command:

```
SELECT CM.NAME AS ClientName, CM. ADDRESS1, CM. ADDRESS2, CM. CITY, CM. PINCODE
FROM SHIVANSH_CLIENT_MASTER CM
JOIN SHIVANSH_SALE_ORDER SO ON CM. CLIENTNO = SO. CLIENTNO
WHERE SO. ORDERNO = 'O19002';
```

Output:



Results Explain Describe Saved SQL History

CLIENTNAME	ADDRESS1	ADDRESS2	CITY	PINCODE
Mamta Muzumdar	Address 1	Address 2	Madras	780001

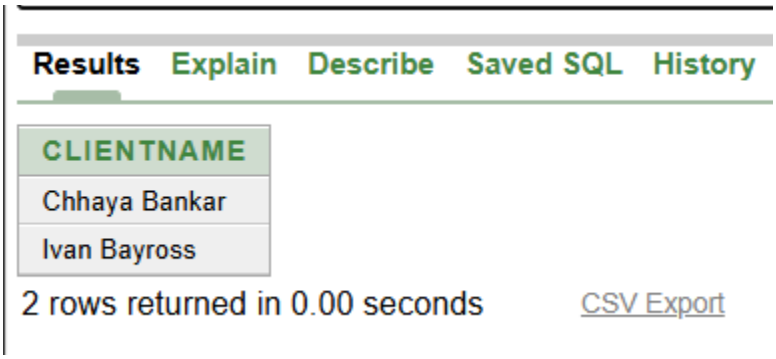
1 rows returned in 0.00 seconds CSV Export

c. List the client names that have placed orders before May '02.

Command:

```
SELECT DISTINCT CM.NAME AS ClientName
FROM SHIVANSH_CLIENT_MASTER CM
JOIN SHIVANSH_SALE_ORDER SO ON CM. CLIENTNO = SO. CLIENTNO
WHERE SO. ORDERDATE < TO_DATE ('01-MAY-2004', 'DD-MON-YYYY');
```

Output:

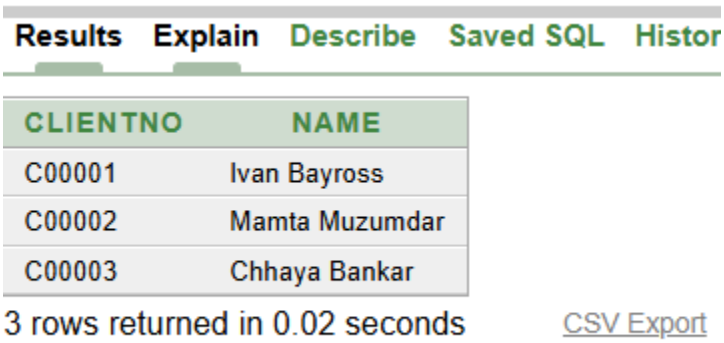


d. List if the product 'Lyra Top' has been ordered by any client and print the ClientNo and Name to whom it was sold.

Command:

```
SELECT DISTINCT CM. CLIENTNO, CM.NAME
FROM SHIVANSH_PRODUCT_MASTER PM
JOIN SHIVANSH_SHIVANSH_SALE_ORDER_DETAILS SOD ON PM. PRODUCTNO = SOD.PRODUCTNO
JOIN SHIVANSH_SALE_ORDER SO ON SOD.ORDERNO = SO. ORDERNO
JOIN SHIVANSH_CLIENT_MASTER CM ON SO. CLIENTNO = CM. CLIENTNO
WHERE PM. DESCRIPTION = 'Product 1';
```

Output:



6. Create the following tables with properly specifying primary keys, Foreign keys and solve following queries:

BRANCH (Branchid,Branchname,Hod)

STUDENT (USN,Name,Address,Branchid,sem)

Author (Authorid,Authorname,Country,Age) Book(Bookid,Book_name,Authorid,Publisher,Branchid)

BORROW (USN,Boookid, Borrowed_Date)

Execute the following queries:

i) **List the details of student who are all studying in 2ns MCA.**

ii) **List the students who are not borrowed any books**

iii) **Display the USN, Student name, Branch_name, book_name, author_name, books_ borrowed _date of 2nd sem MCA students who borrowed books.**

iv) **Display the number of books written by each Author.**

v) **Display the student details who borrowed more than two books.**

vi) **Display the student details who borrowed books of more than one Author.**

vii) **Display the Book names in descending order of their names.**

viii) **List the details of students who borrowed the books which are all published by the same Publisher.**

Commands :

Creating Required Tables :

```
-- Creating BRANCH table
CREATE TABLE Shivansh_BRANCH (
   Branchid INT PRIMARY KEY,
   Branchname VARCHAR(100),
   Hod VARCHAR(100)
);
```

SHIVANSH_BRANCH

Table | Data | Indexes | Model | Constraints | Grants | Statistics | UI Defaults | Triggers | Dependencies | SQL

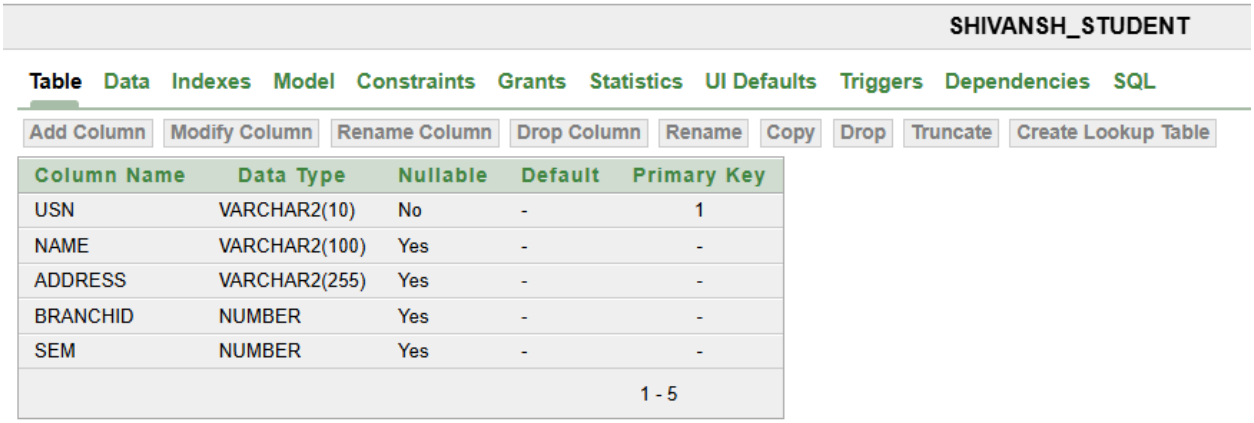
Add Column | Modify Column | Rename Column | Drop Column | Rename | Copy | Drop | Truncate | Create Lookup Table

Column Name	Data Type	Nullable	Default	Primary Key
BRANCHID	NUMBER	No	-	1
BRANCHNAME	VARCHAR2(100)	Yes	-	-
HOD	VARCHAR2(100)	Yes	-	-

1 - 3

```
-- Creating STUDENT table
CREATE TABLE Shivansh_STUDENT (
   USN VARCHAR(10) PRIMARY KEY,
   Name VARCHAR(100),
   Address VARCHAR(255),
   Branchid INT,
```

```
    sem INT,
    FOREIGN KEY (Branchid) REFERENCES Shivansh_BRANCH(Branchid)
);
```



SHIVANSH_STUDENT

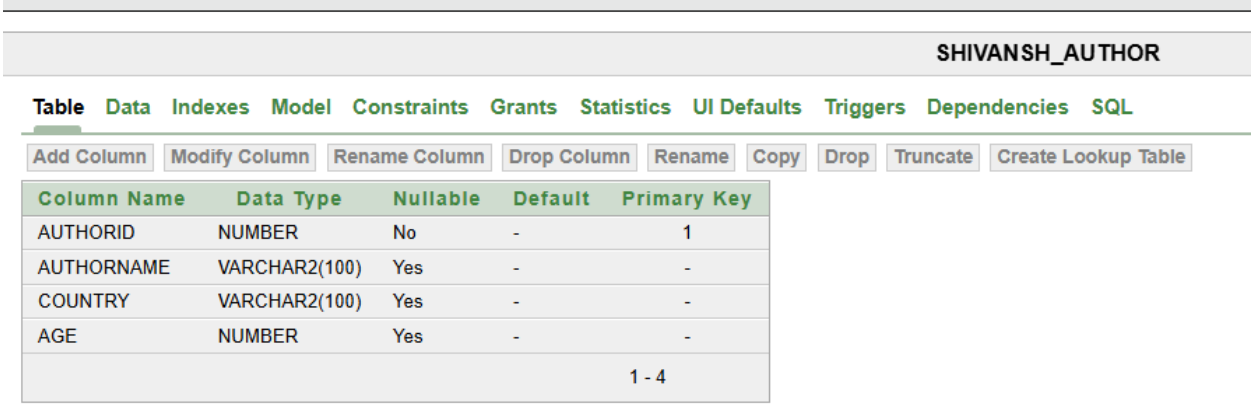
Table | Data | Indexes | Model | Constraints | Grants | Statistics | UI Defaults | Triggers | Dependencies | SQL

Add Column | Modify Column | Rename Column | Drop Column | Rename | Copy | Drop | Truncate | Create Lookup Table

Column Name	Data Type	Nullable	Default	Primary Key
USN	VARCHAR2(10)	No	-	1
NAME	VARCHAR2(100)	Yes	-	-
ADDRESS	VARCHAR2(255)	Yes	-	-
BRANCHID	NUMBER	Yes	-	-
SEM	NUMBER	Yes	-	-

1 - 5

```
-- Creating AUTHOR table
CREATE TABLE Shivansh_AUTHOR (
    Authorid INT PRIMARY KEY,
    Authorname VARCHAR(100),
    Country VARCHAR(100),
    Age INT
);
```



SHIVANSH_AUTHOR

Table | Data | Indexes | Model | Constraints | Grants | Statistics | UI Defaults | Triggers | Dependencies | SQL

Add Column | Modify Column | Rename Column | Drop Column | Rename | Copy | Drop | Truncate | Create Lookup Table

Column Name	Data Type	Nullable	Default	Primary Key
AUTHORID	NUMBER	No	-	1
AUTHORNAME	VARCHAR2(100)	Yes	-	-
COUNTRY	VARCHAR2(100)	Yes	-	-
AGE	NUMBER	Yes	-	-

1 - 4

```
-- Creating BOOK table
CREATE TABLE Shivansh_BOOK (
    Bookid INT PRIMARY KEY,
```

```
    Book_name VARCHAR(100),
    Authorid INT,
    Publisher VARCHAR(100),
    Branchid INT,
    FOREIGN KEY (Authorid) REFERENCES Shivansh_AUTHOR(Authorid),
    FOREIGN KEY (Branchid) REFERENCES Shivansh_BRANCH(Branchid)
);
```

SHIVANSH_BOOK

Table | Data | Indexes | Model | Constraints | Grants | Statistics | UI Defaults | Triggers | Dependencies | SQL

Add Column | Modify Column | Rename Column | Drop Column | Rename | Copy | Drop | Truncate | Create Lookup Table

Column Name	Data Type	Nullable	Default	Primary Key
BOOKID	NUMBER	No	-	1
BOOK_NAME	VARCHAR2(100)	Yes	-	-
AUTHORID	NUMBER	Yes	-	-
PUBLISHER	VARCHAR2(100)	Yes	-	-
BRANCHID	NUMBER	Yes	-	-

1 - 5

```
-- Creating BORROW table
CREATE TABLE Shivansh_BORROW (
    USN VARCHAR(10),
    Bookid INT,
    Borrowed_Date DATE,
    PRIMARY KEY (USN, Bookid),
    FOREIGN KEY (USN) REFERENCES Shivansh_STUDENT(USN),
    FOREIGN KEY (Bookid) REFERENCES Shivansh_BOOK(Bookid)
);
```

SHIVANSH_BORROW

Table | Data | Indexes | Model | Constraints | Grants | Statistics | UI Defaults | Triggers | Dependencies | SQL

Add Column | Modify Column | Rename Column | Drop Column | Rename | Copy | Drop | Truncate | Create Lookup Table

Column Name	Data Type	Nullable	Default	Primary Key
USN	VARCHAR2(10)	No	-	1
BOOKID	NUMBER	No	-	2
BORROWED_DATE	DATE	Yes	-	-

1 - 3

i) List the details of students who are studying in the 2nd semester of MCA.

Command:

```
SELECT *
FROM Shivansh_STUDENT
WHERE Branchid = 1 AND sem = 2;
```

Output:

Results | Explain | Describe | Saved SQL | History

USN	NAME	ADDRESS	BRANCHID	SEM
S01	Shivansh	123 Main St	1	2
S03	Priya	789 Oak St	1	2

2 rows returned in 0.02 seconds CSV Export

ii) List the students who have not borrowed any books.

Command:

```
SELECT s.*
FROM Shivansh_STUDENT s
LEFT JOIN Shivansh_BORROW b ON s.USN = b.USN
WHERE b.USN IS NULL;
```

Output:

Results Explain Describe Saved SQL History

USN	NAME	ADDRESS	BRANCHID	SEM
S02	Rahul	456 Park Ave	2	3

1 rows returned in 0.00 seconds CSV Export

iii) Display the USN, student name, branch name, book name, author name, and borrowed date of 2nd-sem MCA students who borrowed books.

Command:

```
SELECT s.USN, s.Name AS Student_name, br.Branchname, b.Book_name, a.Authorname,
bo.Borrowed_Date

FROM Shivansh_STUDENT s

JOIN Shivansh_BORROW bo ON s.USN = bo.USN

JOIN Shivansh_BOOK b ON bo.Bookid = b.Bookid

JOIN Shivansh_AUTHOR a ON b.Authorid = a.Authorid

JOIN Shivansh_BRANCH br ON s.Branchid = br.Branchid

WHERE s.Branchid = 1 AND s.sem = 2;
```

Output:

Results Explain Describe Saved SQL History

USN	STUDENT_NAME	BRANCHNAME	BOOK_NAME	AUTHORNAME	BORROWED_DATE
S01	Shivansh	MCA	Harry Potter	JK Rowling	15-JAN-24
S01	Shivansh	MCA	1984	George Orwell	10-FEB-24
S03	Priya	MCA	Adventures of Tom Sawyer	Mark Twain	12-MAR-24

3 rows returned in 0.00 seconds CSV Export

iv) Display the number of books written by each Author.

Command:

```
SELECT a.Authorname, COUNT(b.Bookid) AS Number_of_books

FROM Shivansh_AUTHOR a

JOIN Shivansh_BOOK b ON a.Authorid = b.Authorid

GROUP BY a.Authorname;
```

Output:

Results Explain Describe Saved SQL History

AUTHORNAME	NUMBER_OF_BOOKS
Mark Twain	1
JK Rowling	1
George Orwell	1
Leo Tolstoy	1
Agatha Christie	1

5 rows returned in 0.11 seconds CSV Export

v) Display the student details who borrowed more than two books.

Command:

```
SELECT s.*
FROM Shivansh_STUDENT s
JOIN Shivansh_BORROW bo ON s.USN = bo.USN
GROUP BY s.USN
HAVING COUNT(bo.Bookid) > 2;
```

Output:

Results Explain Describe Saved SQL History

no data found

vi) Display the student details who borrowed books of more than one Author.

Command:

```
SELECT s.*
FROM Shivansh_STUDENT s
JOIN Shivansh_BORROW bo ON s.USN = bo.USN
JOIN Shivansh_BOOK b ON bo.Bookid = b.Bookid
GROUP BY s.USN
HAVING COUNT(DISTINCT b.Authorid) > 1;
```

Output:

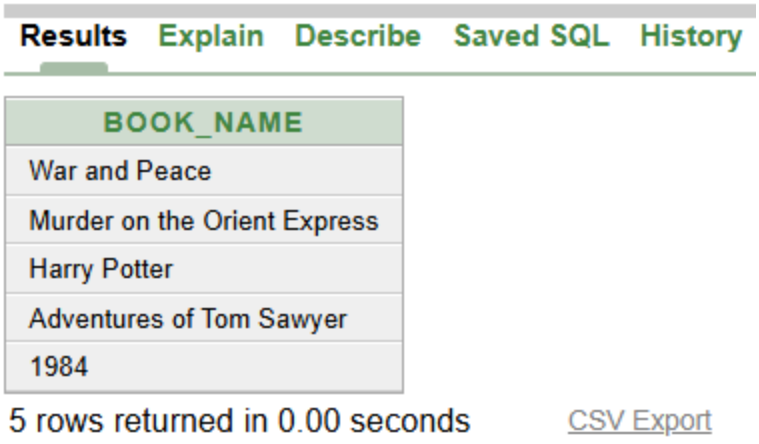
no data found

vii) Display the book names in descending order of their names.

Command:

```
SELECT Book_name
FROM Shivansh_BOOK
ORDER BY Book_name DESC;
```

Output:



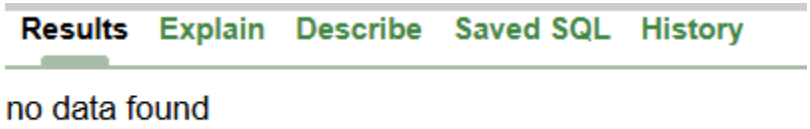
viii) List the details of students who borrowed the books which are all published by the same Publisher.

Command:

```
SELECT s.*
FROM Shivansh_STUDENT s
JOIN Shivansh_BORROW bo ON s.USN = bo.USN
JOIN Shivansh_BOOK b ON bo.Bookid = b.Bookid
WHERE b.Publisher IN (
  SELECT Publisher
  FROM Shivansh_BOOK
  GROUP BY Publisher
  HAVING COUNT(DISTINCT Bookid) > 1
```

);

Output:



7. Create the following schema.
STUDENT(USN,name,date_of_birth,branch,mark1,mark2,mark3,total,GPA)

Execute the following queires:

i) Update the column total by adding the column mark1,mark2,mark3.

ii) Find the GPA score of all the students.

iii) Find the students who born on a particular year of birth from the date _of_birth column.

iv) List the students who are studying in a particular branch of study.

v) Find the maximum GPA score of the student branch-wise.

vi) Find the students whose name starts with alphabet "S".

vii) Find the students whose name ends with alphapets "AR".

viii) Delete the student details whose USN is given as 1001.

Command to create the schema:

```
CREATE TABLE Shivanshs_STUDENT (
   USN VARCHAR(10) PRIMARY KEY,
   name VARCHAR(100),
   date_of_birth DATE,
   branch VARCHAR(50),
   mark1 INT,
   mark2 INT,
   mark3 INT,
   total INT,
   GPA DECIMAL(3, 2)
);
```

Output:

SHIVANSHS_STUDENT

Table | Data | Indexes | Model | Constraints | Grants | Statistics | UI Defaults | Triggers | Dependencies | SQL

Add Column | Modify Column | Rename Column | Drop Column | Rename | Copy | Drop | Truncate | Create Lookup Table

Column Name	Data Type	Nullable	Default	Primary Key
USN	VARCHAR2(10)	No	-	1
NAME	VARCHAR2(100)	Yes	-	-
DATE_OF_BIRTH	DATE	Yes	-	-
BRANCH	VARCHAR2(50)	Yes	-	-
MARK1	NUMBER	Yes	-	-
MARK2	NUMBER	Yes	-	-
MARK3	NUMBER	Yes	-	-
TOTAL	NUMBER	Yes	-	-
GPA	NUMBER(3,2)	Yes	-	-

1 - 9

i) Update the column total by adding the columns mark1, mark2, and mark3.

Command:

```
UPDATE Shivanshs_STUDENT
SET total = mark1 + mark2 + mark3;
```

Output:

SHIVANSHS_STUDENT

Table | Data | Indexes | Model | Constraints | Grants | Statistics | UI Defaults | Triggers | Dependencies | SQL

Query | Count Rows | Insert Row

EDIT	USN	NAME	DATE_OF_BIRTH	BRANCH	MARK1	MARK2	MARK3	TOTAL	GPA
	1001	Shivansh	14-MAY-00	MCA	85	90	88	263	-
	1002	Rahul	22-MAR-99	CSE	78	84	80	242	-
	1003	Simran	10-JAN-01	ECE	92	88	95	275	-
	1004	Arjun	05-DEC-00	Mechanical	67	72	65	204	-
	1005	Sagar	23-NOV-02	MCA	88	79	81	248	-

row(s) 1 - 5 of 5

ii) Find the GPA score of all the students

Command:

```
UPDATE Shivanshs_STUDENT
SET GPA = ROUND((mark1 + mark2 + mark3) / 3, 2);
```

Output:

SHIVANSHS_STUDENT

Table | Data | Indexes | Model | Constraints | Grants | Statistics | UI Defaults | Triggers | Dependencies | SQL

Query | Count Rows | Insert Row

EDIT	USN	NAME	DATE_OF_BIRTH	BRANCH	MARK1	MARK2	MARK3	TOTAL	GPA
	1001	Shivansh	14-MAY-00	MCA	85	90	88	263	87.67
	1002	Rahul	22-MAR-99	CSE	78	84	80	242	80.67
	1003	Simran	10-JAN-01	ECE	92	88	95	275	91.67
	1004	Arjun	05-DEC-00	Mechanical	67	72	65	204	68
	1005	Sagar	23-NOV-02	MCA	88	79	81	248	82.67

row(s) 1 - 5 of 5

Download

iii) Find the students who were born in a particular year (e.g., 2000)

Command:

```
SELECT *
FROM Shivanshs_STUDENT
WHERE EXTRACT(YEAR FROM date_of_birth) = 2000;
```

Output:

Results | Explain | Describe | Saved SQL | History

USN	NAME	DATE_OF_BIRTH	BRANCH	MARK1	MARK2	MARK3	TOTAL	GPA
1001	Shivansh	14-MAY-00	MCA	85	90	88	263	87.67
1004	Arjun	05-DEC-00	Mechanical	67	72	65	204	68

2 rows returned in 0.00 seconds CSV Export

iv) List the students who are studying in a particular branch (e.g., 'MCA')

Command:

```
SELECT *
FROM Shivanshs_STUDENT
WHERE branch = 'MCA';
```

Output:

Results Explain Describe Saved SQL History

USN	NAME	DATE_OF_BIRTH	BRANCH	MARK1	MARK2	MARK3	TOTAL	GPA
1001	Shivansh	14-MAY-00	MCA	85	90	88	263	87.67
1005	Sagar	23-NOV-02	MCA	88	79	81	248	82.67

2 rows returned in 0.00 seconds CSV Export

v) Find the maximum GPA score of the student branch-wise

Command:

```
SELECT branch, MAX(GPA) AS max_GPA
FROM Shivanshs_STUDENT
GROUP BY branch;
```

Output:

Results Explain Describe Saved SQL History

BRANCH	MAX_GPA
Mechanical	68
CSE	80.67
MCA	87.67
ECE	91.67

4 rows returned in 0.00 seconds CSV Export

vi) Find the students whose names start with the letter 'S'

Command:

```
SELECT *
FROM Shivanshs_STUDENT
WHERE name LIKE 'S%';
```

Output:

Results Explain Describe Saved SQL History

USN	NAME	DATE_OF_BIRTH	BRANCH	MARK1	MARK2	MARK3	TOTAL	GPA
1001	Shivansh	14-MAY-00	MCA	85	90	88	263	87.67
1003	Simran	10-JAN-01	ECE	92	88	95	275	91.67
1005	Sagar	23-NOV-02	MCA	88	79	81	248	82.67

3 rows returned in 0.00 seconds CSV Export

vii) Find the students whose names end with the letters 'AR'

Command:

```
SELECT *
FROM Shivanshs_STUDENT
WHERE name LIKE '%AR';
```

Output:

viii) Delete the student details whose USN is '1001'

Command:

```
DELETE FROM Shivanshs_STUDENT
WHERE USN = '1001';
```

Output:

SHIVANSHS_STUDENT

Table | Data | Indexes | Model | Constraints | Grants | Statistics | UI Defaults | Triggers | Dependencies | SQL

Query | Count Rows | Insert Row

EDIT	USN	NAME	DATE_OF_BIRTH	BRANCH	MARK1	MARK2	MARK3	TOTAL	GPA
	1002	Rahul	22-MAR-99	CSE	78	84	80	242	80.67
	1003	Simran	10-JAN-01	ECE	92	88	95	275	91.67
	1004	Arjun	05-DEC-00	Mechanical	67	72	65	204	68
	1005	Sagar	23-NOV-02	MCA	88	79	81	248	82.67

row(s) 1 - 4 of 4

Download