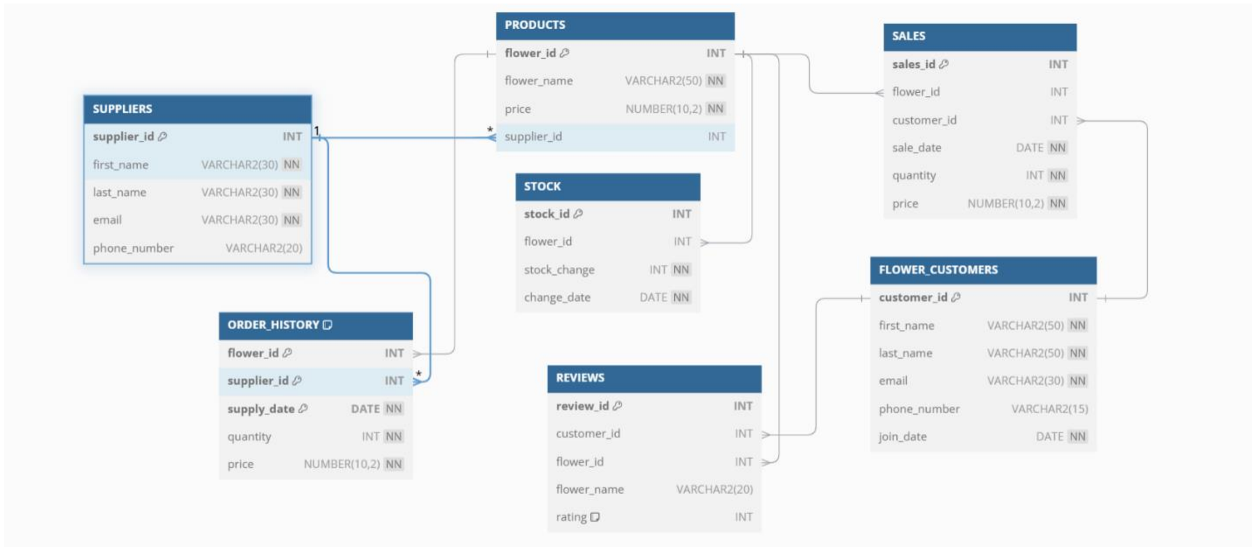
DB PROJECT : Database for Flower Store Management System
Focsa Iulia-Stefania
Group 1074 G

Database for Flower Store Management System

I. Description of the project

This project is a database management system for a flower store. It manages information related to flowers , customers, suppliers, sales, inventory (stock), and reviews.The database design includes seven tables, and it provides flexible data management for running the store. Features include tracking stock levels, managing supplier and customer information, recording transactions and handling reviews

II. Database schema



1. **The PRODUCTS table** stores information about the flowers available for sale, including their name, description, price, and the supplier providing them.
2. **The SALES table** tracks transactions, including which flowers were purchased, by whom, the quantity sold, and the sale price at the time.
3. **The FLOWER_CUSTOMERS table** maintains details of customers, such as their contact information and the date they joined the system.
4. **The SUPPLIERS table** contains information about suppliers, including their names and contact details, who provide the flowers.

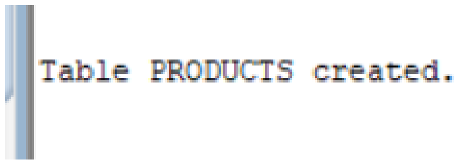
5. **The ORDER_HISTORY table** logs orders made to suppliers, detailing which products were ordered, when, the quantity, and the price paid. This table exists to handle the many-to-many relationship between the PRODUCTS table and the SUPPLIERS table. It connects flower_id (from PRODUCTS) with supplier_id (from SUPPLIERS) and also stores additional details like supply_date, quantity, and price.
6. **The STOCK table** tracks changes in stock levels for flowers, including additions and deductions, with timestamps for inventory management.
7. **The REVIEWS table** records customer feedback and ratings for specific flowers, including optional review text and the date of the review.

III. Constructing the database

```
--Create the Suppliers table.
    CREATE TABLE SUPPLIERS (
supplier_id INT PRIMARY KEY,
first_name VARCHAR2(30) NOT NULL,
last_name VARCHAR2(30) NOT NULL,
email VARCHAR(30) UNIQUE NOT NULL,
phone_number VARCHAR2(20)
);
```

```
Table SUPPLIERS created.
```

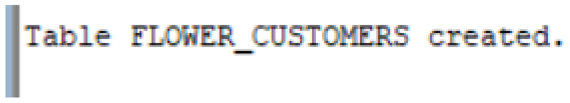
```
--Create the Products table.
      CREATE TABLE PRODUCTS (
  flower_id INT PRIMARY KEY,
  flower_name VARCHAR2(50) NOT NULL,
  price NUMBER(10, 2) NOT NULL,
  supplier_id INT NOT NULL,
  FOREIGN KEY (supplier_id) REFERENCES SUPPLIERS(supplier_id)
);
```



```
Table PRODUCTS created.
```

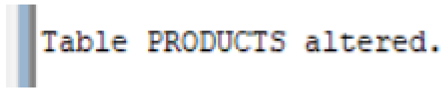
DB PROJECT : Database for Flower Store Management System
Focsa Iulia-Stefania
Group 1074 G

```
--Create a table for the customers.

    CREATE TABLE FLOWER_CUSTOMERS (

  customer_id INT PRIMARY KEY,

  first_name VARCHAR2(50) NOT NULL,

  last_name VARCHAR2(50) NOT NULL,

  email VARCHAR2(30) UNIQUE NOT NULL,

  phone_number VARCHAR2(15),

  join_date DATE NOT NULL

);
```



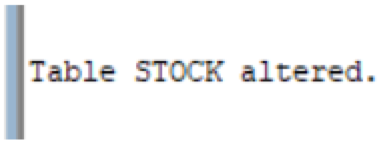
```
Table FLOWER_CUSTOMERS created.
```

```
-- Add a stock column to PRODUCTS table with a default value.

ALTER TABLE PRODUCTS ADD stock NUMBER DEFAULT 0;
```



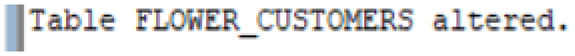
```
Table PRODUCTS altered.
```

```
 -- Remove stock_change column from STOCK table.

ALTER TABLE STOCK DROP COLUMN stock_change;
```



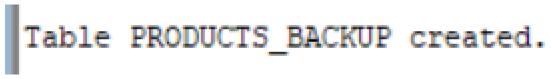
```
Table STOCK altered.
```

```
-- Modify phone_number data type in FLOWER_CUSTOMERS table to VARCHAR2(20) .

ALTER TABLE FLOWER_CUSTOMERS MODIFY phone_number VARCHAR2(20);
```



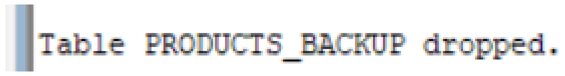
```
Table FLOWER_CUSTOMERS altered.
```

```
-- Create a backup table for the products table.

CREATE TABLE PRODUCTS_BACKUP AS SELECT * FROM PRODUCTS;
```



```
Table PRODUCTS_BACKUP created.
```

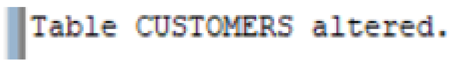
```
-- Drop the backup table.

DROP TABLE PRODUCTS_BACKUP;
```



```
Table PRODUCTS_BACKUP dropped.
```

DB PROJECT : Database for Flower Store Management System
Focsa Iulia-Stefania
Group 1074 G

-- Add a default join_date column to FLOWER_CUSTOMERS.



```
Table CUSTOMERS altered.
```

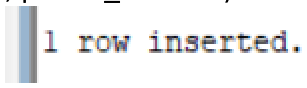
ALTER TABLE CUSTOMERS ADD join_date DATE DEFAULT SYSDATE;

IV. Using DML statements: INSERT, UPDATE, DELETE, and MERGE

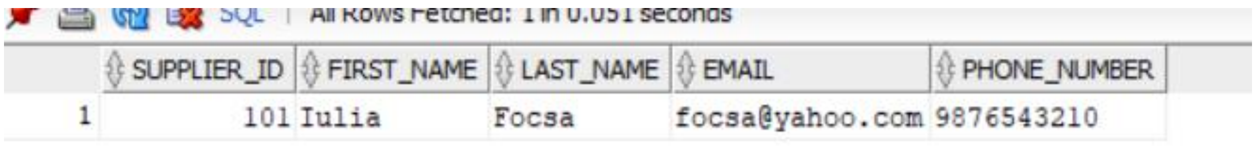
-- Insert supplier data.

INSERT INTO SUPPLIERS (supplier_id, first_name, last_name, email, phone_number)

VALUES (101, 'Iulia', 'Focsa', 'focsa@yahoo.com', '9876543210');



```
1 row inserted.
```



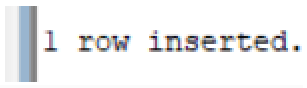
SQL | All Rows Fetched: 1 in 0.051 seconds

	SUPPLIER_ID	FIRST_NAME	LAST_NAME	EMAIL	PHONE_NUMBER
1	101	Iulia	Focsa	focsa@yahoo.com	9876543210

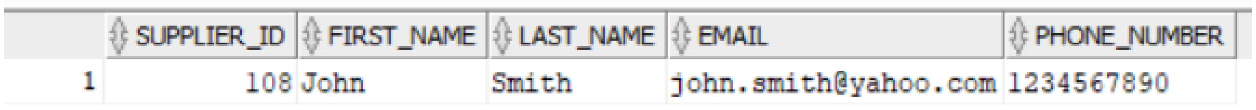
-- Insert supplier data.

INSERT INTO SUPPLIERS (supplier_id, first_name, last_name, email, phone_number)

VALUES (108, 'John', 'Smith', 'john.smith@yahoo.com', '1234567890');



```
1 row inserted.
```

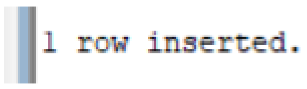


	SUPPLIER_ID	FIRST_NAME	LAST_NAME	EMAIL	PHONE_NUMBER
1	108	John	Smith	john.smith@yahoo.com	1234567890

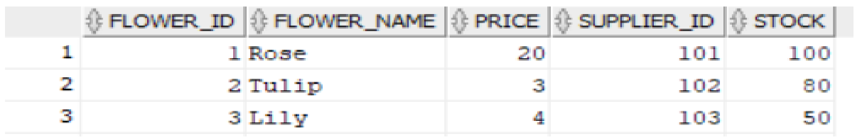
--Insert product data.

INSERT INTO PRODUCTS (flower_id, flower_name, price, supplier_id, stock)

VALUES (3, 'Lily', 4.00, 103, 50);



```
1 row inserted.
```

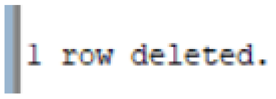


	FLOWER_ID	FLOWER_NAME	PRICE	SUPPLIER_ID	STOCK
1	1	Rose	20	101	100
2	2	Tulip	3	102	80
3	3	Lily	4	103	50

DB PROJECT : Database for Flower Store Management System
Focsa Iulia-Stefania
Group 1074 G

--Delete the review with the review_id=5.

DELETE FROM REVIEWS WHERE review_id=5;



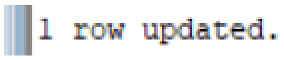
```
1 row deleted.
```

--Set the price to 19 and the stock to 50 to the product with the flower_id=1.

UPDATE PRODUCTS

SET price = 19, stock= 50

WHERE flower_id = 1;



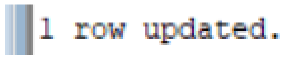
```
1 row updated.
```

--Change the quantity sold to 10 where the customer_id=1002.

UPDATE SALES

SET quantity=10

WHERE customer_id=1002;

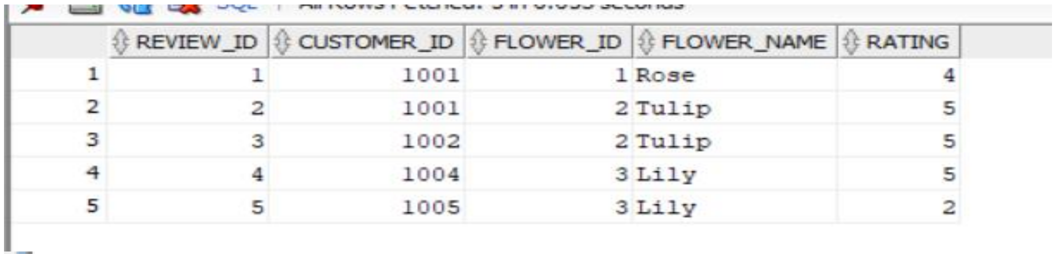


```
1 row updated.
```

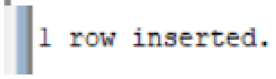
--Insert into the reviews table a new review.

INSERT INTO REVIEWS (review_id, customer_id, flower_id, flower_name, rating)

VALUES (5, 1005, 3, 'Lily', 2);



	REVIEW_ID	CUSTOMER_ID	FLOWER_ID	FLOWER_NAME	RATING
1	1	1001	1	Rose	4
2	2	1001	2	Tulip	5
3	3	1002	2	Tulip	5
4	4	1004	3	Lily	5
5	5	1005	3	Lily	2

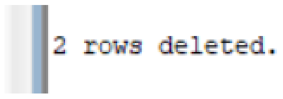


```
1 row inserted.
```

--Delete from the products table the ratings that are smaller then 5.

DELETE FROM REVIEWS

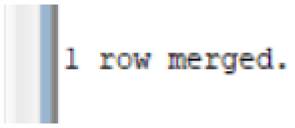
WHERE rating < 5;



```
2 rows deleted.
```

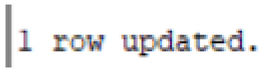
DB PROJECT : Database for Flower Store Management System
Focsa Iulia-Stefania
Group 1074 G

```
--Add a new customer or update their phone number if they already exist.

MERGE INTO flower_customers fc

USING (

   SELECT 1020 AS customer_id, 'Wow' AS first_name, 'Martinez' AS last_name,

          'wowm@yahoo.com' AS email, '1234567888' AS phone_number

   FROM dual

) src

ON (fc.customer_id = src.customer_id)

WHEN MATCHED THEN

   UPDATE SET fc.phone_number = src.phone_number

WHEN NOT MATCHED THEN

   INSERT (customer_id, first_name, last_name, email, phone_number, join_date)

   VALUES (src.customer_id, src.first_name, src.last_name, src.email, src.phone_number,
SYSDATE);
```

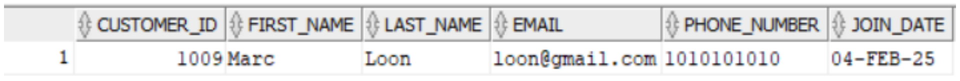


```
1 row merged.
```

```
--Change the phone number of the customer with the id=1009 to 1010101010.

UPDATE flower_customers

SET phone_number = 1010101010

WHERE customer_id = 1009;
```



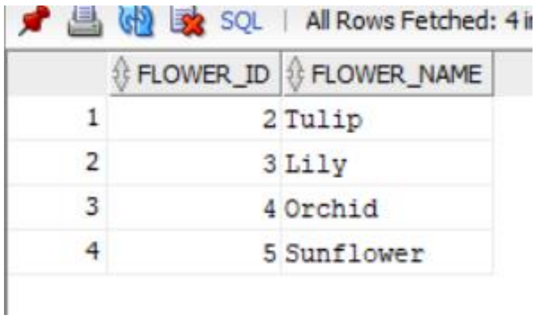
```
1 row updated.
```



	CUSTOMER_ID	FIRST_NAME	LAST_NAME	EMAIL	PHONE_NUMBER	JOIN_DATE
1	1009	Marc	Loon	loon@gmail.com	1010101010	04-FEB-25

V. Diverse and relevant SELECT statements for the project theme

```
--1.Show the flowers which have the flower_id 2,3,4 or 5.

SELECT flower_id, flower_name

FROM PRODUCTS WHERE flower_id IN (2, 3, 4, 5);
```

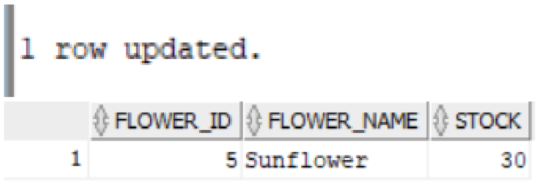


	FLOWER_ID	FLOWER_NAME
1	2	Tulip
2	3	Lily
3	4	Orchid
4	5	Sunflower

DB PROJECT : Database for Flower Store Management System
Focsa Iulia-Stefania
Group 1074 G

--2.Change the stock = 30 where the flower_id is 5.

```
UPDATE PRODUCTS

SET stock = 30

WHERE flower_id = 5;
```

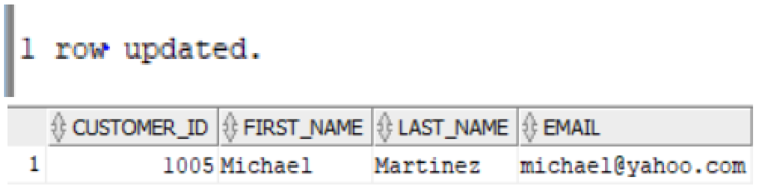


```
1 row updated.
```

	FLOWER_ID	FLOWER_NAME	STOCK
1	5	Sunflower	30

--3.Change the email of the person with the customer_id 1005, to michael@yahoo.com .

```
UPDATE FLOWER_CUSTOMERS

SET email = 'michael@yahoo.com'

WHERE customer_id = 1005;
```

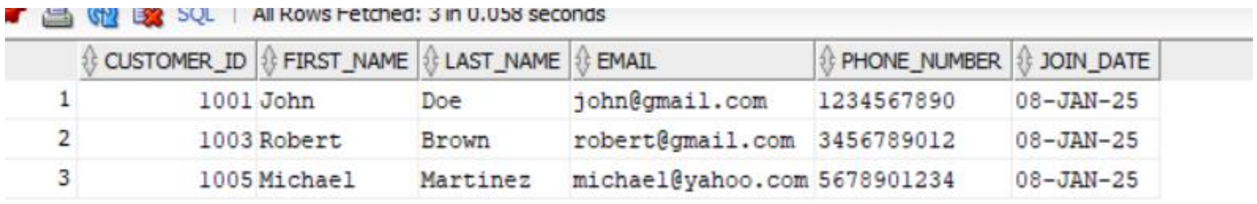


```
1 row updated.
```

	CUSTOMER_ID	FIRST_NAME	LAST_NAME	EMAIL
1	1005	Michael	Martinez	michael@yahoo.com

--4.Show all the cutomers that bought at least 5 products .

```
SELECT f.*

FROM flower_customers f, sales s

WHERE f.customer_id = s.customer_id AND s.quantity>=5 ;
```

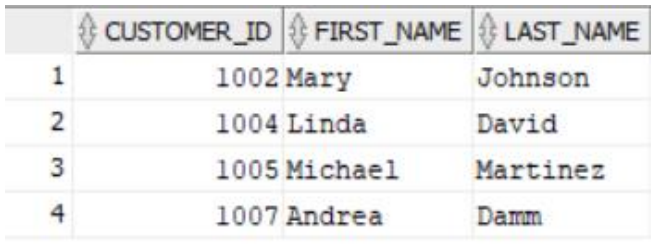


SQL | All Rows Fetched: 3 in 0.058 seconds

	CUSTOMER_ID	FIRST_NAME	LAST_NAME	EMAIL	PHONE_NUMBER	JOIN_DATE
1	1001	John	Doe	john@gmail.com	1234567890	08-JAN-25
2	1003	Robert	Brown	robert@gmail.com	3456789012	08-JAN-25
3	1005	Michael	Martinez	michael@yahoo.com	5678901234	08-JAN-25

--5.select the customers that have the letter a in their first_name.

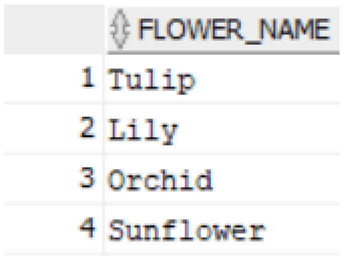
```
SELECT customer_id, first_name, last_name

FROM flower_customers

WHERE first_name LIKE '%a%';
```



	CUSTOMER_ID	FIRST_NAME	LAST_NAME
1	1002	Mary	Johnson
2	1004	Linda	David
3	1005	Michael	Martinez
4	1007	Andrea	Damm

--6.select the flower_name with the price >= 1 and price <= 10.

```
SELECT flower_name FROM products

WHERE price BETWEEN 1 AND 10;
```

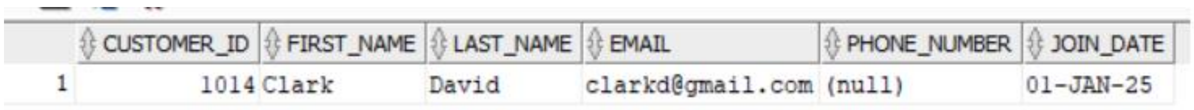


	FLOWER_NAME
1	Tulip
2	Lily
3	Orchid
4	Sunflower

DB PROJECT : Database for Flower Store Management System
Focsa Iulia-Stefania
Group 1074 G

--7.Show the customers that do not have a phone number inserted.

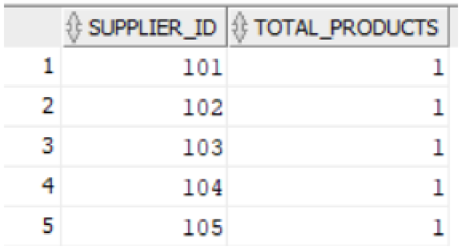
```
SELECT *

FROM flower_customers

WHERE phone_number IS NULL;
```



	CUSTOMER_ID	FIRST_NAME	LAST_NAME	EMAIL	PHONE_NUMBER	JOIN_DATE
1	1014	Clark	David	clarkd@gmail.com	(null)	01-JAN-25

--8.Show the total number of suppliers that supplied each product.

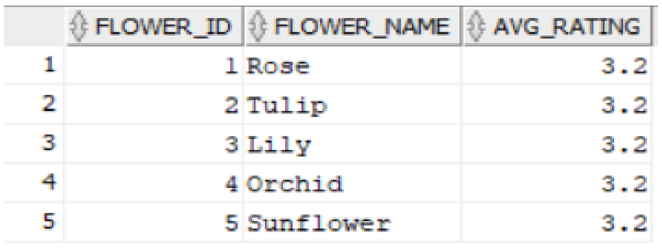
```
SELECT supplier_id, COUNT(supplier_id) AS total_products

FROM PRODUCTS

GROUP BY supplier_id;
```



	SUPPLIER_ID	TOTAL_PRODUCTS
1	101	1
2	102	1
3	103	1
4	104	1
5	105	1

--9. Select the flowers with an average rating bigger than 3.

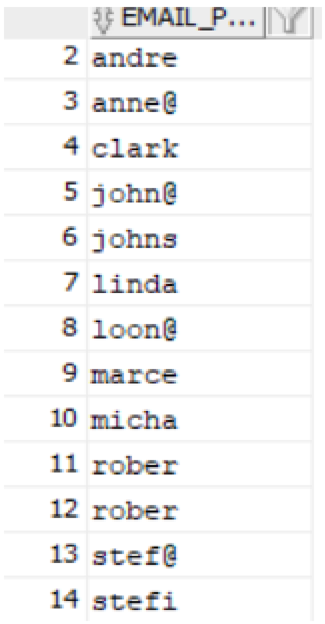
```
SELECT p.flower_id,p.flower_name, AVG(r.rating) AS avg_rating

FROM REVIEWS r,products p

GROUP BY p.flower_id,p.flower_name

HAVING AVG(r.rating) > 3;
```



	FLOWER_ID	FLOWER_NAME	AVG_RATING
1	1	Rose	3.2
2	2	Tulip	3.2
3	3	Lily	3.2
4	4	Orchid	3.2
5	5	Sunflower	3.2

--10.Select the first 5 characters from the email section for the customers.

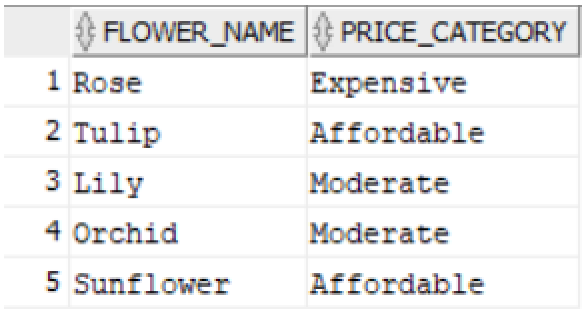
```
SELECT SUBSTR(email, 1, 5) AS email_prefix

FROM FLOWER_CUSTOMERS;
```



	EMAIL_P...
2	andre
3	anne@
4	clark
5	john@
6	johns
7	linda
8	loon@
9	marce
10	micha
11	rober
12	rober
13	stef@
14	stefi

--11.Show the expensive flowers(price >19), moderate flowers(price > = 4 and price < = 19) and the affordable flowers(the rest).

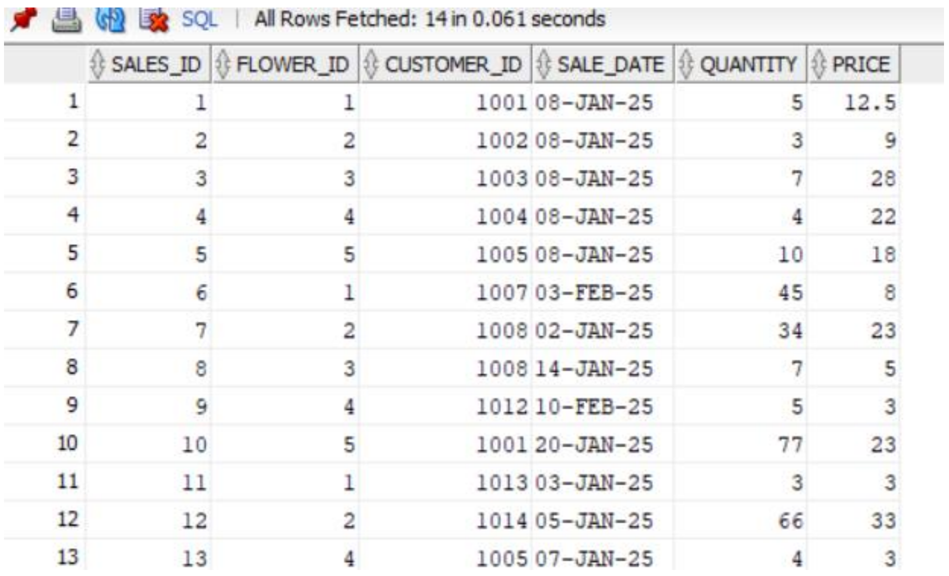
```
SELECT flower_name,
    CASE
      WHEN price > 19 THEN 'Expensive'
      WHEN price BETWEEN 4 AND  19 THEN 'Moderate'
      ELSE 'Affordable'
    END AS price_category
FROM PRODUCTS;
```



	FLOWER_NAME	PRICE_CATEGORY
1	Rose	Expensive
2	Tulip	Affordable
3	Lily	Moderate
4	Orchid	Moderate
5	Sunflower	Affordable

--12. select the customers that bought a product later than 01 jan 2025, but sooner than 10 feb 2025.

```
SELECT *
FROM sales
WHERE sale_date BETWEEN TO_DATE('01 JAN 2025', 'DD MON YYYY') AND TO_DATE('10 FEB 2025', 'DD MON YYYY');
```



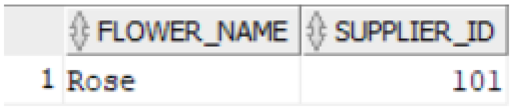
All Rows Fetched: 14 in 0.061 seconds

	SALES_ID	FLOWER_ID	CUSTOMER_ID	SALE_DATE	QUANTITY	PRICE
1	1	1	1001	08-JAN-25	5	12.5
2	2	2	1002	08-JAN-25	3	9
3	3	3	1003	08-JAN-25	7	28
4	4	4	1004	08-JAN-25	4	22
5	5	5	1005	08-JAN-25	10	18
6	6	1	1007	03-FEB-25	45	8
7	7	2	1008	02-JAN-25	34	23
8	8	3	1008	14-JAN-25	7	5
9	9	4	1012	10-FEB-25	5	3
10	10	5	1001	20-JAN-25	77	23
11	11	1	1013	03-JAN-25	3	3
12	12	2	1014	05-JAN-25	66	33
13	13	4	1005	07-JAN-25	4	3

DB PROJECT : Database for Flower Store Management System
Focsa Iulia-Stefania
Group 1074 G

--13. Select the flower name and supplier ID from the PRODUCTS table, for the supplier whose supplier ID matches that of the person with the first name 'Iulia'.

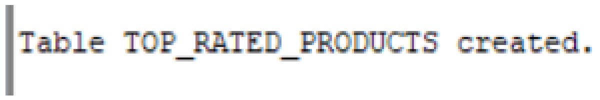
```
SELECT flower_name , supplier_id

FROM products

WHERE supplier_id = (SELECT supplier_id FROM suppliers WHERE first_name = 'Iulia');
```



	FLOWER_NAME	SUPPLIER_ID
1	Rose	101

--14. Create a new table who prints the flower name and the average rating >= to 4.

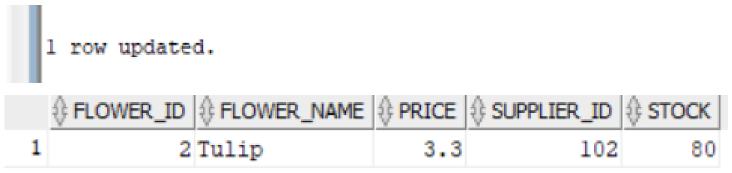
```
CREATE TABLE Top_Rated_Products AS

SELECT p.flower_name, AVG(r.rating) AS avg_rating

FROM products p,reviews r

WHERE p.flower_id = r.flower_id

GROUP BY p.flower_name

HAVING AVG(r.rating) >= 4;
```



```
Table TOP_RATED_PRODUCTS created.
```

--15. Raise with 110% the price of the product with the supplier_id=102;

```
UPDATE products

SET price = price * 1.1

WHERE supplier_id = 102;
```

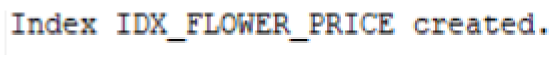


```
1 row updated.
```

	FLOWER_ID	FLOWER_NAME	PRICE	SUPPLIER_ID	STOCK
1	2	Tulip	3.3	102	80

--16. Create an index for the price of the flowers.

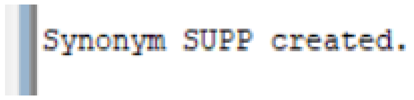
```
CREATE INDEX idx_flower_price ON PRODUCTS(price);
```



```
Index IDX_FLOWER_PRICE created.
```

--17. Create a synonym for the suppliers table.

```
CREATE SYNONYM supp FOR suppliers;
```

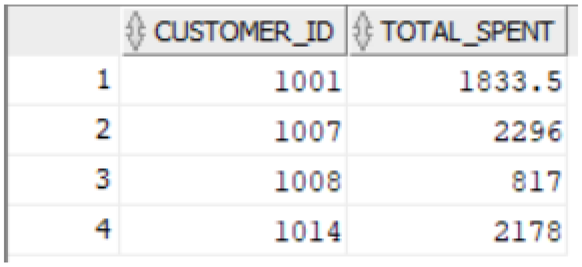


```
Synonym SUPP created.
```

DB PROJECT : Database for Flower Store Management System
Focsa Iulia-Stefania
Group 1074 G

--18.Show customers who have spent more than the average spending of all customers.

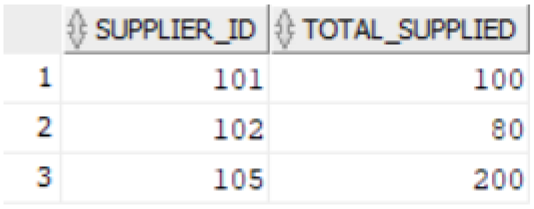
```
SELECT customer_id, SUM(price * quantity) AS total_spent

FROM sales s

GROUP BY customer_id

HAVING SUM(price * quantity) > (

   SELECT AVG(SUM(price * quantity))

   FROM sales

   GROUP BY customer_id

);
```



	CUSTOMER_ID	TOTAL_SPENT
1	1001	1833.5
2	1007	2296
3	1008	817
4	1014	2178

--19.Show the suppliers who have supplied more than 60 items.

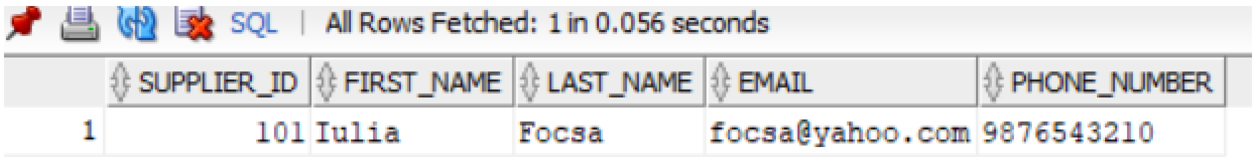
```
SELECT supplier_id, SUM(quantity) AS total_supplied

FROM order_history

GROUP BY supplier_id

HAVING SUM(quantity) > 60;
```



	SUPPLIER_ID	TOTAL_SUPPLIED
1	101	100
2	102	80
3	105	200

--20. Select the supplier with the supplier_id=101 from the suppliers table using the synonym.

```
SELECT *

FROM supp

WHERE supplier_id = 101;
```



All Rows Fetched: 1 in 0.056 seconds

	SUPPLIER_ID	FIRST_NAME	LAST_NAME	EMAIL	PHONE_NUMBER
1	101	Iulia	Focsa	focsa@yahoo.com	9876543210