**Group members:**

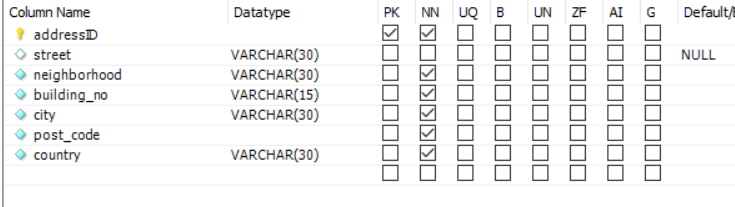
|  |  |  |  |
| --- | --- | --- | --- |
| **Group No** | **Student ID** | **Name** | **Project Topic** |
| **7** | **181805051** | **Şevval Şimşek** | **Supermarket Stock Control System** |
| **181805067** | **Buse Latife Beker** |
| **181805077** | **Emine Ece Coşkunçay** |

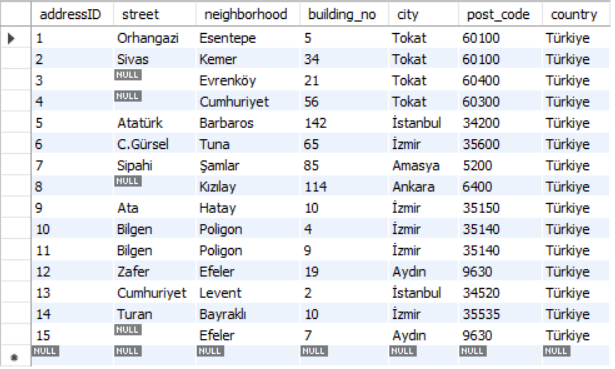
|  |  |
| --- | --- |
| **Query No** | **Requirements** |
| **1** | 1 table, at least 2 condition in WHERE clause |
| **2** | 1 table, retrieve all attributes and ORDER BY primary key. |
| **3** | 1 table, use BETWEEN … AND … |
| **4** | 1 table, use NOT NULL |
| **5** | 1 table, use MAX, MIN, AVG (Ex: Find max, min, avg salary of employee) |
| **6** | 2 table, use COUNT, SUM |
| **7** | use GROUP BY |
| **8** | use HAVING |
| **9** | use LIKE (substring comparision) |
| **10** | UPDATE one of the existing record in a table |
| **11** | Inner Join |
| **12** | Join |
| **13** | Join |
| **14** | Left Outer Join |
| **15** | Right Outer Join |
| **16** | Full Outer Join |
| **17** | Stored Procedure |
| **18** | Stored Procedure |
| **19** | View |
| **20** | View |

**The tables and data we use in the supermarket stock control system:**

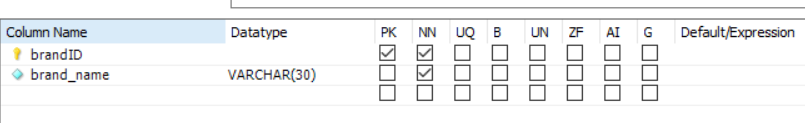
**We have 11 tables:**

**1-address table**

****

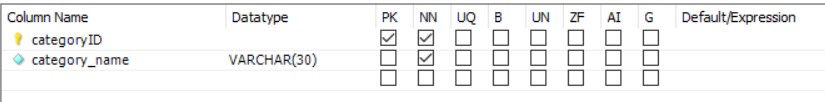
****

**2-brand table**

****

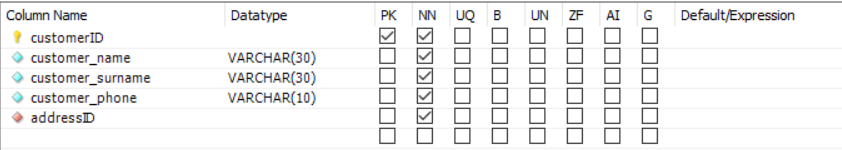
** **

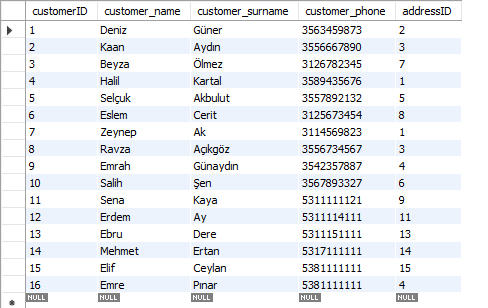
**3-category table**

****

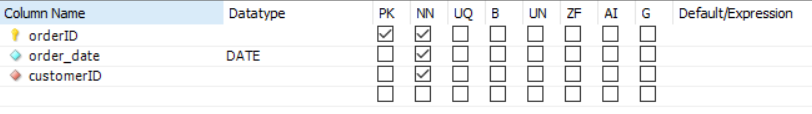
****

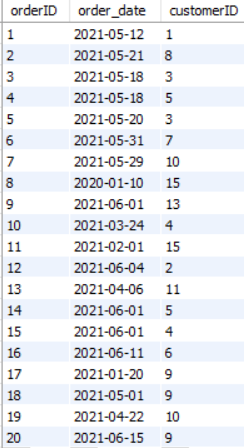
**4-customer table**

****

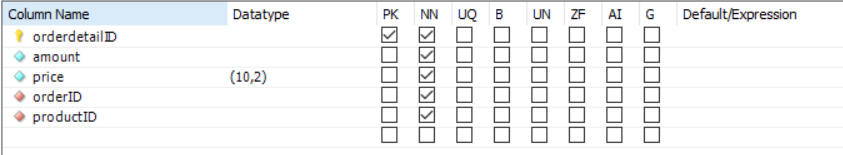
****

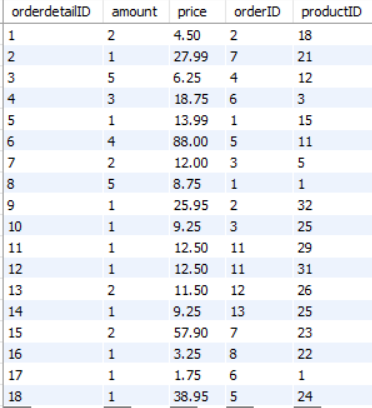
**5-customerorder table**

****

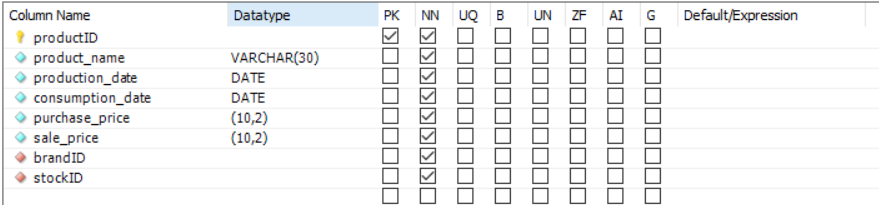
****

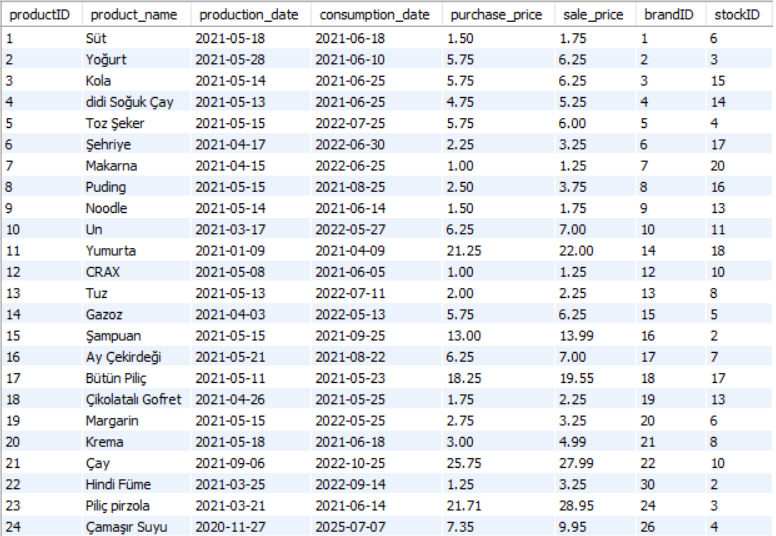
**6-orderdetail table**

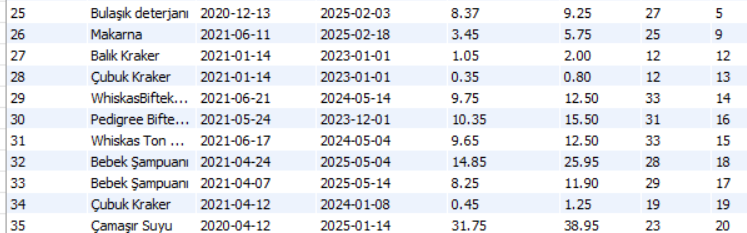
****

****

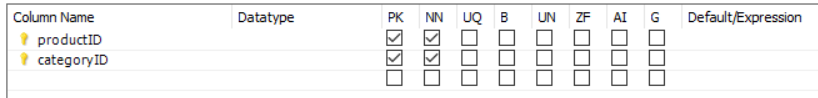
**7-product table**

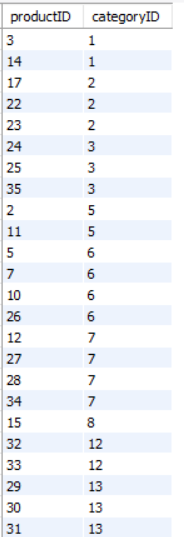
****

****

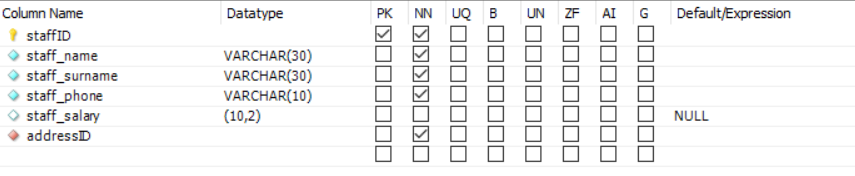
****

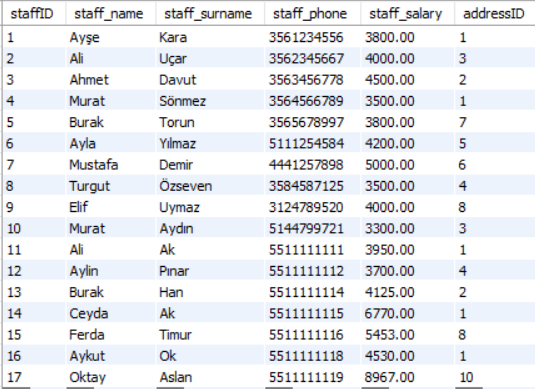
**8-product\_category table**

****

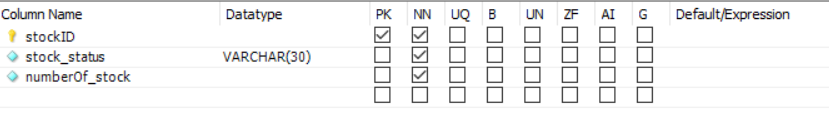
****

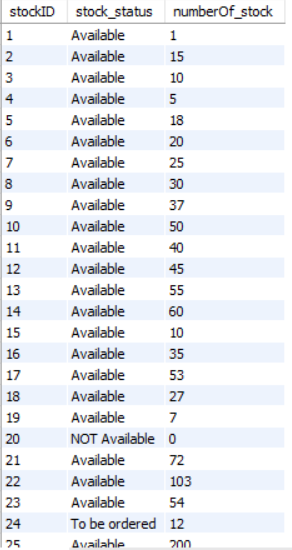
**9-staff table**

****

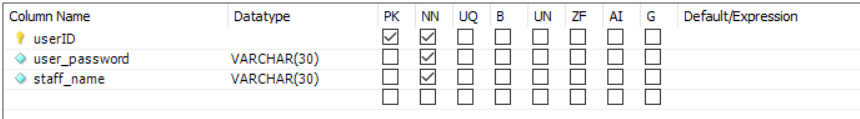
****

**10-stock table**

****

** **

**11-user table**

****

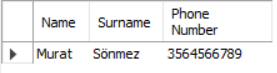
****

**Query 1: Retrieves the name, surname and phone information of the people whose name is Murat and whose salary is 3500 and above in the staff table.**

SELECT staff\_name AS "Name", staff\_surname AS "Surname", staff\_phone AS "Phone Number"

FROM staff

WHERE staff\_name= 'Murat' AND staff\_salary>=3500;

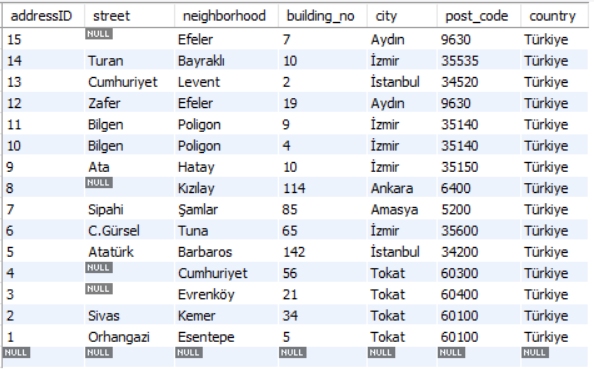


**Query 2: Retrieves the information in the address table in reverse(DESC) order by addressID.**

SELECT \*

FROM address

ORDER BY addressID DESC;



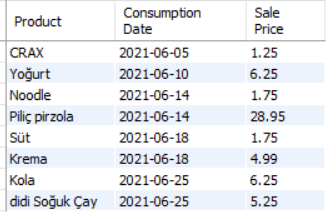
**Query3: In the product table, retrieves the product name, consumption date and sale price information of the products whose consumption dates are between 2021-06-01 and 2021-06-30 in order according to their consumption dates.**

SELECT product\_name AS "Product", consumption\_date AS "Consumption Date", sale\_price AS "Sale Price"

FROM product

WHERE consumption\_date BETWEEN '2021-06-01' AND '2021-06-30'

ORDER BY consumption\_date;



**Query4: In the address table, the street field retrieve the street, neighborhood and city information of the address data that is not null.**

SELECT street, neighborhood, city

FROM address

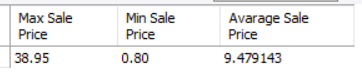
WHERE street IS NOT NULL;



**Query5: Finds the maximum selling price, minimum selling price, and average selling price in the Product table.**

SELECT MAX(sale\_price) AS "Max Sale Price", MIN(sale\_price) AS "Min Sale Price", AVG(sale\_price) AS "Avarage Sale Price"

FROM product;



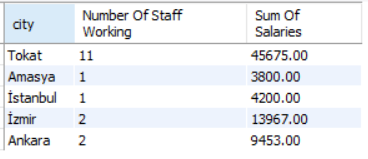
**Query6: In the address table, it retrieves how many staff there are in each city and the sum of the salaries of these staff (the total number of staff working in the same city and their salaries).**

SELECT A.city, COUNT(\*) AS "Number Of Staff Working", SUM(S.staff\_salary) AS "Sum Of Salaries"

FROM address A , staff S

WHERE A.addressID=S.addressID

GROUP BY A.city;



**Query7: For each address, retrieve the address ID, the street, the city and the number of customers who living at that address.**

SELECT A.addressID, A.street, A.city, COUNT(\*) AS "Number Of Customers"

FROM address A, customer C

WHERE A.addressID=C.addressID

GROUP BY A.addressID;



**Query8: For each address on which more than two customers live, retrieve the address ID, the street, the city, and the number of customers who living at that address.**

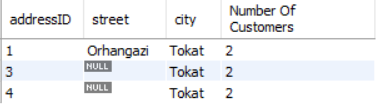
SELECT A.addressID, A.street, A.city, COUNT(\*) AS "Number Of Customers"

FROM address A, customer C

WHERE A.addressID=C.addressID

GROUP BY A.addressID

HAVING COUNT(\*) >= 2;

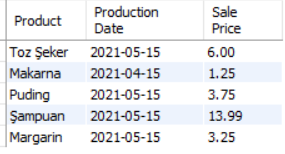


**Query 9: Retrieves the name, production date and sale price of all products mentioned 15 in the production date information in the product table.**

SELECT product\_name AS "Product", production\_date AS "Production Date", sale\_price AS "Sale Price"

FROM product

WHERE production\_date LIKE "%15%";



**Query 10: Updates the name data of the customer record with the customerID of 4 in the customer table to "Halis".**

UPDATE customer

SET customer\_name = "Halis"

WHERE customerID=4;

The updated record looks like this in the customer table:



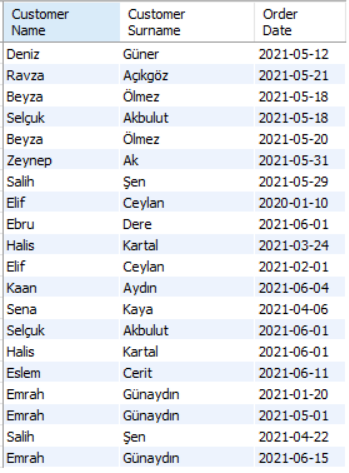
**Query 11: Retrieves each costumer's name, surname and their order date using INNER JOIN.**

SELECT C.customer\_name AS "Customer Name", C.customer\_surname AS "Customer Surname", O.order\_date AS "Order Date"

FROM customer C

INNER JOIN customerorder O

ON C.customerID=O.customerID;

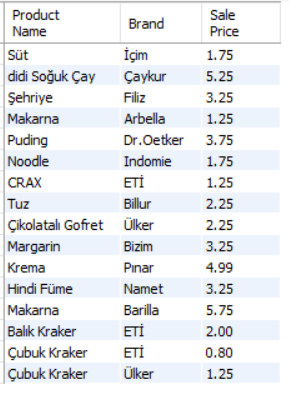


**Query12: Retrieves the product name, brand and sale price of all product whose sales price is less than 6 TL.**

SELECT P.product\_name AS "Product Name", B.brand\_name AS "Brand", P.sale\_price AS "Sale Price"

FROM product P, brand B

WHERE P.brandID=B.brandID AND sale\_price<6;



**Query13: Retrieves all product's full name, their category and their sale price.**

SELECT P.product\_name AS "Product Name", C.category\_name AS "Category", P.sale\_price AS "Sale Price"

FROM product P, product\_category PC, category C

WHERE P.productID=PC.productID AND PC.categoryID=C.categoryID;



**Query14: Retrieve customer's order date of all customers using LEFT OUTER JOIN.**

SELECT C.customer\_name AS "Customer Name", C.customer\_surname AS "Customer Surname", O.order\_date AS "Order Date"

FROM customer C

LEFT OUTER JOIN customerorder O

ON C.customerID=O.customerID;



**Query15: Retrieve customer's order date of all customers using RIGHT OUTER JOIN.**

SELECT C.customer\_name AS "Customer Name", C.customer\_surname AS "Customer Surname", O.order\_date AS "Order Date"

FROM customer C

RIGHT OUTER JOIN customerorder O

ON C.customerID=O.customerID;



**Query16: Retrieve customer's order date of all customers using FULL OUTER JOIN.**

SELECT C.customer\_name AS "Customer Name", C.customer\_surname AS "Customer Surname", O.order\_date AS "Order Date"

FROM customer C

LEFT OUTER JOIN customerorder O

ON C.customerID=O.customerID

UNION

SELECT C.customer\_name AS "Customer Name", C.customer\_surname AS "Customer Surname", O.order\_date AS "Order Date"

FROM customer C

RIGHT OUTER JOIN customerorder O

ON C.customerID=O.customerID;



**Query17: Creates a stored procedure that retrieves stock status of all products.**

DELIMITER $$

CREATE PROCEDURE GetStockStatus()

BEGIN

SELECT P.product\_name AS "Product Name", S.stock\_status AS "Status", S.numberOf\_stock AS "Number of Stock"

FROM product P, stock S

WHERE P.stockID=S.stockID;

END$$

DELIMITER ;

CALL GetStockStatus();

**Query18: Creates a stored procedure that retrieves “customer information”(name, surname, phone, street, neighborhood, building no, city, post code) according to given “city input“(We used it for those who are in İstanbul).**

DELIMITER $$

CREATE PROCEDURE GetCustomer(IN cityName VARCHAR(30))

BEGIN

SELECT C.customer\_name AS "Customer Name", C.customer\_surname AS "Customer Surname",

C.customer\_phone AS "Phone Number", A.street, A.neighborhood, A.building\_no, A.city, A.post\_code

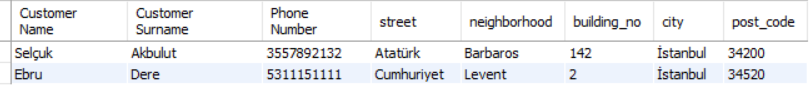
FROM address A, customer C

WHERE A.addressID=C.addressID AND city = cityName;

END $$

DELIMITER ;

CALL GetCustomer("İstanbul");



**Query19: Creates View from Order table as: customer name, customer surname, customer phone, product name, amount, order price, order date.**

CREATE VIEW viewOrder AS

SELECT C.customer\_name AS "Name", C.customer\_surname AS "Surname", C.customer\_phone AS "Phone Number",

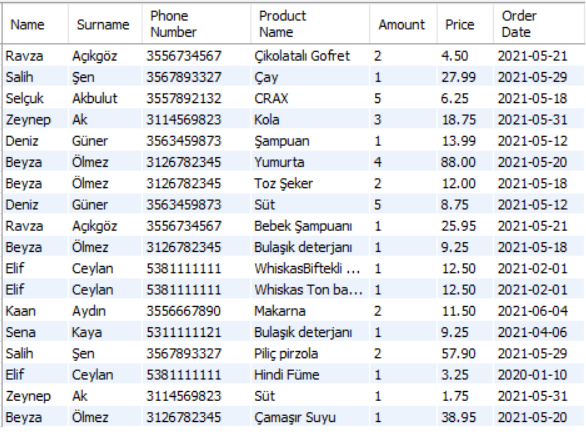
P.product\_name AS "Product Name", OD.amount AS "Amount", OD.price AS "Price", CO.order\_date AS "Order Date"

FROM customer C, product P, orderdetail OD, customerorder CO

WHERE C.customerID=CO.customerID AND CO.orderID=OD.orderID AND OD.productID=P.productID;

SELECT \*

FROM viewOrder;



**Query20: Creates view that displays number of products and total sale price for each category. Also gives different column names to view.**

CREATE VIEW view\_Product\_Info(Category\_name, Number\_of\_product, Total\_sale\_price) AS

SELECT C.category\_name, COUNT(\*), SUM(P.sale\_price)

FROM product P, product\_category PC, category C

WHERE P.productID=PC.productID AND PC.categoryID=C.categoryID

GROUP BY C.category\_name;

SELECT \*

FROM view\_Product\_Info;

