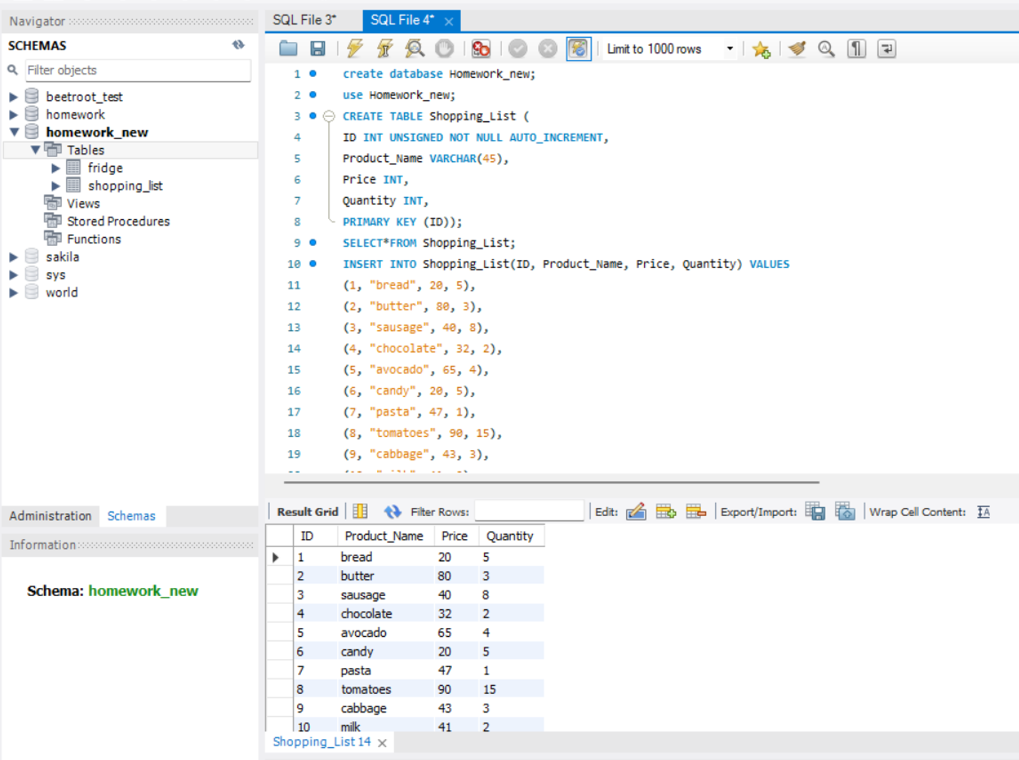
1. Create a database (with an arbitrary name)

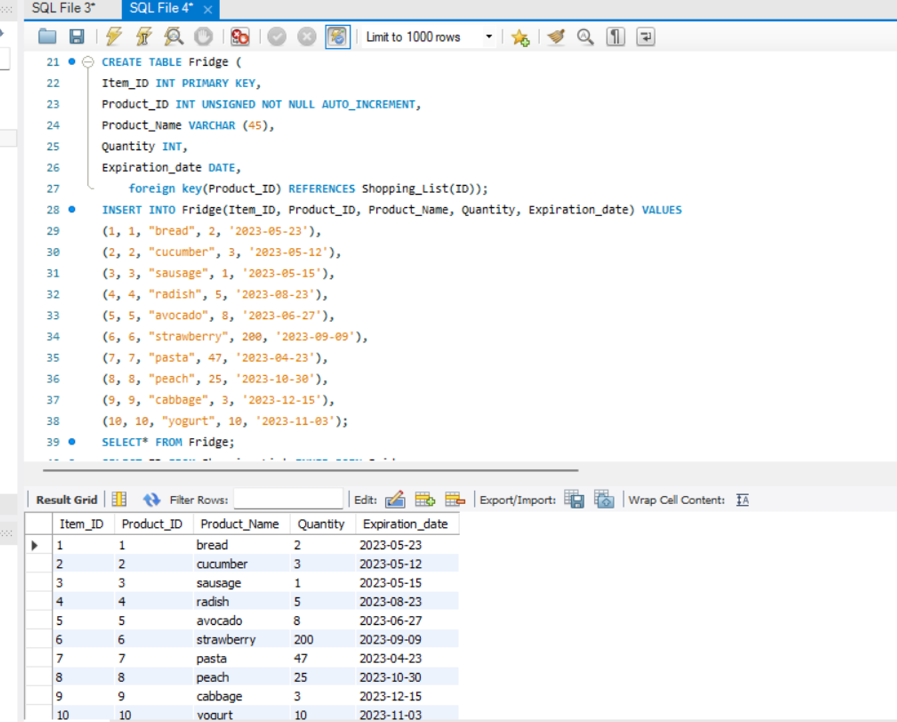
In this database, create a table named Shopping\_List that contains fields with names: ID, Product\_Name, Price, Quantity;

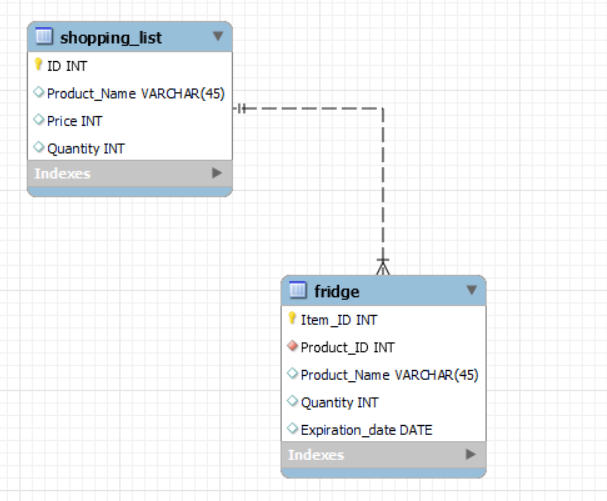
Determine which field will be the primary key.

1. Enter 10 product names into the table and fill in the Price and Quantity fields with arbitrary numbers.
2. Create a query that displays all the fields of the created table.



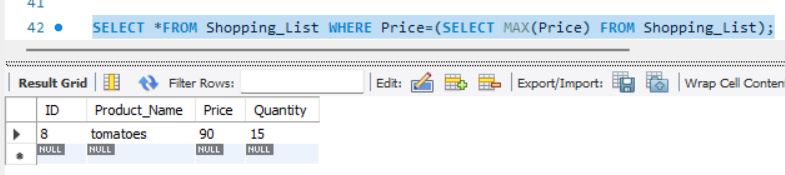
1. Create a product table (named Fridge) with the fields (Item\_ID, Product\_ID, Product\_Name, Quantity, Expiration\_Date) in the already created database.
2. Add 10 product names to the Fridge table, 5 of which will match 5 products from the Shopping\_List table.
3. Complete the table fields with data of your choice.
4. Link this table to the Shopping\_List table using a foreign key.
5. Draw a diagram of the entity relationships\* of the created database, indicating the types of fields, primary keys, and relationships between tables.



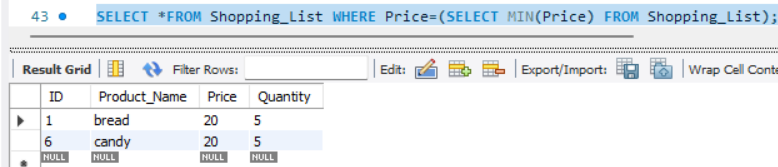


1. Display the information from the Shopping\_List table (created earlier) about the most expensive and cheapest product. Execute the query using the SELECT command.

SELECT \*FROM Shopping\_List WHERE Price=(SELECT MAX(Price) FROM Shopping\_List);

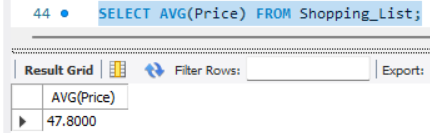


SELECT \*FROM Shopping\_List WHERE Price=(SELECT MIN(Price) FROM Shopping\_List);



1. Calculate the average cost of the products.

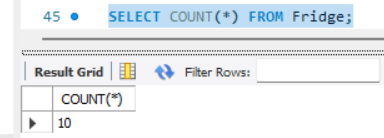
SELECT AVG(Price) FROM Shopping\_List;



1. In the "Fridge" table created for the previous lesson, display information about:

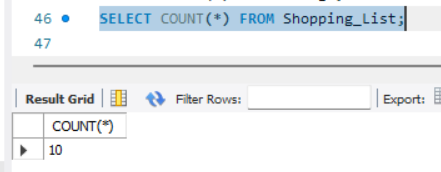
* the number of products in the refrigerator;

SELECT COUNT(\*) FROM Fridge;



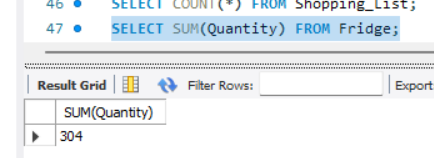
* the number of products in the shopping list (from the Shopping\_List table);

SELECT COUNT(\*) FROM Shopping\_List;



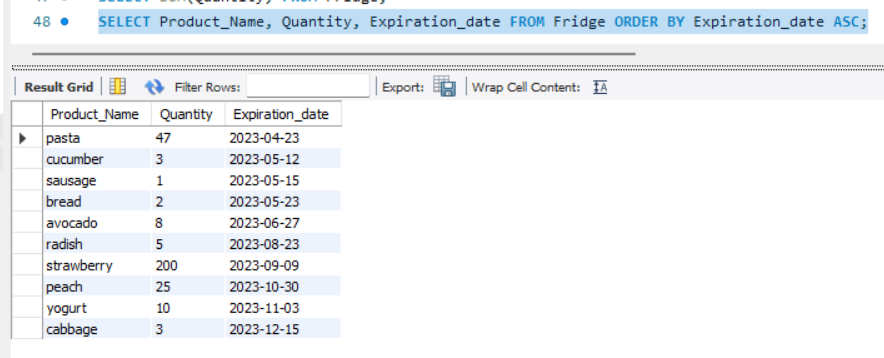
* the total number of all products in the Fridge table.

SELECT SUM(Quantity) FROM Fridge;



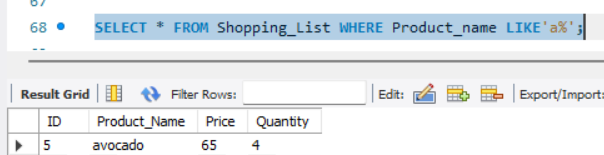
| 1. Display information about the number of products and their names (select columns from the Fridge table). Group the result by expiration date |
| --- |

SELECT Product\_Name, Quantity, Expiration\_date FROM Fridge ORDER BY Expiration\_date ASC;

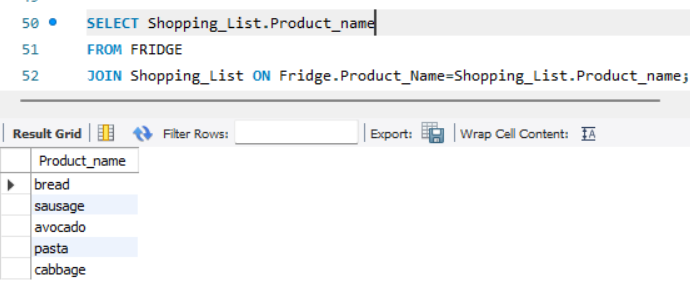


1. Run a query that displays all products that start with the letter A (in the Shopping\_List table)

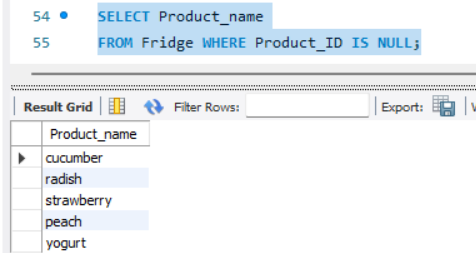
SELECT \* FROM Shopping\_List WHERE Product\_name LIKE'a%';



1. Use the JOIN operator to make a query to find products from the shopping list that are already in the fridge.



1. Display information about products that are not in the fridge and need to be bought (from the Shopping\_List and Fridge tables).



| 1. Create an Order table in an existing database with fields (Order\_ID, Product\_ID, Item\_ID, Date, Quantity):  * Add the products to be ordered (fill it with data according to the columns you have). * Link this table to the previous two. * Add to it all the products that are in the Shopping\_List and Fridge tables.          1. Display information about all products from the Order table that are in the Shopping\_List table and have expired.   SELECT Orders.Product\_ID, Orders.Date, Orders.Quantity,  Shopping\_List.Product\_Name, Fridge.Expiration\_date  FROM Orders  JOIN Shopping\_List ON Orders.Product\_ID=Shopping\_List.ID  JOIN Fridge ON Orders.Item\_ID=Fridge.Item\_ID  WHERE Expiration\_date<'2023-05-17'; |
| --- |