1. A new organic product, "Almond Nuts", is being added to the Arlington Organic Market inventory. The

following steps must be completed:

1. The vendor “Organic Farms” (vId = 201) should be added to the system. The vendor’s address is "123 Greenway Blvd, Dallas, TX, 75001".

2. Insert the new item "Almond Nuts" (iId = 101) into the inventory with a selling price of $12.99 under the

category "Nuts".

3. Link the vendor to this new item.

4. Ensure that store ID = 1 has 50 units of "Almond Nuts" in stock.

**SQL Entries**

1. INSERT INTO vendors (vId, vendor\_name, address)

VALUES (201, 'Organic Farms', '123 Greenway Blvd, Dallas, TX, 75001');

2. INSERT INTO items (iId, item\_name, selling\_price, category)

VALUES (101, 'Almond Nuts', 12.99, 'Nuts');

3. INSERT INTO vendor\_items (vId, iId)

VALUES (201, 101);

4. INSERT INTO store\_inventory (store\_id, iId, stock\_count)

VALUES (1, 101, 50);

A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

A table with numbers and letters

AI-generated content may be incorrect.

Q2. A customer visits store ID = 1 and wants to see a list of available products. The system should retrieve:

 The item name

 The selling price

 The stock count

Write an SQL query to fetch this information.

**SQL Entries:**

SELECT

I.Iname, I.Sprice, SI.Scount

FROM

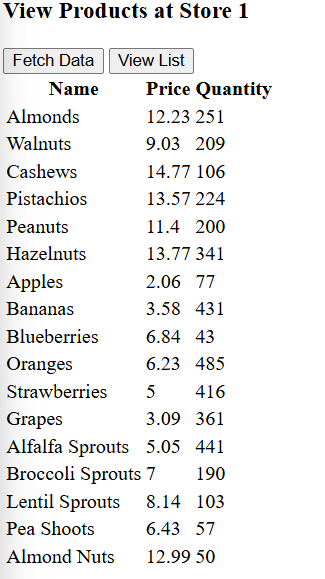
ITEM I

JOIN

STORE\_ITEM SI ON I.iId = SI.iId

WHERE

SI.sId = 1;



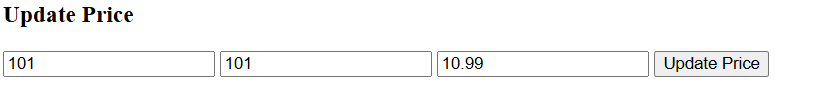
Q3. The store decides to lower the price of "Almond Nuts" from $12.99 to $10.99 due to seasonal promotions.

Write an SQL query to update the item price.

**SQL Queries**UPDATE ITEM

SET Sprice = 10.99

WHERE iId = 101



Q4. The store decides to remove "Almond Nuts" (iId = 101) from the inventory. The following conditions must be met:

1. The item must be removed from store inventory.

2. The vendor-item relationship must be deleted.

3. The item should be deleted from the item table.

4. If the vendor ("Organic Farms") no longer supplies any other items, it should also be removed.

Write an SQL query to perform these actions.

**SQL Queries**

1.

DELETE FROM STORE\_ITEM

WHERE iId = 101;

2.

DELETE FROM VENDOR\_STORE

WHERE vId = 201 AND sId IN (

SELECT sId FROM STORE\_ITEM WHERE iId = 101

);

3.

DELETE FROM ITEM

WHERE iId = 101;

4.

DELETE FROM VENDOR

WHERE vId = 201

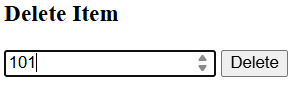
AND NOT EXISTS (

SELECT 1

FROM VENDOR\_ITEMS

WHERE vId = 201

);



QV1 Retrieve the top 3 items that have generated the highest revenue from the ItemSalesSummary view.

Show Item Name, Total Revenue, and sort the results in descending order of revenue.

SELECT Iname, TotalRevenue,

FROM ItemSalesSummary

ORDER BY TotalRevenue DESC

LIMIT 3;

QV2 Retrieve items that have sold more than 50 units in total from the ItemSalesSummary view.

Display Item Name and Total Quantity Sold.

SELECT Iname, TotalQuantitySold AS "Total Quantity Sold"

FROM ItemSalesSummary,

WHERE TotalQuantitySold > 50;

QV3 Retrieve the customer with the highest loyalty score from the TopLoyalCustomers view.

Display Customer Name and Loyalty Score.

SELECT Cname, LoyaltyScore,

FROM TopLoyalCustomers

ORDER BY LoyaltyScore DESC

LIMIT 1;

QV4 Retrieve customers whose loyalty score is between 4 and 5 from the TopLoyalCustomers view.

Show Customer Name and Loyalty Score.

SELECT Cname, LoyaltyScore,

FROM TopLoyalCustomers

WHERE LoyaltyScore BETWEEN 4 AND 5;

QV5 Calculate the total revenue generated across all items in the ItemSalesSummary view.

SELECT SUM(TotalRevenue) AS "Total Revenue"

FROM ItemSalesSummary;