Piri Reis University

Management Information Systems

YBS 221 Data Structures and Databases

Assignment

Market Inventory Management

Salim Can YILDIZ 20200305038

Yağmur Sude YOLCU 20200305036

Instructor: Dr. Volkan USLAN

29 May 2023

# Introduction

We performed on MS SQL in our homework. We chose the Market inventory management principles in the first place, and then we added extensions to the database using MS SQL commands and created a relational archive book. Finally, we used the desired query types to assign to archives.

## Project Purpose

The purpose of this project is to create a database on MS SQL, to view the data of a market inventory in detail and to perform transactions with various queries.

**SUMMARY**

* Firstly, We chose our database topic which is Breakfast Market. The project topic revolves around managing and tracking sales data in a breakfast market context, including distributors, brands, products, and sales information.
* After that, We identified entities and designed Relational Database Schema.
* And then , We wrote SQL Statements to Create the Database.
* We prepared data and inserted it.
* We wrote General (SELECT, FROM, WHERE, ORDER BY, and JOIN.) And Advanced Queries ( GROUP BY, HAVING etc)

**Market Entity**

**metin, ekran görüntüsü, daire, yazı tipi içeren bir resim

Açıklama otomatik olarak oluşturuldu**

**Here, is our market entity. It consists of 4 main parts.**

**Relational Database Schema**

**diyagram, çizgi, ekran görüntüsü, tasarım içeren bir resim

Açıklama otomatik olarak oluşturuldu**

**The relational database schema allows for the organization and management of data related to sales transactions, products, brands, and stores. The relationships between the tables enable data retrieval and analysis by connecting relevant information from different tables based on the defined foreign keys.**

1. **Satislar (Sales):**
   * **SatisID: Unique identifier for each sales transaction.**
   * **SatisTarihi: Date of the sales transaction.**
   * **BayiID: ID of the store where the sales occurred.**
   * **UrunID: ID of the product sold.**
   * **SatisAdedi: Quantity of the product sold in the transaction.**
   * **SatisFiyati: Price of the product in the transaction.**
2. **Markalar (Brands):**
   * **MarkaID: Unique identifier for each brand.**
   * **MarkaAdi: Name of the brand.**
3. **Urunler (Products):**
   * **UrunID: Unique identifier for each product.**
   * **Urunkodu: Code or identifier for the product.**
   * **MarkaID: ID of the brand associated with the product.**
   * **Fiyati: Price of the product.**
4. **Bayiler (Deals):**
   * **BayiID: Unique identifier for each store.**
   * **BayiAdi: Name of the store.**

**While we are working, we used these queries:**

* USE [BREAKFAST MARKET]

SELECT \* FROM Urunler

(This query selects all rows and columns from the table named "Urunler" in the "BREAKFAST MARKET" database. It retrieves all the products information.)

* USE [BREAKFAST MARKET]

SELECT\* FROM SatisBilgileri WHERE BayiAdi = ‘AMASYA’ AND SatisFiyati < 200

ORDER BY SatisFiyati DESC

(This query selects all rows and columns from the table named "SatisBilgileri" in the "BREAKFAST MARKET" database. It retrieves sales information for the 'AMASYA' store where the sales price is less than 200, and the results are ordered in descending order based on the sales price)

* USE [BREAKFAST MARKET]

SELECT SatisTarihi, BayiAdi, UrunKodu, MarkaAdi , SatisAdedi , SatisFiyati FROM Satislar s

LEFT JOIN Bayiler ba on ba.BayiID = s.BayiID

LEFT JOIN Urunler u on u.UrunID = s.UrunID

LEFT JOIN Markalar m on m.MarkaID = s.MarkaID

(This query retrieves specific columns (SatisTarihi, BayiAdi, UrunKodu, MarkaAdi, SatisAdedi, SatisFiyati) from multiple tables (Satislar, Bayiler, Urunler, Markalar) in the "BREAKFAST MARKET" database. It uses LEFT JOIN to combine data from these tables based on matching IDs.)

* SELECT \* FROM Markalar

SELECT \* FROM Markalar ORDER BY MarkaID OFFSET 4 ROWS FETCH NEXT 4 ROWS ONLY;

(This query selects all rows and columns from the table named "Markalar" and orders the results based on the "MarkaID" column. It uses OFFSET to skip the first 4 rows and FETCH NEXT to retrieve the next 4 rows only.)

* USE [BREAKFAST MARKET]

SELECT BayiAdi,

YEAR (SatisTarihi) AS Tarih,

SUM( SatisFiyati) AS SatisTutariToplami,

FROM SatisBilgileri,

GROUP BY BayiAdi , YEAR (SatisTarihi)

HAVİNG BayiAdi in (‘AMASYA’, ‘KIRKLARELİ’,’GİRESUN’)

ORDER BY BayiAdi ASC

(This query retrieves the store name (BayiAdi), sales year (Tarih), and the total sales amount (SatisTutariToplami) from the "SatisBilgileri" table. It groups the results by store name and sales year, filters the results to include only 'AMASYA', 'KIRKLARELİ', and 'GİRESUN' stores, and orders the results in ascending order based on the store name.)

* SELECT BayiID, COUNT (UrunID) AS Toplam\_Urun\_Sayisi

FROM Satislar

GROUP BY BayiID

HAVİNG COUNT (UrunID) > 5

ORDER BY COUNT (UrunID) DESC

(This query selects the store ID (BayiID) and the count of unique product IDs (UrunID) from the "Satislar" table. It groups the results by store ID, filters the results to include only stores with more than 5 unique products sold, and orders the results in descending order based on the count of unique products.)

* USE [BREAKFAST MARKET]

SELECT\*FROM SatisBilgileri

WHERE BayiAdi in (‘GİRESUN’, ‘AMASYA’, ‘DENİZLİ’) AND YEAR(SatisTarihi)= 2010

ORDER BY BayiAdi ASC

(This query selects all rows and columns from the "SatisBilgileri" table in the "BREAKFAST MARKET" database. It filters the results to include only the stores with names 'GİRESUN', 'AMASYA', and 'DENİZLİ', and the sales made in the year 2010. The results are ordered in ascending order based on the store name.)

**These SQL statements are used to create tables in the "BREAKFAST MARKET" database.** **These tables and their columns allow you to store and manage data related to bayiler (stores), markalar (brands), urunler (products), and satislar (sales) in the "BREAKFAST MARKET" database:**

* USE [BREAKFAST MARKET]
* CREATE TABLE Bayiler (

BayiID int NOT NULL PRIMARY KEY,

BayiAdi nvarchar(20) NOT NULL

);

* CREATE TABLE Markalar (

MarkaID int NOT NULL PRIMARY KEY,

MarkaAdi nvarchar(20) NOT NULL

);

* CREATE TABLE Urunler (

UrunID int NOT NULL PRIMARY KEY,

Urunkodu nvarchar(20) NOT NULL,

MarkaID int NULL,

Fiyati numeric(8,2) NOT NULL

);

* CREATE TABLE Satislar (

SatisID int PRIMARY KEY,

SatisTarihi date NOT NULL,

BayiID int NOT NULL,

UrunID int NOT NULL,

SatisFiyati numeric(8,2) NOT NULL,

SatisAdedi int NOT NULL

);

**These SQL statements are INSERT statements to insert data into the tables of the "Bayiler," "Markalar," and "Urunler" in the "BREAKFAST MARKET" database:**

INSERT INTO Bayiler (BayiID, BayiAdi) VALUES

(1, 'AMASYA'),

(2, 'GİRESUN'),

(3, 'ANTALYA'),

(4, 'KIRKLARELİ'),

(5, 'DENİZLİ');

INSERT INTO Markalar (MarkaID, MarkaAdi) VALUES

(1, 'Panar'),

(2, 'Sütaş'),

(3, 'Tahsildaroğlu'),

(4, 'Icim'),

(5, 'Tat'),

(6, 'Polonez'),

(7, 'Name'),

(8, 'Karper'),

(9, 'Bahçıvan'),

(10, 'Yudum');

INSERT INTO Urunler (UrunID, Urunkodu, MarkaID, Fiyati) VALUES

(1, 'Milk', 1, 25),

(2, 'Cheese', 2, 150),

(3, 'WCheese', 3, 89),

(4, 'Yogurt', 4, 20),

(5, 'TomatoPaste', 5, 36),

(6, 'Sausage', 6, 240),

(7, 'Brie', 8, 150),

(8, 'Cheddar', 9, 250),

(9, 'Salami', 7, 160),

(10, 'Ayran', 4, 20),

(11, 'Meat', 6, 350),

(12, 'Butter', 1, 250),

(13, 'Mozarella', 2, 200),

(14, 'Goat Milk', 3, 70);

**The aim of the provided INSERT statement is to insert data into the "Satislar" table in the "BREAKFAST MARKET" database. Each row represents a sales transaction with information such as the sales ID, date of sale, the ID of the store ("BayiID"), the ID of the product ("UrunID"), the sales price ("SatisFiyati"), and the quantity sold ("SatisAdedi"):**

INSERT INTO Satislar (SatisID, SatisTarihi, BayiID, UrunID, SatisFiyati, SatisAdedi) VALUES

(2, '2010-01-01', 1, 1, 79, 25),

(3, '2010-01-01', 2, 2, 82, 150),

(4, '2010-01-01', 1, 3, 81, 89),

(5, '2010-01-01', 3, 1, 27, 25),

(6, '2010-01-01', 3, 4, 18, 20),

(7, '2010-01-01', 4, 5, 19, 36),

(8, '2010-01-01', 2, 6, 96, 240),

(9, '2010-01-01', 4, 7, 28, 160),

(10, '2010-01-01', 5, 8, 89, 150),

………..

**This stage, I add the tables and values ​​in the database I created step by step.**

**Step 1 :** The tables are created as seen in the screenshots below.

**metin, ekran görüntüsü, yazı tipi, sayı, numara içeren bir resim

Açıklama otomatik olarak oluşturuldu**

**Step 2 :** Data were entered into the tables as seen in the screenshots below.

**metin, ekran görüntüsü, yazı tipi, sayı, numara içeren bir resim

Açıklama otomatik olarak oluşturuldu**

Data for the sales table was entered sequentially and proportionally to all other data.**metin, ekran görüntüsü içeren bir resim

Açıklama otomatik olarak oluşturuldumetin içeren bir resim

Açıklama otomatik olarak oluşturuldumetin, ekran görüntüsü, yazı tipi, tipografi içeren bir resim

Açıklama otomatik olarak oluşturuldu**

**We will see the tables formed at this stage.**

**metin, ekran görüntüsü, yazı tipi, sayı, numara içeren bir resim

Açıklama otomatik olarak oluşturuldu**

**metin, ekran görüntüsü, sayı, numara, yazı tipi içeren bir resim

Açıklama otomatik olarak oluşturuldu**

**metin, ekran görüntüsü, sayı, numara, yazı tipi içeren bir resim

Açıklama otomatik olarak oluşturuldu**

**metin, ekran görüntüsü, sayı, numara, yazılım içeren bir resim

Açıklama otomatik olarak oluşturuldu**

**Created View using Left Join**

**metin, yazı tipi, ekran görüntüsü, çizgi içeren bir resim

Açıklama otomatik olarak oluşturuldu**

View is created to specify each column name.

**Queries**

General query of the products table.

**metin, ekran görüntüsü, sayı, numara, yazı tipi içeren bir resim

Açıklama otomatik olarak oluşturuldu**

Query to list sales whose sales price is less than 200 units using the where condition.

metin, ekran görüntüsü, sayı, numara içeren bir resim

Açıklama otomatik olarak oluşturuldu

Query created by connecting data with Join functions.metin, elektronik donanım, ekran görüntüsü, yazılım içeren bir resim

Açıklama otomatik olarak oluşturuldu

Brand query created using offset function.

metin, ekran görüntüsü, yazılım, ekran, görüntüleme içeren bir resim

Açıklama otomatik olarak oluşturuldu

An advanced aggregation query.

metin, elektronik donanım, ekran görüntüsü, ekran, görüntüleme içeren bir resim

Açıklama otomatik olarak oluşturuldu

A clustering query using count.

metin, ekran görüntüsü, ekran, görüntüleme, yazılım içeren bir resim

Açıklama otomatik olarak oluşturuldu

Where condition subquery example.

metin, ekran görüntüsü, sayı, numara, ekran, görüntüleme içeren bir resim

Açıklama otomatik olarak oluşturuldu