Basic Databases – Report01

Wroclaw University of Science and Technology, Date: March 9, 2022

Student:	Email: 245784@student.pwr.edu.pl	Grade
Identifier	245784	
First name	Rahul	
Last name	Vijayvargiya	

This laboratory assignment consists of 1 task. If you cannot solve the task, try to give at least a partial solution or justification for the reason for the lack of a solution.

Task 1

Please analyse the conceptual data model of "Services" (Fig. 1.), which is incomplete, but the classes and the relationships between them may represent a part of the reality under consideration. 'Services' are understood directly

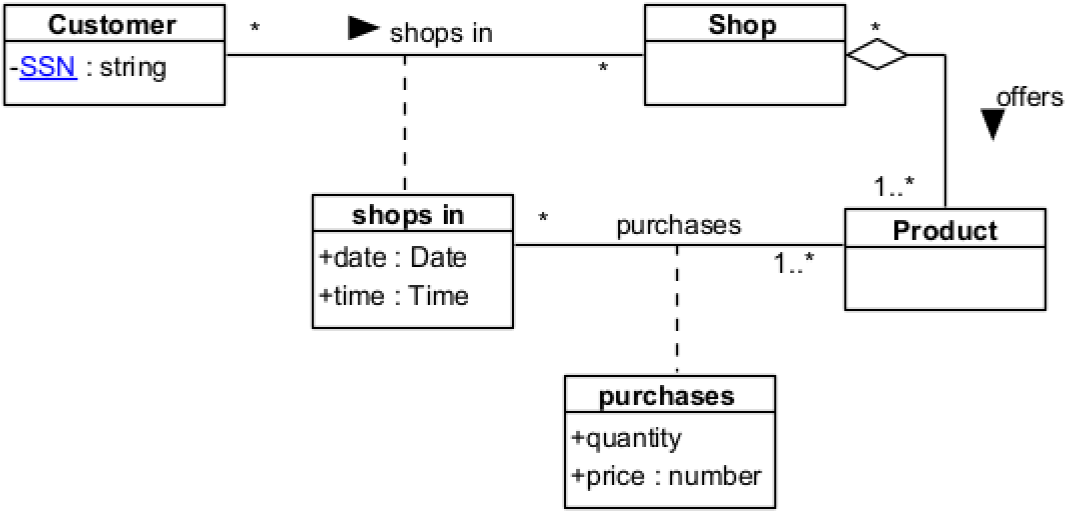


Fig. 1. Conceptual data model for "Services"

Then, complete the following tasks:

1. Verify the data model in the context of a given set of rules and domain constraints. Further, modify the set of rules and constraints (by supplementing their definition) or correct the conceptual data model (justify!) to obtain a valid model.

   Domain rules and constraints – preliminary list:

R01 – Any customer can shop in many stores
R02 – A customer can shop repeatedly in the same store
R03 - Multiple customers can shop at the store
R04 – Each purchase is made by the customer in the store on a specific day and time
R05 – A store must offer at least one product
R06 – The same product (type) can be offered by multiple stores
R07 – Each store can individually propose the price and quantity of the offered product
R08 - …

C01 – Price > 0
C02 - …

2. Provide a revised and complete version of the data model (complete UML class diagram)
3. Create a logical/physical data model as a DDL SQL script (including domain rules and constraints) while trying to comply with the SQL standard (omitting, if possible, native SQL implementation constructs).
4. Create a database in MS SQL 2017 or 2019. The implemented database should be a physical data model of the modeled part of the reality.
5. Test the created database. Enter several records into each table, checking the correctness of the implementation (remember to check both correct data and inconsistent data with the applicable rules – please comment and explain the obtained messages from the DBMS system).

SOLUTIONS:

Use this section to provide your solutions:

1

All the Relationship I Implemented in tables, I have provided code as well and properly implemented from best of knowledge.

2

I do not have sufficient Knowledge to create a proper logical diagram

3

…BEGIN

DROP DATABASE IF EXISTS lab;

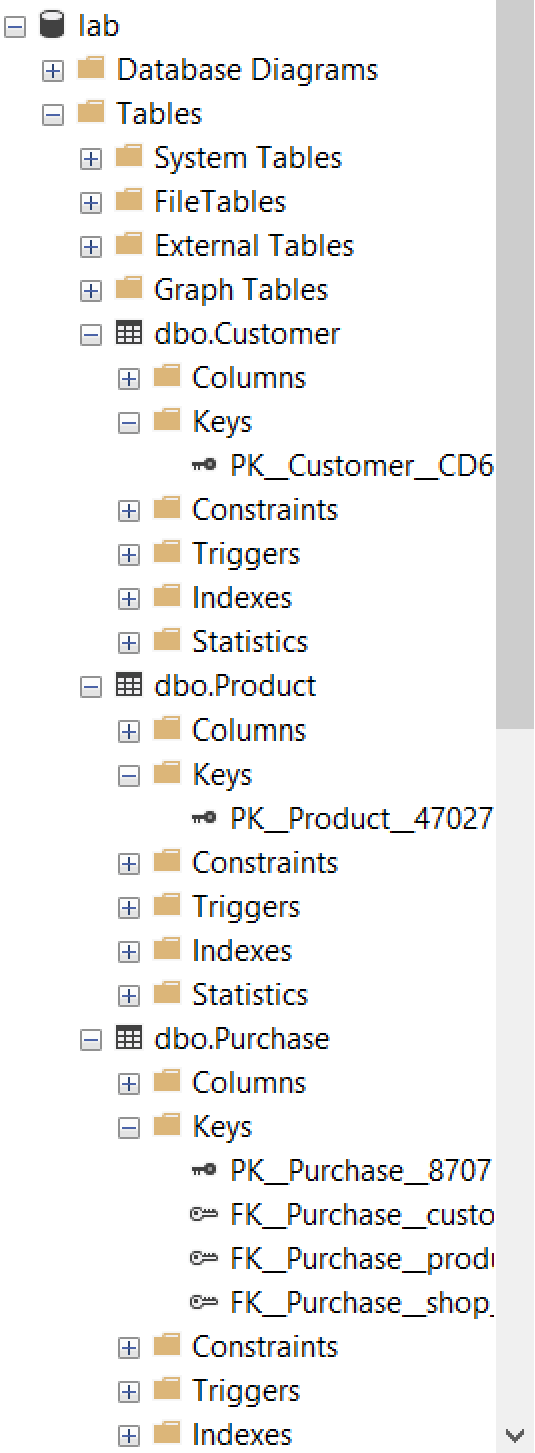
END;

```
GO

BEGIN
IF NOT EXISTS (SELECT * FROM sys.databases WHERE name = 'lab')
CREATE DATABASE lab;
END;
GO

BEGIN
USE lab;
END;
GO

BEGIN
IF NOT EXISTS (SELECT * FROM sysobjects WHERE name='Customer' and xtype='U')
   CREATE TABLE Customer (
      customer_id INT PRIMARY KEY NOT NULL,
                     first_name VARCHAR (255) NOT NULL,
                     last_name VARCHAR (255) NOT NULL,
                     city VARCHAR (50)
   );
END;
GO

BEGIN
IF NOT EXISTS (SELECT * FROM sysobjects WHERE name='Product' and xtype='U')
```

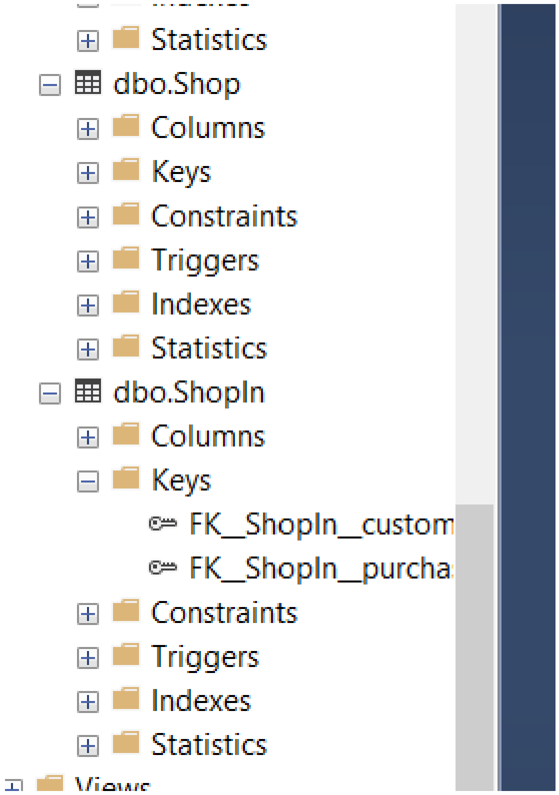
```
   CREATE TABLE Product(
                     product_id INT PRIMARY KEY NOT NULL,
                     product_price INT NOT NULL,
                     product_quantity INT NOT NULL
   );
END;
GO

BEGIN
IF NOT EXISTS (SELECT * FROM sysobjects WHERE name='Shop' and xtype='U')
   CREATE TABLE Shop(
                     shop_id INT PRIMARY KEY NOT NULL,
                     product_id INT NOT NULL,
                     product_offer BIT,
                     FOREIGN KEY (product_id)
      REFERENCES Product(product_id)
   );
END;
GO

BEGIN
IF NOT EXISTS (SELECT * FROM sysobjects WHERE name='Purchase' and xtype='U')
   CREATE TABLE Purchase (
                     customer_id INT NOT NULL,
                     purchase_id INT PRIMARY KEY NOT NULL,
                     product_id INT NOT NULL,
            shop_id INT NOT NULL,
            purchase_date DATE,
            FOREIGN KEY (product_id)
    REFERENCES Product (product_id),
            FOREIGN KEY (shop_id)
    REFERENCES Shop (shop_id),
            FOREIGN KEY (customer_id)
    REFERENCES Customer(Customer_id)
  );
END;
GO

BEGIN
IF NOT EXISTS (SELECT * FROM sysobjects WHERE name='ShopIn' and xtype='U')
  CREATE TABLE ShopIn (
            customer_id INT NOT NULL,
            purchase_id INT NOT NULL,
            shopin_date DATETIME,
            FOREIGN KEY (purchase_id)
    REFERENCES Purchase (purchase_id),
            FOREIGN KEY (customer_id)
    REFERENCES Customer (Customer_id)
  );
END;
GO
```

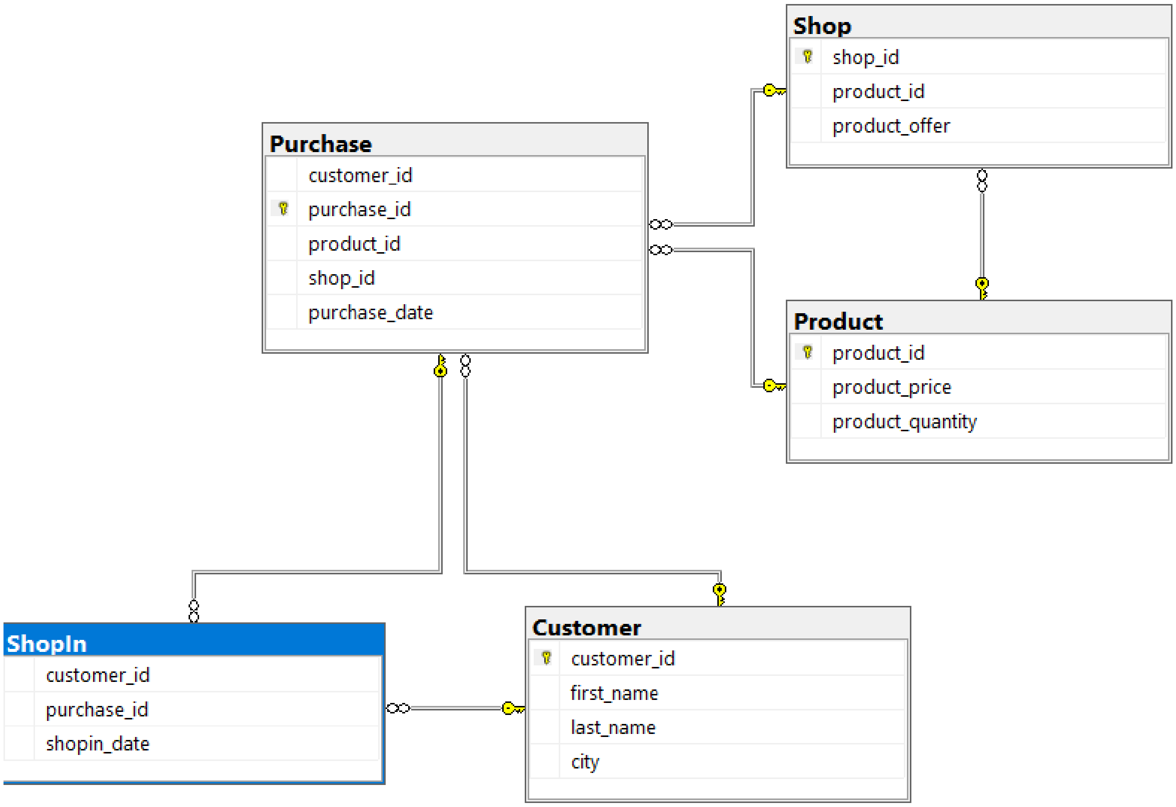
```
INSERT INTO Customer (
        customer_id,
        first_name,
        last_name,
        city
)
VALUES (
        13,
        'rahul',
        'vijay',
        'wroclaw'
);
GO

INSERT INTO Product (
        product_id,
        product_price,
        product_quantity
)
VALUES (
        1,
        10,
        100
```

```
	,
);
GO

INSERT INTO Shop (
	shop_id,
	product_id,
	product_offer
)
VALUES (
	201,
	10,
	'TRUE'
);
GO

INSERT INTO Purchase (
	customer_id,
	purchase_id,
	product_id,
	shop_id,
	purchase_date
)
VALUES (
	101,
	1001,
	105,
        13,
        '10/10/20'
);
GO

INSERT INTO ShopIn (
        customer_id,
        purchase_id,
        shopin_date
)
VALUES (
        123,
        145,
        '10/12/2021'
);
GO
```



- ⊟ lab
  - ⊞ Database Diagrams
  - ⊟ Tables
    - ⊞ System Tables
    - ⊞ FileTables
    - ⊞ External Tables
    - ⊞ Graph Tables
    - ⊟ dbo.Customer
      - ⊞ Columns
      - ⊟ Keys
        - PK__Customer__CD6
      - ⊞ Constraints
      - ⊞ Triggers
      - ⊞ Indexes
      - ⊞ Statistics
    - ⊟ dbo.Product
      - ⊞ Columns
      - ⊟ Keys
        - PK__Product__47027
      - ⊞ Constraints
      - ⊞ Triggers
      - ⊞ Indexes
      - ⊞ Statistics
    - ⊟ dbo.Purchase
      - ⊞ Columns
      - ⊟ Keys
        - PK__Purchase__8707
        - FK__Purchase__custo
        - FK__Purchase__produ
        - FK__Purchase__shop_
      - ⊞ Constraints
      - ⊞ Triggers
      - ⊞ Indexes


Statistics
dbo.Shop
Columns
Keys
Constraints
Triggers
Indexes
Statistics
dbo.ShopIn
Columns
Keys
FK__ShopIn__custom
FK__ShopIn__purcha
Constraints
Triggers
Indexes
Statistics
Views



Shop

shop_id
product_id
product_offer

Purchase

customer_id
purchase_id
product_id
shop_id
purchase_date

Product

product_id
product_price
product_quantity

ShopIn

customer_id
purchase_id
shopin_date

Customer

customer_id
first_name
last_name
city

CONCLUSIONS:

Use this section to provide your conclusions:

1. A well-designed database ensures the security of storage

correct data and protects us from pasting and entering incorrect data.

2. for creation of database we should a proper E-R Diagram, So we can have proper Idea before Implementing a database, how and what we was as a End product look like and what sort of table and functionalities it should consist, it could have been better with proper logical diagram, but due to not posses proper knowledge or Creation of ER Diagram, I couldn’t implement

3. We also check the correctness of the data in several different ways, by using the options

NOT NULL, CONSTRAINTS and IF EXISTS.

REMARKS:

- *A report without final conclusions will not be checked and results in a negative score!*
- *The report file should be named Rep01DW-StudentID 2022, please use the PDF format*
- *You should use MS SQL SERVER 2017 or 2019*
- *A conceptual model of data should be prepared using a dedicated tool supporting the UML language, e.g. Visual Paradigm, StarUML.*

Please do not forget that the conclusions are a summary of the problems considered and proposed solutions!