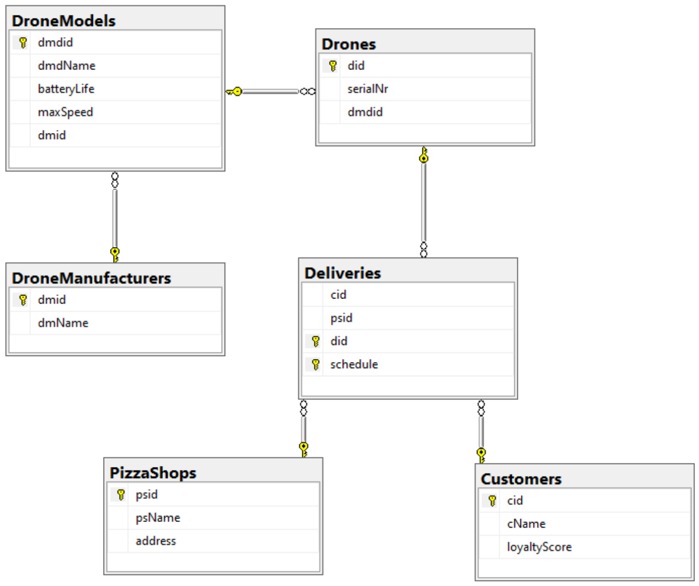
```
II. a)

USE [pizzaDb]
GO

CREATE TABLE DroneManufacturers(
	dmid INT PRIMARY KEY IDENTITY(1,1),
	dmName VARCHAR(30)
)
CREATE TABLE DroneModels(
	dmdid INT PRIMARY KEY IDENTITY(1,1),
	dmdName VARCHAR(30),
	batteryLife INT,
	maxSpeed INT,
	dmid INT REFERENCES DroneManufacturers(dmid)
)
CREATE TABLE Drones(
	did INT PRIMARY KEY IDENTITY(1,1),
	-- serial number can contain letters also
	serialNr VARCHAR(30),
	dmdid INT REFERENCES DroneModels(dmdid)
)
CREATE TABLE PizzaShops(
	psid INT PRIMARY KEY IDENTITY(1,1),
	psName VARCHAR(30),
	address VARCHAR(30)
)
CREATE TABLE Customers(
	cid INT PRIMARY KEY IDENTITY(1,1),
	cName VARCHAR(30),
	-- loyalty score is stored as an integer ( maybe 1$ spent could be 1 point )
	loyaltyScore INT
)
CREATE TABLE Deliveries(
	cid INT REFERENCES Customers(cid),
	psid INT REFERENCES PizzaShops(psid),
	did INT REFERENCES Drones(did),
	schedule DATETIME,
	-- a drone can only be at one place at one given time
	PRIMARY KEY(did,schedule)
)
```



II. b)

```
using System;
using System.Data;
using System.Windows.Forms;
using System.Data.SqlClient;

namespace WindowsFormsApp1
{
    public partial class Form1 : Form
    {
        // To establish the connection
        SqlConnection dbConn;

        // To bring the data
        SqlDataAdapter daManufacturers, daModels;

        // To fill with data from the database
        DataSet ds;

        // To generate insert, update, delete commands
        SqlCommandBuilder cbModels;

        BindingSource bsManufacturers, bsModels;

        private void Button1_Click(object sender, EventArgs e)
        {
            // Send our changes to the database when clicked
            daModels.Update(ds, "Models");
        }

        public Form1()
        {
            InitializeComponent();
        }

        private void Form1_Load(object sender, EventArgs e)
        {
            // Instantiate binding sources
            bsManufacturers = new BindingSource();
            bsModels = new BindingSource();

            // Bind the dataGridView to the binding sources
            dgvManufacturers.DataSource = bsManufacturers;
            dgvModels.DataSource = bsModels;

            // Instantiate the connection object
            // Integrated security is true to specify that we're using the current windows account
            // credentials
            dbConn = new SqlConnection("Data Source = DESKTOP-A70AQJO\\SQLEXPRESS;" +
                " Initial Catalog = pizzaDb; Integrated Security = true");

            // Instantiatae the dataSet and dataAdapters
            ds = new DataSet();

            daManufacturers = new SqlDataAdapter("SELECT * FROM DroneManufacturers", dbConn);
            daModels = new SqlDataAdapter("SELECT * FROM DroneModels", dbConn);

            cbModels = new SqlCommandBuilder(daModels);

            // We'll have 2 tables in the data adapters called Manufacturers and Models containing all
            // the rows from the db for the respective table
            daManufacturers.Fill(ds, "Manufacturers");
            daModels.Fill(ds, "Models");

            // We will represent the relation using a Relation object
            DataRelation dr = new DataRelation("ManufacturersModels",
                ds.Tables["Manufacturers"].Columns["dmid"], // the parent column
                ds.Tables["Models"].Columns["dmid"]);   // the child column

            ds.Relations.Add(dr); // We add the relation to the dataSet

            // We specify the binding details for the binding sources
            bsManufacturers.DataSource = ds;
            bsManufacturers.DataMember = "Manufacturers";

            bsModels.DataSource = bsManufacturers;
            // We specify the relation by name in order to filter
            bsModels.DataMember = "ManufacturersModels";
        }
    }
}
```

II. c)

```
-- T1
-- Just update a field ( first i initialize it with abc then i update it in the transaction
-- in order to see the change)
USE [pizzaDb]
GO

UPDATE DroneManufacturers SET dmName='ABC' WHERE dmid=1
BEGIN TRAN
WAITFOR DELAY '00:00:07'
UPDATE DroneManufacturers SET dmName='Pro' WHERE dmid=1
COMMIT TRAN
```

```
-- T2: problem
-- If one transaction reads a database row without applying
-- a shared lock on the fetched record, then a concurrent
-- transaction might change this row before the first transaction has ended.

-- Here we would see a different result in the two selects
USE [pizzaDb]
GO

SET TRANSACTION ISOLATION LEVEL READ COMMITTED
BEGIN TRAN
SELECT * FROM DroneManufacturers
WAITFOR DELAY '00:00:15'
SELECT * FROM DroneManufacturers
COMMIT TRAN

------------------------------------------------------------------------

-- T2:  solution
-- solution: iso level -> repeatable read ;
-- whereas here we would have only the final result in both selects
USE [pizzaDb]
GO

SET TRANSACTION ISOLATION LEVEL REPEATABLE READ
BEGIN TRAN
SELECT * FROM DroneManufacturers
WAITFOR DELAY '00:00:15'
SELECT * FROM DroneManufacturers
COMMIT TRAN
```