**Project Assignment**

NAME: **PAVANI BADAM**

PROJECT: **TRANSACTION MANAGEMENT SYSTEM**

**Instructions:**

Answer the following questions and paste the picture or the code below each question. When done, submit this document back to Canvas.

1. Create an ER diagram for this database using the ERDPlus software. Paste the picture below.

A picture containing drawing, diagram, sketch, line

Description automatically generated

1. Create the schema for this database with proper field types and sizes. Paste the picture below.

A picture containing text, diagram, plan, line

Description automatically generated

1. Generate the SQL code using the schema in Question2. Paste the code below.

CREATE TABLE Customer

(

lastName VARCHAR(40) NOT NULL,

customerId INT NOT NULL,

firstName VARCHAR(40) NOT NULL,

city VARCHAR(20) NOT NULL,

zip\_code CHAR(5) NOT NULL,

state VARCHAR(30) NOT NULL,

PRIMARY KEY (customerId)

);

CREATE TABLE Transaction

(

transactionID INT NOT NULL,

customerId INT NOT NULL,

trans\_type VARCHAR(10) NOT NULL,

transactionDate DATE NOT NULL,

trans\_Amount FLOAT NOT NULL,

accountID VARCHAR(10) NOT NULL,

PRIMARY KEY (transactionID)

);

CREATE TABLE Account

(

acct\_type VARCHAR(20) NOT NULL,

customerID INT NOT NULL,

accountID VARCHAR(10) NOT NULL,

bankId VARCHAR(10) NOT NULL,

total\_amount FLOAT NOT NULL,

PRIMARY KEY (accountID)

);

CREATE TABLE Bank

(

bankName VARCHAR(40) NOT NULL,

bankId VARCHAR(10) NOT NULL,

Manager VARCHAR(30) NOT NULL,

PRIMARY KEY (bankId)

);

CREATE TABLE has

(

customerId INT NOT NULL,

accountID VARCHAR(10) NOT NULL,

PRIMARY KEY (customerId, accountID),

FOREIGN KEY (customerId) REFERENCES Customer(customerId),

FOREIGN KEY (accountID) REFERENCES Account(accountID)

);

CREATE TABLE makes

(

customerId INT NOT NULL,

transactionID INT NOT NULL,

PRIMARY KEY (customerId, transactionID),

FOREIGN KEY (customerId) REFERENCES Customer(customerId),

FOREIGN KEY (transactionID) REFERENCES Transaction(transactionID)

);

CREATE TABLE in

(

accountID VARCHAR(10) NOT NULL,

bankId VARCHAR(10) NOT NULL,

PRIMARY KEY (accountID, bankId),

FOREIGN KEY (accountID) REFERENCES Account(accountID),

FOREIGN KEY (bankId) REFERENCES Bank(bankId)

);

CREATE TABLE updates

(

transactionID INT NOT NULL,

accountID VARCHAR(10) NOT NULL,

PRIMARY KEY (transactionID, accountID),

FOREIGN KEY (transactionID) REFERENCES Transaction(transactionID),

FOREIGN KEY (accountID) REFERENCES Account(accountID)

);

1. Using the same MariaDB account provided to you in the midterm exam, run your SQL code in your DBstudentXX database to create the tables. Put a prefix (such as prj\_) in front of your table names so that they stand out from other tables in the database. Your tables in the server will be available to me for grading.
2. Now enter at least 6 imaginary records in each table for testing purposes using DBeaver. Make sure your name is entered in one of the tables, such as customer or employee tables.

A screenshot of a computer

Description automatically generated with medium confidence

1. Using two or three tables inner joined, write a query that includes aggregation (e.g. AVG, SUM, etc.) with GROUP BY. Paste the code below.

# Write a query to find total transaction amount by each customer with their firstname, lastname, customerId and accountID

**SELECT** pc.customerId, pc.firstName , pc.lastName ,

**sum**(pt.trans\_Amount) **as** transaction\_amount ,

pa.accountID

**from** prj\_Customer pc

**inner** **join** prj\_Transaction pt **on** pc.customerId = pt.customerId

**inner** **join** prj\_Account pa **on** pt.accountID = pa.accountID

**group** **by** pc.customerId ;

A screenshot of a computer

Description automatically generated with medium confidence

1. Now answer the same question in Question6 but this time use R code. Paste the code below.

library(Rcpp)

library(RMariaDB)

con1 <- dbConnect(RMariaDB::MariaDB(),

host = "24.45.130.70",

port = 6612,

user = "student06",

password = "banana",

dbname="DBstudent06"

)

query =

"

SELECT pc.customerId, pc.firstName , pc.lastName ,

sum(pt.trans\_Amount) as transaction\_amount ,

pa.accountID

from prj\_Customer pc

inner join prj\_Transaction pt on pc.customerId = pt.customerId

inner join prj\_Account pa on pt.accountID = pa.accountID

group by pc.customerId

"

res = dbGetQuery(con1, query)

View(res)

dbDisconnect(con1)

1. In KNIME, connect the same tables in Question6 and inner join them. Finish the workflow with a statistics node showing basic descriptive statistics. Make sure your name shows somewhere in the workflow. Copy and paste a picture of the entire workflow below.

