Lab Assignment 2

UCS 310 DBMS

Name = Tanmay Garg

Batch = 3ENC1

Roll No. = 101915001

QUESTION 1: create customer table based on given information

CUSTOMER Table Example

CUSTOMER_NAME Unique customer name associated with an account. VARCHAR(15). PRIMARY

KEY.

CUSTOMER_STREET Name of the street in which customer lives. VARCHAR(15). NULL allowed.

CUSTOMER_CITY Name of the city in which customer lives. VARCHAR (15).

```
create table Customer(
Customer_Name varchar(15) primary key,
Customer_Street varchar(15) not null,
Customer_City varchar(15) not null,
);
```

Customer_Name	Customer_Street	Customer_City

QUESTION 2: create branch table based on given information

BRANCH Table

BRANCH_NAME Unique branch name for different branches of the bank. VARCHAR(15). PRIMARY KEY.

BRANCH_CITY Name of the city in which branch is located. VARCHAR(15).

ASSETS The bank monitors the assets of each branch. INTEGER(8).

```
create table Branch(
Branch_Name varchar(15) primary key,
Branch_City varchar(15) not null, Assets int
not null
);
```

Branch_Name	Branch_City	Assets

QUESTION 3: create account table based on given information

ACCOUNT Table

ACCOUNT_NUMBER Unique account number in a bank branch. INTEGER(8). PRIMARY KEY.

BRANCH_NAME Brach name in which the account is opened. VARCHAR(15), FOREIGN KEY

to BRANCH table.

BALANCE Keeps track of the balance left in the account. INTEGER(8).

DATE The most recent date on which the account was accessed by the customer.

DATE. Stores year, month, and day values (yyyy-mm-dd)

```
create table Account(
Account_Number int primary key,
Branch_Name varchar(15) foreign key references Branch(Branch_Name) not null, Balance int
not null,
Dates date not null
);
```

Account_Number	Branch_Name	Balance	Dates

QUESTION 4: create loan table based on given information

LOAN Table

LOAN_NUMBER A loan originates at a particular branch and can be held by one or more

customers.

A loan is identified by a unique loan number. INTEGER(8). PRIMARY KEY.

BRANCH_NAME Brach name in which the account is opened. VARCHAR(15), FOREIGN KEY to

BRANCH table.

AMOUNT Keeps track of the loan amount taken by a customer. INTEGER(8).

```
create table Loan( Loan_Number
int primary key,
Branch_Name varchar(15) foreign key references Branch(Branch_Name) not null, Amount int
not null
);
```

Loan_Number	Branch_Name	Amount

QUESTION 5: create depositor table based on given information

DEPOSITOR Table

CUSTOMER_NAME Unique customer name associated with an account. VARCHAR(15). FOREIGN KEY to CUSTOMER table.

ACCOUNT_NUMBER Unique account number in a bank branch. INTEGER(8).FOREIGN KEY to ACCOUNT table.

```
create table Depositor(
Customer_Name varchar(15) unique foreign key references Customer(Customer_Name) not null,
Account_Number int foreign key references Account(Account_Number) not null
);
```

Customer_Name	Account_Number

QUESTION 6: create borrower table based on given information

BORROWER Table

CUSTOMER_NAME Unique customer name associated with an account. VARCHAR(15). FOREIGN KEY to CUSTOMER table.

LOAN_NUMBER A loan is identified by a unique loan number. INTEGER(8). FOREIGN KEY to LOAN table.

```
create table Borrower(
Customer_Name varchar(15) unique foreign key references Customer(Customer_Name) not null,
Loan_Number int foreign key references Loan(Loan_Number) not null
);
```

Customer_Name	Loan_Number

QUESTION 7: create persons table based on given information

Persons Table having columns

- ID integer
- FIRST_NAME varchar(200)
- LAST_NAME varchar(200)
- ADDRESS varchar(200)
- CITY varchar(200)

```
create table Persons( Id
int not null,
First_Name varchar(200) not null,
Last_Name varchar(200) not null,
Addresses varchar(200) ,
City varchar(200) not null
);
```

Id	First_Name	Last_Name	Addresses	City

QUESTION 8: Modify persons table by adding one new column named age of type integer

```
alter table Persons add
Age int;
```

Id	First_Name	Last_Name	Addresses	City	Age

QUESTION 9: Modify persons table by deleting one existing column named city

```
alter table Persons
drop column City;
```

Id	First_Name	Last_Name	Addresses	Age

QUESTION 10: Modify persons table by adding following constraints

ID→ Primary Key

FIRST_NAME and LAST_NAME→ Unique

ADDRESS and AGE → NOT NULL

```
alter table Persons add
primary key(Id); alter table
Persons
add unique(First_Name);
alter table Persons
add unique(Last_Name);
alter table Persons
alter column Addresses varchar(20) not null; alter table
Persons
alter column Age varchar(20) not null;
```

Id	First_Name	Last_Name	Addresses	Age

QUESTION 11: Modify persons table by adding following constraints

ADDRESS→ default value: Patiala

AGE→ must be greater than 20

```
alter table Employee
add check(Age>=20);
```

Id	First_Name	Last_Name	Addresses	Age