Assignment 1

Contents

[1. Create following banking database schema and insert appropriate data: 2](#_Toc121730653)

[a) Design all tables with appropriate keys and constraints. 2](#_Toc121730654)

[b) Find the names of all branches located in “Chicago”. 4](#_Toc121730655)

[c) Add new customer in branch “Atlanta”, Consider appropriate values for other attributes. 4](#_Toc121730656)

[d) Find all the account numbers having balance greater than $1000 4](#_Toc121730657)

[e) Create a view to retrieve all the customers in “Atlanta” city with balance less than balance of customer “John”. (subquery) 5](#_Toc121730658)

[f) Modify the relations so that the default branch city will be “Washington”. 5](#_Toc121730659)

[g) Get the list of all customers only if the asset value for that branch is $35000 (EXISTS clause). 5](#_Toc121730660)

[2. Create Insurance database and insert appropriate data: 6](#_Toc121730661)

[a) Design all tables with appropriate keys and constraints. 6](#_Toc121730662)

[b) Add all appropriate foreign keys. 6](#_Toc121730663)

[c) Add a new accident to the database; assume any values for required attributes 7](#_Toc121730664)

[d) Find report numbers for all the accidents in which the cars belonging to “John Smith” were involved. ( subquery) 8](#_Toc121730665)

[e) Delete the Mazda belonging to “John Smith”. 8](#_Toc121730666)

[f) Create a view to get date and locations of all accidents. 8](#_Toc121730667)

[g) Find the name and addresses of owners of cars involved in accidents for which damage amount is greater than $2000. 8](#_Toc121730668)

[h) Add car prices in the appropriate relation. 9](#_Toc121730669)

[i) Create view to have the information of persons and their cars . 9](#_Toc121730670)

[j) Rename the name of the table person to owners. 9](#_Toc121730671)

[k) Change the type of attribute damage amount from int to numeric. 9](#_Toc121730672)

[l) Delete all values in accident table but keep the relation in schema. 10](#_Toc121730673)

[m) Modify the accident table to increase the report number by one for each new insertion. 10](#_Toc121730674)

[3. Write a query to calculate 123-56\*3+23. 11](#_Toc121730675)

# 1. Create following banking database schema and insert appropriate data:

branch (branch name, branch city, asset\_value)

customer (customer name, customer street, customer city)

account (account number, customer name, branch name, balance)

## a) Design all tables with appropriate keys and constraints.

Solution:

-- Create new database called 'banking' and use it

CREATE DATABASE banking;

USE banking;

-- a) Design all tables with appropriate keys and constraints.

-- Create table 'branch' with 'branch\_name' as primary key

CREATE TABLE branch(branch\_name VARCHAR(50) PRIMARY KEY,

branch\_city VARCHAR(50),

asset\_value BIGINT NOT NULL);

-- Create table for 'customer' with 'customer\_name' as primary key, assiming no two or more customers can have same value

CREATE TABLE customer(customer\_name VARCHAR(50) PRIMARY KEY,

customer\_street VARCHAR(50),

customer\_city VARCHAR(50) NOT NULL);

-- Create 'account' table with account\_number as primary key,

-- customer name is referenced from customer table, branch name is referenced from branch table as foreign key,

-- default balance for new accounts is 0

CREATE TABLE account(account\_number INT PRIMARY KEY,

customer\_name VARCHAR(50) NOT NULL,

branch\_name VARCHAR(50),

balance DOUBLE DEFAULT 0,

FOREIGN KEY fk\_cust\_name(customer\_name) REFERENCES customer(customer\_name), FOREIGN KEY fk\_branch\_nmae(branch\_name) REFERENCES branch(branch\_name));

-- Insert data into the tables

INSERT INTO branch VALUES

('Chi-1', 'Chicago', 10000),

('Chi-2', 'Chicago', 20000),

('Atl-1', 'Atlanta', 25000),

('Atl-2', 'Atlanta', 23000),

('Was-1', 'Washington', 35000),

('Was-2', 'Washington', 36000),

('Nyc-1', 'New York', 45000),

('Nyc-2', 'New York', 46000);

-- Insert data into the tables

INSERT INTO customer VALUES

('John', 'Streat 1', 'New York'),

('Nick', 'Atl street 2', 'Atlanta'),

('Harry', 'First street', 'Chicago'),

('Ron', 'North street', 'Washington'),

('David', 'Second street', 'Chicago'),

('Natasha', 'Green lane', 'Atlanta');

-- Insert data into account table

INSERT INTO account VALUES

(1, 'John', 'Nyc-1', 1400),

(2, 'Nick', 'Atl-2', 1800),

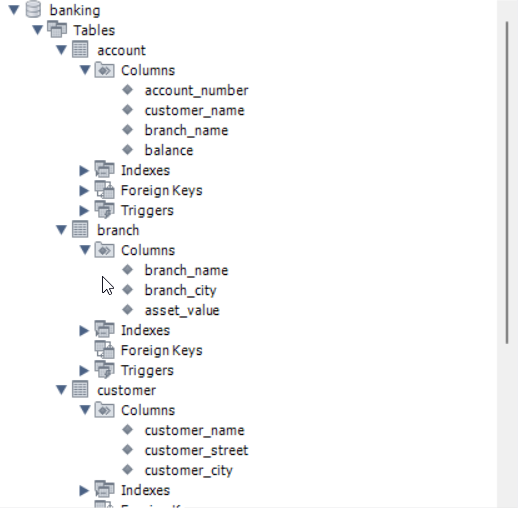
(3, 'Harry', 'Chi-1', 1000),

(4, 'Ron', 'Was-1', 500),

(5, 'David', 'Chi-2', 1500),

(6, 'Natasha', 'Atl-1', 1100);

Output:



Inserted data:

|  |  |  |
| --- | --- | --- |
| Branch | account | |
|  |  | |
| Customer | |
|  | |

## b) Find the names of all branches located in “Chicago”.

Solution:

SELECT branch\_name FROM branch WHERE branch\_city='Chicago';

Output:

|  |
| --- |
| branch |
|  |

## c) Add new customer in branch “Atlanta”, Consider appropriate values for other attributes.

Solution:

INSERT INTO customer VALUES ('Tony', 'Silver street', 'Atlanta');

INSERT INTO account VALUES (7, 'Tony', 'Atl-2', 1200);

Output:

|  |
| --- |
| account |
|  |

## d) Find all the account numbers having balance greater than $1000

Solution:

SELECT \* FROM account WHERE balance > 1000;

Output:

|  |
| --- |
| account |
|  |

## e) Create a view to retrieve all the customers in “Atlanta” city with balance less than balance of customer “John”. (subquery)

Solution:

CREATE VIEW bal\_less\_than\_john\_atlanta AS

SELECT \* FROM account WHERE

balance < (SELECT balance FROM account WHERE customer\_name='John') AND

branch\_name IN (SELECT branch\_name FROM branch WHERE branch\_city='Atlanta');

SELECT \* FROM bal\_less\_than\_john\_atlanta;

Output:

|  |
| --- |
| bal\_less\_than\_john\_atlanta |
|  |

## f) Modify the relations so that the default branch city will be “Washington”.

Solution:

ALTER TABLE branch ALTER branch\_city SET DEFAULT 'Washington';

## g) Get the list of all customers only if the asset value for that branch is $35000 (EXISTS clause).

Solution:

SELECT \* FROM customer WHERE EXISTS (SELECT asset\_value FROM branch WHERE asset\_value = 35000)

Output:

|  |
| --- |
| customer |
|  |

# 2. Create Insurance database and insert appropriate data:

person (driver id, name, address)

car (license, model, year)

accident (report number, location)

owns (driver id, license)

participated (report number, license, driver id, damage amount)

## a) Design all tables with appropriate keys and constraints.

## b) Add all appropriate foreign keys.

Solution:

-- Create database 'insurance' and use it

CREATE DATABASE insurance;

USE insurance;

-- Create tables for person, car, accident, owns and participated

CREATE TABLE person(driver\_id INT PRIMARY KEY, name VARCHAR(50) NOT NULL, address VARCHAR(100) NOT NULL);

CREATE TABLE car (license VARCHAR(10) PRIMARY KEY, model VARCHAR(50) NOT NULL, manufacture\_year YEAR);

CREATE TABLE owns (driver\_id INT NOT NULL, license VARCHAR(10) NOT NULL UNIQUE,

FOREIGN KEY fk\_own\_driver\_id(driver\_id) REFERENCES person(driver\_id),

FOREIGN KEY fk\_own\_license(license) REFERENCES car(license));

CREATE TABLE participated (report\_number INT PRIMARY KEY, license VARCHAR(10) NOT NULL,

driver\_id INT NOT NULL, damage\_amount INT NOT NULL DEFAULT 0,

FOREIGN KEY fk\_part\_driver\_id(driver\_id) REFERENCES person(driver\_id),

FOREIGN KEY fk\_part\_license(license) REFERENCES car(license));

CREATE TABLE accident (report\_number INT PRIMARY KEY, location VARCHAR(50),

FOREIGN KEY fk\_acc\_report\_number(report\_number) REFERENCES participated(report\_number));

-- Insert data

INSERT INTO person VALUES

(1, 'Raj', 'Indore'),

(2, 'Ajay', 'Nagpur'),

(3, 'Bala', 'Pune'),

(4, 'John Smith', 'Agra');

INSERT INTO car VALUES

('A1B2C1', 'Maruti', 2010),

('A1B2C2', 'Honda', 2011),

('A1B2C3', 'Mazda', 2012),

('A1B2C4', 'Hyundai', 2020),

('A2B3C4', 'Honda', 2021),

('M1M2M3', 'Mazda', 2019);

INSERT INTO owns VALUES

(1, 'A1B2C1'),

(2, 'A1B2C2'),

(3, 'A1B2C3'),

(2, 'A1B2C4'),

(4, 'A2B3C4'),

(4, 'M1M2M3');

INSERT INTO participated VALUES

(1, 'A2B3C4', 2, 1500),

(2, 'A1B2C2', 1, 1000),

(3, 'A2B3C4', 2, 2100);

INSERT INTO accident VALUES

(1, 'Noida'),

(2, 'Delhi'),

(3, 'Patna');

Output:

|  |  |
| --- | --- |
|  |  |

Inserted data:

|  |  |  |
| --- | --- | --- |
| person | car | owns |
|  |  |  |
| participated | accident |
|  |  |

## c) Add a new accident to the database; assume any values for required attributes

Solution:

-- c) Add a new accident to the database; assume any values for required attributes

INSERT INTO participated VALUES(4, 'A1B2C1', 2, 20);

INSERT INTO accident VALUES (4, 'Delhi');

Output:

|  |  |
| --- | --- |
| participated | accident |
|  |  |

## d) Find report numbers for all the accidents in which the cars belonging to “John Smith” were involved. ( subquery)

Solution:

SELECT report\_number FROM participated WHERE

license IN (SELECT license FROM owns WHERE

driver\_id = (SELECT driver\_id FROM person WHERE name = 'John Smith'));

Output:

|  |
| --- |
| participated |
|  |

## e) Delete the Mazda belonging to “John Smith”.

Solution:

DELETE FROM owns WHERE

driver\_id = (SELECT driver\_id FROM person WHERE name = 'John Smith') AND

license IN (SELECT license FROM car WHERE model = 'mazda');

Output:

|  |
| --- |
| owns |
|  |

## f) Create a view to get date and locations of all accidents.

Solution:

CREATE VIEW dam\_and\_location AS SELECT accident.location, participated.damage\_amount FROM accident, participated WHERE accident.report\_number = participated.report\_number;

Output:

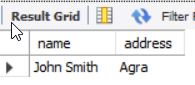
|  |
| --- |
| dam\_and\_location |
|  |

## g) Find the name and addresses of owners of cars involved in accidents for which damage amount is greater than $2000.

Solution:

SELECT name, address FROM person WHERE driver\_id = (SELECT driver\_id FROM owns WHERE license = (SELECT license FROM participated WHERE damage\_amount > 2000));

Output:



## h) Add car prices in the appropriate relation.

Solution:

ALTER TABLE car ADD COLUMN price INT DEFAULT 5000;

Output:

|  |
| --- |
| DESCRIBE car |
|  |

## i) Create view to have the information of persons and their cars .

Solution:

CREATE VIEW person\_car AS SELECT person.driver\_id, person.name, person.address, car.license, car.model, car.manufacture\_year, car.price FROM person, car, owns

WHERE person.driver\_id = owns.driver\_id AND car.license = owns.license;

SELECT \* FROM person\_car;

Output:

|  |
| --- |
| person\_car |
|  |

## j) Rename the name of the table person to owners.

Solution:

RENAME TABLE person to owners;

Output:

|  |
| --- |
| DESCRIBE owners |
|  |

## k) Change the type of attribute damage amount from int to numeric.

Solution:

ALTER TABLE participated MODIFY damage\_amount NUMERIC;

Output:

|  |
| --- |
| DESCRIBE participated |
|  |

## l) Delete all values in accident table but keep the relation in schema.

Solution:

SET SQL\_SAFE\_UPDATES = 0;

DELETE FROM accident;

Output:

|  |
| --- |
| accident |
|  |

## m) Modify the accident table to increase the report number by one for each new insertion.

Solution:

ALTER TABLE accident MODIFY report\_number INT AUTO\_INCREMENT;

Output:

Error:

Error Code: 1832. Cannot change column 'report\_number': used in a foreign key constraint 'accident\_ibfk\_1'

Explanation:

Changing column report number from accident table is not allowed because it is a foreign key referring to report number from participated table.

CREATE TABLE accident (report\_number INT PRIMARY KEY, location VARCHAR(50),

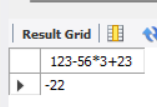
FOREIGN KEY fk\_acc\_report\_number(report\_number) REFERENCES participated(report\_number));

# 3. Write a query to calculate 123-56\*3+23.

Solution:

SELECT 123-56\*3+23

Output:



========================= End of Assignment =========================