Assignment 1

Contents

[1. Create following banking database schema and insert appropriate data: 2](#_Toc121696513)

[a) Design all tables with appropriate keys and constraints. 2](#_Toc121696514)

[b) Find the names of all branches located in “Chicago”. 5](#_Toc121696515)

[c) Add new customer in branch “Atlanta”, Consider appropriate values for other attributes. 5](#_Toc121696516)

[d) Find all the account numbers having balance greater than $1000 5](#_Toc121696517)

[e) Create a view to retrieve all the customers in “Atlanta” city with balance less than balance of customer “John”. (subquery) 6](#_Toc121696518)

[f) Modify the relations so that the default branch city will be “Washington”. 6](#_Toc121696519)

[g) Get the list of all customers only if the asset value for that branch is $35000 (EXISTS clause). 6](#_Toc121696520)

[2. Create Insurance database and insert appropriate data: 7](#_Toc121696521)

# 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),

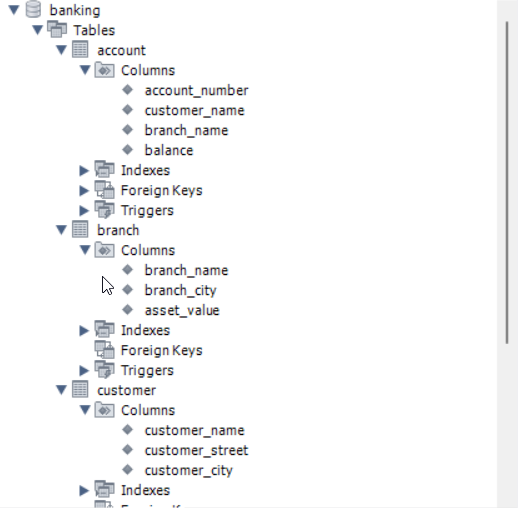
(3, 'Harry', 'Chi-1', 1000),

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

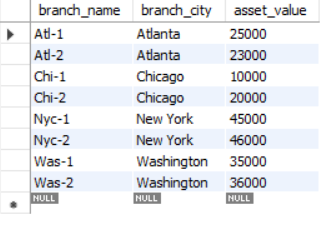
(5, 'David', 'Chi-2', 1500),

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

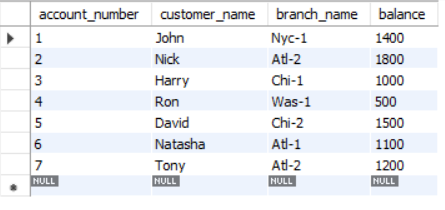
Output:



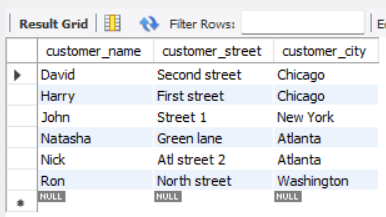
branch table:



account table:



customer table:

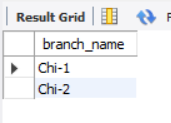


## b) Find the names of all branches located in “Chicago”.

Solution:

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

Output:



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

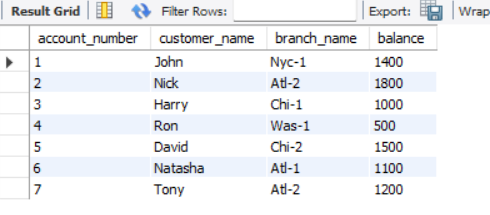
Solution:

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

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

SELECT \* FROM account;

Output:

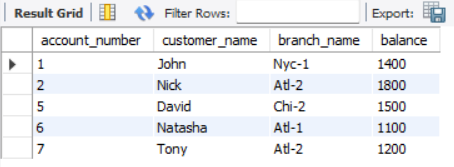


## d) Find all the account numbers having balance greater than $1000

Solution:

SELECT \* FROM account WHERE balance > 1000;

Output:



## 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

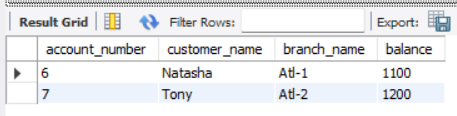
SELECT \* FROM account WHERE

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:



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

Solution:

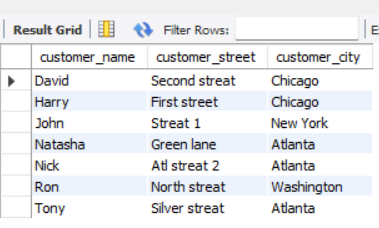
ALTER TABLE branch ALTER branch\_city SET DEFAULT 'Washington';

## 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:



# 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 accident (report\_number INT PRIMARY KEY, location VARCHAR(50));

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

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));

Output:

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