Bandr AlSwyan
Dr. Mohammad Mehdi Owrang
CSC-634-001 (Summer 2020)
14 July 2020

HW 2: Bank Database

Retrieval Queries:

1. Find all loan number for loans made at the Perryridge branch with loan amounts greater than $1100.
    a. SELECT loan_number FROM loan WHERE branch_name = 'Perryridge' AND amount > 1100;

loan_number
L-15
L-16

2. Find the loan number of those loans with loan amounts between $1,000 and $1,500 (that is, >=$1,000 and <=$1,500)
    a. SELECT loan_number FROM loan WHERE amount >= 1000 AND amount <= 1500;

loan_number
L-14
L-15
L-16
L-17

3. Find the names of all branches that have greater assets than some branch located in Brooklyn
    a. SELECT DISTINCT X.branch_name FROM branch AS X, branch AS Y WHERE X.assets > Y.assets AND Y.branch_city='Brooklyn';

branch_name
Downtown
Round Hill

4. Find the customer names and their loan numbers for all customers having a loan at some branch.
    a. SELECT borrower.customer_name, borrower.loan_number FROM borrower INNER JOIN loan ON borrower.loan_number = loan.loan_number;

customer_name	loan_number
Adams	L-16
Curry	L-93
Hayes	L-15
Jackson	L-14
Jones	L-17
Smith	L-11
Smith	L-23
Williams	L-17

Bandr AlSwyan
Dr. Mohammad Mehdi Owrang
CSC-634-001 (Summer 2020)
14 July 2020

5. Find all customers who have a loan, an account, or both:
    a. (SELECT customer_name FROM depositor) UNION (SELECT customer_name FROM borrower);

customer_name
Hayes
Johnson
Jones
Lindsay
Smith
Turner
Adams
Curry
Jackson
Williams

6. Find all customers who have an account but no loan.
    a. SELECT customer_name FROM depositor WHERE customer_name NOT IN (SELECT customer_name FROM borrower);

customer_name
Johnson
Johnson
Lindsay
Turner

7. Find the number of depositors for each branch.
    a. First way: SELECT branch_name, count(DISTINCT customer_name) FROM depositor, account WHERE depositor.account_number = account.account_number GROUP BY account.branch_name;
    b. Second way: SELECT branch_name, count(DISTINCT customer_name) FROM depositor LEFT JOIN account ON depositor.account_number = account.account_number GROUP BY account.branch_name;

branch_name	count(DISTINCT customer_name)
Brighton	2
Mianus	1
Perryridge	2
Redwood	1
Round Hill	1

Bandr AlSwyan
Dr. Mohammad Mehdi Owrang
CSC-634-001 (Summer 2020)
14 July 2020

8. Find the names of all branches where the average account balance is more than $500.
    a. SELECT branch_name, AVG(balance) FROM account GROUP BY branch_name HAVING AVG(balance) > 500;

branch_name	AVG(balance)
Brighton	825
Mianus	700
Redwood	700

9. Find all customers who have both an account and a loan at the bank.
    a. SELECT customer_name FROM depositor WHERE customer_name IN (SELECT customer_name FROM borrower);

customer_name
Hayes
Jones
Smith

10. Find all customers who have a loan at the bank but do not have an account at the bank
    a. SELECT DISTINCT customer_name FROM borrower WHERE NOT EXISTS (SELECT customer_name FROM depositor WHERE borrower.customer_name = depositor.customer_name);

customer_name
Adams
Curry
Jackson
Williams

11. Find the names of all branches that have greater assets than all branches located in Horseneck. (using both non-nested and nested select statement)
    a. SELECT branch_name FROM branch WHERE assets > ALL(SELECT assets FROM branch WHERE branch_city = 'Horseneck');

branch_name
Downtown

12. 1 query of your choice involving aggregate functions
    a. SELECT MAX(assets) FROM branch;

MAX(assets)
9000000

Bandr AlSwyan
Dr. Mohammad Mehdi Owrang
CSC-634-001 (Summer 2020)
14 July 2020

13. 1 query of your choice involving group by feature.
    a. SELECT DISTINCT I.branch_city, I.branch_name, SUM(I.assets) FROM branch AS I, branch as J WHERE I.branch_city = J.branch_city GROUP BY I.branch_city;

branch_city	branch_name	SUM(I.assets)
Bennington	Pownal	300000
Brooklyn	Brighton	32200000
Horseneck	Mianus	30300000
Palo Alto	Redwood	2100000
Rye	North Town	3700000

Bandr AlSwyan
Dr. Mohammad Mehdi Owrang
CSC-634-001 (Summer 2020)
14 July 2020

Insert Queries:

Do 2 insert queries requiring multiple records insertions as follow:

1. Create a HighLoan table with loan amount >=1500.
    a. CREATE TABLE HighLoan (loan_number varchar(5), branch_name varchar(10), amount double);
    b. INSERT INTO HighLoan SELECT loan_number, branch_name, amount FROM loan WHERE amount >= 1500;

loan_number	branch_name	amount
L-14	Downtown	1500
L-15	Perryridge	1500
L-23	Redwood	2000

2. Create a HighSalaryEmployee table with employee having salary more than 2000.
    a. CREATE TABLE HighSalaryEmployee (employee_name varchar(20) NOT NULL, branch_name varchar(10) NOT NULL, salary double DEFAULT NULL, PRIMARY KEY (`employee_name`,`branch_name`));
    b. INSERT INTO HighSalaryEmployee SELECT employee_name, branch_name, salary FROM employee WHERE salary >= 2000;

employee_name	branch_name	salary
Gopal	Perryridge	5300
Peterson	Downtown	2500

3. 1 more query (meaningful) of your choice on any table.
    a. CREATE TABLE VIPAccount (account_number char(5) NOT NULL, branch_name varchar(10) DEFAULT NULL, balance double DEFAULT NULL, PRIMARY KEY (`account_number`));
    b. INSERT INTO VIPAccount SELECT account_number, branch_name, balance FROM account WHERE balance >= 500;

account_number	branch_name	balance
A-101	Downtown	500
A-201	Brighton	900
A-215	Mianus	700
A-217	Brighton	750
A-222	Redwood	700

Bandr AlSwyan
Dr. Mohammad Mehdi Owrang
CSC-634-001 (Summer 2020)
14 July 2020

Update Queries:

1. Increase all accounts with balances over $800 by 7%, all other accounts receive 8%.
    a. UPDATE account SET balance = CASE WHEN BALANCE >= 800 then balance * 1.07 ELSE balance * 1.08 END;
    b. SELECT account_number, branch_name, balance FROM account

account_number	branch_name	balance
A-101	Downtown	500
A-102	Perryridge	400
A-201	Brighton	900
A-215	Mianus	700
A-217	Brighton	750
A-222	Redwood	700
A-305	Round Hill	350

account_number	branch_name	balance
A-101	Downtown	540
A-102	Perryridge	432
A-201	Brighton	963
A-215	Mianus	756
A-217	Brighton	810
A-222	Redwood	756
A-305	Round Hill	378

2. Do 2 update queries, each involving 2 tables.
    a. UPDATE account INNER JOIN VIPAccount ON VIPAccount.account_number = account.account_number SET account.balance = 799, VIPAccount.balance = 799;

account_number	branch_name	balance
A-101	Downtown	799
A-102	Perryridge	432
A-201	Brighton	799
A-215	Mianus	799
A-217	Brighton	799
A-222	Redwood	799
A-305	Round Hill	378

    b. UPDATE loan INNER JOIN HighLoan ON HighLoan.loan_number = loan.loan_number SET loan.amount = loan.amount + 20, HighLoan.amount = HighLoan.amount + 20;

Bandr AlSwyan
Dr. Mohammad Mehdi Owrang
CSC-634-001 (Summer 2020)
14 July 2020

loan_number	branch_name	amount
L-11	Round Hill	900
L-14	Downtown	1500
L-15	Perryridge	1500
L-16	Perryridge	1300
L-17	Downtown	1000
L-23	Redwood	2000
l-93	Mianus	500

loan_number	branch_name	amount
L-11	Round Hill	900
L-14	Downtown	1520
L-15	Perryridge	1520
L-16	Perryridge	1300
L-17	Downtown	1000
L-23	Redwood	2020
l-93	Mianus	500

3. 1 more update query of your choice on any table.
    a. UPDATE branch SET assets = assets + 200 WHERE branch_city = 'Horseneck';

branch_name	branch_city	assets
Brighton	Brooklyn	7100000
Downtown	Brooklyn	9000000
Mianus	Horseneck	400000
North Town	Rye	3700000
Perryridge	Horseneck	1700000
Pownal	Bennington	300000
Redwood	Palo Alto	2100000
Round Hill	Horseneck	8000000

branch_name	branch_city	assets
Brighton	Brooklyn	7100000
Downtown	Brooklyn	9000000
Mianus	Horseneck	400200
North Town	Rye	3700000
Perryridge	Horseneck	1700200
Pownal	Bennington	300000
Redwood	Palo Alto	2100000
Round Hill	Horseneck	8000200

Bandr AlSwyan
Dr. Mohammad Mehdi Owrang
CSC-634-001 (Summer 2020)
14 July 2020

Delete Queries:

1. Delete the record of all accounts with balances below the average at the bank.
    a. DELETE FROM account WHERE balance < (select AVG(balance) FROM account);
2. Do 2 update queries, each involving 2 tables.
    a. DELETE FROM employee WHERE employee.employee_name IN(SELECT HighSalaryEmployee.employee_name FROM HighSalaryEmployee WHERE HighSalaryEmployee.employee_name = employee.employee_name);

employee_name	branch_name	salary
Adams	Perryridge	1500
Brown	Perryridge	1300
Gopal	Perryridge	5300
Johnson	Downtown	1500
Loreena	Downtown	1300
Peterson	Downtown	2500
Rao	Austin	1500
Sato	Austin	1600

employee_name	branch_name	salary
Adams	Perryridge	1500
Brown	Perryridge	1300
Johnson	Downtown	1500
Loreena	Downtown	1300
Rao	Austin	1500
Sato	Austin	1600

3. 1 more delete query of your choice from any table.
    a. DELETE FROM customer WHERE customer_name = 'Adams';

customer_name	customer_street	customer_city
Adams	Spring	Pittsfield
Brooks	Senator	Brooklyn
Curry	North	Rye
Glenn	Sand Hill	Woodside
Green	Walnut	Stamford
Hayes	Main	Harrison
Johnson	Alma	Palo Alto
Jones	Main	Harrison
Lindsay	Park	Pittsfield
Smith	North	Rye
Turner	Putnam	Stamford
Williams	Nassau	Princeton

Bandr AlSwyan
Dr. Mohammad Mehdi Owrang
CSC-634-001 (Summer 2020)
14 July 2020

customer_name	customer_street	customer_city
Brooks	Senator	Brooklyn
Curry	North	Rye
Glenn	Sand Hill	Woodside
Green	Walnut	Stamford
Hayes	Main	Harrison
Johnson	Alma	Palo Alto
Jones	Main	Harrison
Lindsay	Park	Pittsfield
Smith	North	Rye
Turner	Putnam	Stamford
Williams	Nassau	Princeton

Views Queries:

1. A view consisting of branches and their customers
    a. CREATE VIEW customers_view AS (SELECT branch_name, customer_name FROM depositor INNER JOIN account ON account.account_number = depositor.account_number) UNION (SELECT branch_name, customer_name FROM borrower INNER JOIN loan ON loan.loan_number = borrower.loan_number);

branch_name	customer_name
Perryridge	Hayes
Perryridge	Johnson
Brighton	Johnson
Brighton	Jones
Redwood	Lindsay
Mianus	Smith
Round Hill	Turner
Perryridge	Adams
Mianus	Curry
Downtown	Jackson
Downtown	Jones
Round Hill	Smith
Redwood	Smith
Downtown	Williams

2. Create a view of HQEmployee who work in downtown branch.
    a. CREATE VIEW HQEmployee AS (SELECT employee_name, branch_name, salary FROM employee WHERE branch_name = 'Downtown');

employee_name	branch_name	salary
Johnson	Downtown	1500
Loreena	Downtown	1300

Bandr AlSwyan
Dr. Mohammad Mehdi Owrang
CSC-634-001 (Summer 2020)
14 July 2020

3. Do one insert, delete, update, and select queries on HQEmployee view.
    a. INSERT INTO HQEmployee(employee_name, branch_name, salary) VALUES ('Bandar', 'Downtown', 2020);

```
| employee_name | branch_name | salary |
|---------------|-------------|--------|
|        Bandar |    Downtown |   2020 |
|       Johnson |    Downtown |   1500 |
|       Loreena |    Downtown |   1300 |
```

    b. DELETE FROM HQEmployee WHERE employee_name = 'Bandar';

```
| employee_name | branch_name | salary |
|---------------|-------------|--------|
|       Johnson |    Downtown |   1500 |
|       Loreena |    Downtown |   1300 |
```

    c. SELECT employee_name, salary FROM HQEmployee WHERE salary > 1300;

```
| employee_name | salary |
|---------------|--------|
|       Johnson |   1500 |
```

Complex Queries: (provide results)

1. 1 select query involving 3 tables
    a. SELECT C.customer_name, D.account_number, A.balance FROM customer AS C INNER JOIN depositor AS D ON D.customer_name = C.customer_name INNER JOIN account AS A ON A.account_number = D.account_number;

```
$sqlite3 database.sdb < main.sql

Hayes|A-102|432.0
Johnson|A-102|432.0
Johnson|A-201|799.0
Jones|A-217|799.0
Lindsay|A-222|799.0
Smith|A-215|799.0
Turner|A-305|378.0
```

2. 1 Delete query involving 3 tables
    a. DELETE C.*, D.*, A.* FROM customer AS C
    INNER JOIN depositor AS D ON D.customer_name = C.customer_name
    INNER JOIN account AS A ON A.account_number = D.account_number
    WHERE A.balance = 378;

```
$sqlite3 database.sdb < main.sql

Hayes|A-102|432.0
Johnson|A-102|432.0
Johnson|A-201|799.0
Jones|A-217|799.0
Lindsay|A-222|799.0
Smith|A-215|799.0
```

Bandr AlSwyan
Dr. Mohammad Mehdi Owrang
CSC-634-001 (Summer 2020)
14 July 2020

3. 1 Update query involving 3 tables
    a. UPDATE branch as B INNER JOIN account as A ON A.branch_name = B.branch_name
    INNER JOIN employee as E ON A.branch_name = E.branch_name
    SET B.branch_name = 'AU', A.branch_name = 'AU', E.branch_name = 'AU'
    WHERE B.branch_name = 'North Town';

```
Brighton|Brooklyn|7100000.0
Downtown|Brooklyn|9000000.0
Mianus|Horseneck|400200.0
North Town|Rye|3700000.0
Perryridge|Horseneck|1700200.0
Pownal|Bennington|300000.0
Redwood|Palo Alto|2100000.0
Round Hill|Horseneck|8000200.0
```

```
$sqlite3 database.sdb < main.sql

Brighton|Brooklyn|7100000.0
Downtown|Brooklyn|9000000.0
Mianus|Horseneck|400200.0
AU|Rye|3700000.0
Perryridge|Horseneck|1700200.0
Pownal|Bennington|300000.0
Redwood|Palo Alto|2100000.0
Round Hill|Horseneck|8000200.0
```