STEVENS INSTITUTE OF TECHNOLOGY
FE 513 - Database Design
Instructor: Cheng Lu

Homework 2

Student: Aidana Bekboeva

Contents

1 Part I. Basic Query

1.1 Task 1

Given the data of "banks_al_2001.csv", define the table structure accordingly in the database.

```
DROP TABLE banks
CREATE TABLE IF NOT EXISTS public.banks
(
    id integer,
    date date,
    asset integer,
    liability integer
)
```

1.2 Task 2

Import data from "banks_al_2001.csv"

```
COPY banks (id, date, asset, liability)
FROM 'C:\Users\Public\banks_al_2001.csv'
DELIMITER ','
CSV HEADER

SELECT * FROM banks
```

	id integer	date date	asset integer	liability integer
1	23373	2001-09-30	90716	82518
2	23375	2001-03-31	221592	202713
3	23375	2001-06-30	236213	218446
4	23376	2001-12-31	79250	72170
5	23376	2001-03-31	72025	65682
6	23376	2001-06-30	72180	65576
7	23376	2001-09-30	72941	66057
8	234	2001-12-31	56657	49558
9	234	2001-03-31	33686	27517
10	234	2001-06-30	32979	26823

1.3 Task 3

Query the table and count the number of banks for each quarter.

```
SELECT EXTRACT (quarter from date) AS quarter, COUNT(distinct id) FROM banks
GROUP BY quarter
ORDER BY quarter ASC
```

	quarter double precision	count bigint
1	1	9838
2	2	9763
3	3	9718
4	4	9630

1.4 Task 4

Query the table and report the average of asset for each bank.

```
SELECT id, ROUND(AVG(asset), 2) AS asset_avg
FROM banks
GROUP BY id
```

```
id    asset_avg
30052 923423
26264 84237.75
11719 32653.5
23626 103813.5
25173 88386.75
29089 99908.25
35532 11445.67
25886 40250.5
34418 35471
15820 25620.75
34266 41027.75
9719  386023
4993  538870.5
10628 38985.25
1003  1485000
292   225362.25
17573 16829.75
2784  88488.75
1552  268852
```

Listing 1: R output

1.5 Task 5

Query the table and report the bank id who has the second largest asset for second quarter.

```
SELECT id FROM banks
WHERE date = '2001-06-30' ORDER BY asset DESC LIMIT 1 OFFSET 1;
```

Result: 628

1.6 Task 6

Query the table and report the bank id whose equity is over 10% of its asset in the first quarter(hint: equity = asset-liability)

```
WITH tabl AS (
SELECT *, EXTRACT(quarter FROM date) AS quarter, asset-liability AS equity
  FROM banks
)
```

```
5
6  SELECT id FROM tabl WHERE equity > 0.1 * asset AND quarter = 1
7  ORDER BY id
```

	id integer
1	28
2	39
3	41
4	43
5	51
6	52
7	54
8	59
9	60
10	63
11	68
12	70
13	72
14	77
15	81

2 Part II. Advance Query

A company's executives are interested in seeing who earns the most money in each of the company's departments. A high earner in a department is an employee who has a salary in the top three unique salaries for that department.

```
1  DROP TABLE IF EXISTS department;
2  DROP TABLE IF EXISTS employee;
3
4  CREATE TABLE department (department_id VARCHAR (255) PRIMARY KEY,
5              department_name VARCHAR (255));
6
7  CREATE TABLE employee ( employee_id INT PRIMARY KEY,
8              employee_name VARCHAR (255),
9              employee_salary INT NOT NULL,
10             department_id VARCHAR (255),
11         FOREIGN KEY (department_id) REFERENCES department (department_id) ON DELETE SET NULL
       );
12
13 INSERT INTO department
14 VALUES (1, 'IT'),
15 (2, 'Sales');
16 INSERT INTO employee
17 VALUES (1, 'Joe', 85000, 1),
18      (2, 'Henry', 80000, 2),
19      (3, 'Sam', 60000, 2),
20      (4, 'Max', 90000, 1),
21      (5, 'Janet', 69000, 1),
22      (6, 'Randy', 85000, 1),
23      (7, 'Will', 70000, 1);
24
25 SELECT * FROM department;
26 SELECT * FROM employee;
```

	department_id [PK] character varying (255)	department_name character varying (255)
1	1	IT
2	2	Sales

	employee_id [PK] integer	employee_name character varying (255)	employee_salary integer	department_id character varying (255)
1	1	Joe	85000	1
2	2	Henry	80000	2
3	3	Sam	60000	2
4	4	Max	90000	1
5	5	Janet	69000	1
6	6	Randy	85000	1
7	7	Will	70000	1

Write an SQL query to find the employees who are high earners in each of the departments.

```sql
SELECT a.department_name AS "Department",
       b.employee_name AS "Employee",
     b.employee_salary
FROM Department a
LEFT JOIN Employee b
ON a. department_id = b.department_id
WHERE (SELECT count (DISTINCT employee_salary) FROM Employee
     WHERE department_id = a.department_id and employee_salary > b.employee_salary) < 3
AND b.employee_salary IS NOT NULL;
```

	Department character varying (255)	Employee character varying (255)	employee_salary integer
1	IT	Joe	85000
2	Sales	Henry	80000
3	Sales	Sam	60000
4	IT	Max	90000
5	IT	Randy	85000
6	IT	Will	70000