**FE 513: Homework Assignment 3**

Rishikesh Yadav

CWID: 20007668

1. **Querying Multiple Tables**
2. Import data from banks\_sec\_2002 and banks\_al\_2002. Delete duplicate rows from banks\_sec\_2002.

**Script:**

**--Creating tables**

DROP TABLE IF EXISTS banks\_sec\_2002;

CREATE TABLE IF NOT EXISTS banks\_sec\_2002 (

id INTEGER NOT NULL,

date DATE NOT NULL,

security INTEGER NOT NULL

);

DROP TABLE IF EXISTS banks\_al\_2002;

CREATE TABLE IF NOT EXISTS banks\_al\_2002 (

id INTEGER NOT NULL,

date DATE NOT NULL,

asset INTEGER NOT NULL,

liability INTEGER NOT NULL

);

**--Reading CSV files and adding data to the tables**

COPY banks\_sec\_2002(id, date, security) FROM 'C:\Users\psyad\Desktop\Stevens\Sem 4\FE 513\HW-Assignments\Assignment 3\banks\_sec\_2002.csv' DELIMITER ',' CSV HEADER;

COPY banks\_al\_2002(id, date, asset, liability) FROM 'C:\Users\psyad\Desktop\Stevens\Sem 4\FE 513\HW-Assignments\Assignment 3\banks\_al\_2002-1.csv' DELIMITER ',' CSV HEADER;

SELECT \* FROM banks\_sec\_2002;

Table

Description automatically generated

SELECT \* FROM banks\_al\_2002;

Table

Description automatically generated

**--Check duplicate in banks\_sec\_2002**

SELECT id, date, security, count(id) as cnt from banks\_sec\_2002 group by id, date, security having count(id) > 1;

Graphical user interface, application, table

Description automatically generated

**--Delete duplicate from banks\_sec\_2002**

DELETE FROM banks\_sec\_2002 WHERE ctid not in ( SELECT MIN(ctid) FROM banks\_sec\_2002 GROUP BY id, date, security);

Graphical user interface, text, application, email

Description automatically generated

1. Select proper join manner to join banks\_sec\_2002 and banks\_al\_2002. Make sure that all data from banks\_sec\_2002 are kept in the joint table. Report the first 10 observations.

**Script:**

SELECT bs.id, bs.date, bs.security, ba.asset, ba.liability FROM

banks\_sec\_2002 bs INNER JOIN banks\_al\_2002 ba ON ba.id = bs.id AND ba.date = bs.date

LIMIT 10;

Table

Description automatically generated

1. Create a new table bank\_total. Insert the values from previous joint table into this new one. And set a primary key for the table.

**Script:**

**--Create table banks\_total**

CREATE TABLE banks\_total AS

SELECT ROW\_NUMBER() OVER (ORDER BY bs.id, bs.date) AS pkey, bs.id, bs.date, bs.security, ba.asset, ba.liability from banks\_sec\_2002 bs

INNER JOIN banks\_al\_2002 ba ON ba.id = bs.id AND ba.date = bs.date;

Graphical user interface, text, application, chat or text message, email

Description automatically generated

ALTER TABLE banks\_total ADD PRIMARY KEY(pkey);

Graphical user interface, text, application

Description automatically generated

SELECT \* FROM banks\_total;

Table

Description automatically generated

1. For each quarter of the year 2002 count how many banks have security over 20% of its’ asset.

**Script:**

Select COUNT(\*) as Q1\_banks

from banks\_total

where (extract(quarter from date) = 1) and security > (0.2 \* asset);

Graphical user interface, text, application, chat or text message, email

Description automatically generated

Select COUNT(\*) as Q2\_banks

from banks\_total

where (extract(quarter from date) = 2) and security > (0.2 \* asset);

Graphical user interface, text, application, email

Description automatically generated

Select COUNT(\*) as Q3\_banks

from banks\_total

where (extract(quarter from date) = 3) and security > (0.2 \* asset);

Graphical user interface, text, application, email

Description automatically generated

Select COUNT(\*) as Q4\_banks

from banks\_total

where (extract(quarter from date) = 4) and security > (0.2 \* asset)

Graphical user interface, text, application, email

Description automatically generated

1. How many banks have liability over 90% of assets in first quarter of 2002 but goes below 90% in the second quarter of 2002.

**Script:**

SELECT COUNT(\*) AS banks FROM

(SELECT id FROM banks\_total WHERE (extract(quarter from date) = 1) AND liability > (0.9 \* asset)) inc

INNER JOIN

(SELECT id FROM banks\_total WHERE (extract(quarter from date) = 2) and liability < (0.9 \* asset)) dec

ON dec.id = inc.id;

Graphical user interface, text, application, chat or text message, email

Description automatically generated

1. Export the joint table (banks total) to a csv file.

**Script:**

COPY banks\_total TO 'C:\Users\psyad\Desktop\Stevens\Sem 4\FE 513\HW-Assignments\Assignment 3\banks\_total.csv' DELIMITER ',' CSV HEADER;

Graphical user interface, text, application, email

Description automatically generated Graphical user interface, application, table, Excel

Description automatically generated

1. **PostgreSQL API in R**

Complete following tasks in R using PostgreSQL API:

1. Make a connection to your local PostgreSQL database using API.

**Script:**

library(RPostgreSQL)

database\_name <- "FE\_513"

username <- "postgres"

drv <- dbDriver("PostgreSQL")

con<-dbConnect(drv, dbname = database\_name, user = username, password = "root")

A picture containing shape

Description automatically generated

1. Import the csv file you got from Problem 1 (banks\_total) into a new table in the database using API. (Hint. Please give the table a new name if table ‘banks\_total’ exists in the database)

**Script:**

banks\_total <- read.csv("C:\Users\psyad\Desktop\Stevens\Sem 4\FE 513\HW-Assignments\Assignment 3\banks\_total.csv", header = TRUE, sep = ",")

dbWriteTable(con, "banks\_total\_new", banks\_total, row.names=TRUE, append=TRUE)

Graphical user interface, table

Description automatically generated

1. Retrieve the data of table ‘banks\_total’ using API. Count how many rows in the table.

**Script:**

res<- dbGetQuery(con, "Select \* from banks\_total")

nrow(res)



Graphical user interface, application

Description automatically generated