**LORENZO AUSIELLO**

**ASSIGNMENT 3**

**PROBLEM 1**

-- Import data from banks sec 2002 and banks al 2002. Delete duplicate

-- rows from banks sec 2002

DROP TABLE IF EXISTS banks\_sec\_2002;

SET datestyle = 'MDY';

CREATE TABLE banks\_sec\_2002(

id INTEGER,

date DATE,

security INTEGER

);

SELECT \* FROM banks\_sec\_2002;

COPY banks\_sec\_2002 (id, date, security)

FROM 'C:\Users\Public\banks\_sec\_2002.csv'

DELIMITER ','

CSV HEADER;

SELECT id, date, security, COUNT(\*)

FROM banks\_sec\_2002

GROUP BY id, date, security

HAVING COUNT(\*) > 1;

DELETE FROM banks\_sec\_2002 a

USING (

SELECT id, date, security, MIN(ctid) AS min\_ctid

FROM banks\_sec\_2002

GROUP BY id, date, security

HAVING COUNT(\*) > 1

) AS b

WHERE a.id = b.id

AND a.date = b.date

AND a.security = b.security

AND a.ctid <> b.min\_ctid;

DROP TABLE IF EXISTS banks\_al\_2002;

CREATE TABLE banks\_al\_2002(

id INTEGER,

date DATE,

asset INTEGER,

liability INTEGER

);

SELECT \* FROM banks\_al\_2002;

COPY banks\_al\_2002 (id, date, asset,liability)

FROM 'C:\Users\Public\banks\_al\_2002-1.csv'

DELIMITER ','

CSV HEADER;

SELECT id, date, asset, liability, COUNT(\*)

FROM banks\_al\_2002

GROUP BY id, date, asset, liability

HAVING COUNT(\*) > 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.

SELECT \*

FROM banks\_sec\_2002 AS bs

LEFT JOIN banks\_al\_2002 AS ba

ON bs.id = ba.id

AND bs.date = ba.date

LIMIT 10;

Immagine che contiene testo, schermata, software, Icona del computer

Descrizione generata automaticamente

-- Create a new table banks total. Insert the values from previous joint table

-- into this new one. And set a primary key for the table.

DROP TABLE IF EXISTS banks\_total;

CREATE TABLE banks\_total (

id INTEGER,

date DATE,

security INTEGER,

asset INTEGER,

liability INTEGER,

PRIMARY KEY (id, date)

);

SELECT \* FROM banks\_total;

INSERT INTO banks\_total (id, date, security, asset, liability)

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

FROM banks\_sec\_2002 AS bs

LEFT JOIN banks\_al\_2002 AS ba

ON bs.id = ba.id

AND bs.date = ba.date;

SELECT \* FROM banks\_total;

Immagine che contiene testo, schermata, software, numero

Descrizione generata automaticamente

-- For each quarter of the year 2002 count how many banks have security

-- over 20% of its’ asset.

SELECT

EXTRACT(quarter FROM date) AS quarter,

COUNT(\*) AS banks\_count\_over\_20\_percent

FROM banks\_total

WHERE

EXTRACT(year FROM date) = 2002

AND security > (0.2 \* asset)

GROUP BY quarter

ORDER BY quarter;

Immagine che contiene testo, schermata, software, Icona del computer

Descrizione generata automaticamente

-- How many banks have liability over 90% of assets in first quarter of 2002

-- but goes below 90% in the second quarter of 2002

SELECT COUNT(DISTINCT t1.id)

FROM banks\_total t1

JOIN banks\_total t2 ON t1.id = t2.id

WHERE t1.date = '2002-03-31' AND t2.date = '2002-06-30'

AND t1.liability > 0.9 \* t1.asset

AND t2.liability < 0.9 \* t2.asset;

Immagine che contiene testo, schermata, software, Icona del computer

Descrizione generata automaticamente

-- create csv

COPY banks\_total

TO 'C:\Users\Public\banks\_total\_export.csv'

DELIMITER ','

CSV HEADER;

**PROBLEM 2**

library("RPostgres")

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

con <- dbConnect(RPostgres::Postgres(),

dbname = "postgres",

host = "127.0.0.1",

port = 5432,

user = "postgres",

password = '\*\*\*\*\*\*\*\*')

## Import the csv file you got from Problem 1 (banks total)

## into a new table in the database using API.

setwd('C:/Users/Public')

dbWriteTable(con, "new\_banks\_total", read.csv("banks\_total\_export.csv"))

## Retrieve the data of table ‘banks total‘ using API. Count how many rows

## in the table.

banks\_total <- dbGetQuery(con, "SELECT \* FROM banks\_total")

result <- dbGetQuery(con, "SELECT COUNT(\*) FROM banks\_total")

head(banks\_total)

Immagine che contiene testo, schermata, schermo, software

Descrizione generata automaticamente

print(result)

Immagine che contiene testo, schermata, schermo, software

Descrizione generata automaticamente

dbDisconnect(con)