FE 513: Homework Assignment 3

Rishikesh Yadav
CWID: 20007668

1. Querying Multiple Tables

1. 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

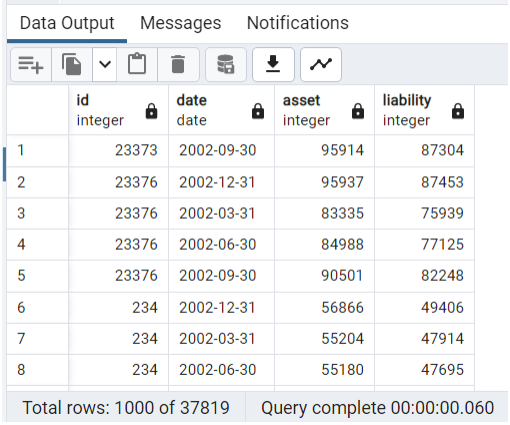
```
COPY banks_sec_2002(id, date, security) FROM 'C:\Users\psyad\Desktop\Stevens\Sem 4\FE 513\HW-Assignments\Assignment 3\banks_sec_2002.csv' DELIMITER ',' CSV HEADER;

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;
```

Data Output | Messages | Notifications

	id integer	date date	security integer
1	32307	2002-09-30	0
2	22598	2002-03-31	0
3	15879	2002-06-30	5357
4	35373	2002-06-30	0
5	5226	2002-09-30	7960
6	22092	2002-12-31	0
7	13749	2002-03-31	17476
8	823	2002-12-31	44972

Total rows: 1000 of 37822 | Query complete 00:00:00.074

SELECT * FROM banks_al_2002;



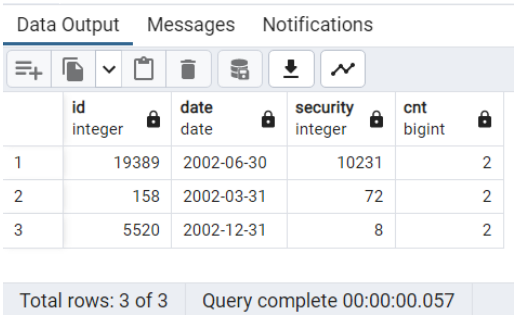
Data Output | Messages | Notifications

	id integer	date date	asset integer	liability integer
1	23373	2002-09-30	95914	87304
2	23376	2002-12-31	95937	87453
3	23376	2002-03-31	83335	75939
4	23376	2002-06-30	84988	77125
5	23376	2002-09-30	90501	82248
6	234	2002-12-31	56866	49406
7	234	2002-03-31	55204	47914
8	234	2002-06-30	55180	47695

Total rows: 1000 of 37819 | Query complete 00:00:00.060

--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;



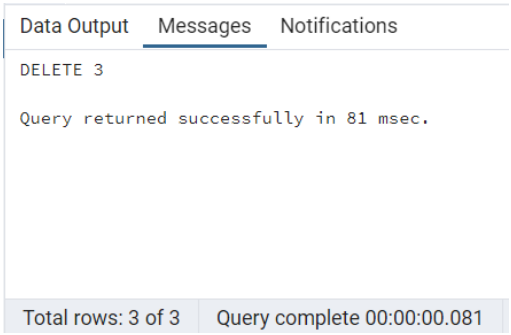
Data Output | Messages | Notifications

	id integer	date date	security integer	cnt bigint
1	19389	2002-06-30	10231	2
2	158	2002-03-31	72	2
3	5520	2002-12-31	8	2

Total rows: 3 of 3 | Query complete 00:00:00.057

--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);



Data Output | Messages | Notifications

```
DELETE 3

Query returned successfully in 81 msec.
```

Total rows: 3 of 3 | Query complete 00:00:00.081

2. 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;

Data Output | Messages | Notifications

	id integer	date date	security integer	asset integer	liability integer
1	32307	2002-09-30	0	53714	49350
2	22598	2002-03-31	0	57360	53205
3	15879	2002-06-30	5357	76960	62945
4	35373	2002-06-30	0	46551	38721
5	5226	2002-09-30	7960	53873	48146

Total rows: 10 of 10 Query complete 00:00:00.096

3. 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;
```

Data Output | Messages | Notifications

```
SELECT 37819

Query returned successfully in 118 msec.
```

Total rows: 10 of 10 Query complete 00:00:00.118

```
ALTER TABLE banks_total ADD PRIMARY KEY(pkey);
```

Data Output | Messages | Notifications

```
ALTER TABLE

Query returned successfully in 63 msec.
```

Total rows: 10 of 10 Query complete 00:00:00.063

```
SELECT * FROM banks_total;
```

Data Output | Messages | Notifications

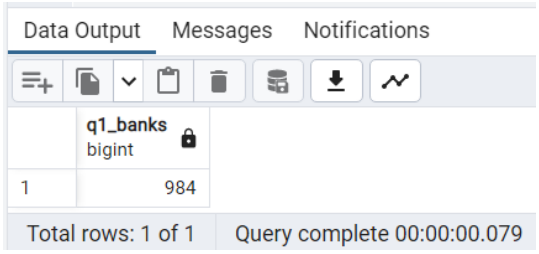
	pkey [PK] bigint	id integer	date date	security integer	asset integer	liability integer
1	1	9	2002-03-31	35605	348727	321479
2	2	9	2002-06-30	39440	361953	332900
3	3	9	2002-09-30	37081	383246	352456
4	4	9	2002-12-31	34957	371812	340365
5	5	14	2002-03-31	3700000	68600000	64300000
6	6	14	2002-06-30	3680000	73600000	69200000

Total rows: 1000 of 37819 Query complete 00:00:00.144

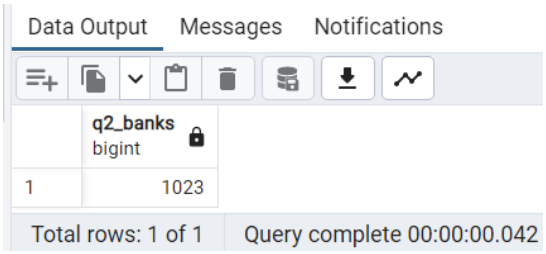
4. For each quarter of the year 2002 count how many banks have security over 20% of its' asset.

Script:

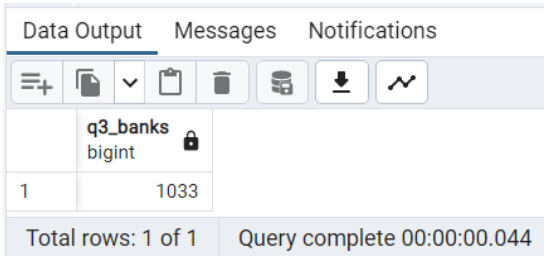
```
Select COUNT(*) as Q1_banks
from banks_total
where (extract(quarter from date) = 1) and security > (0.2 * asset);
```



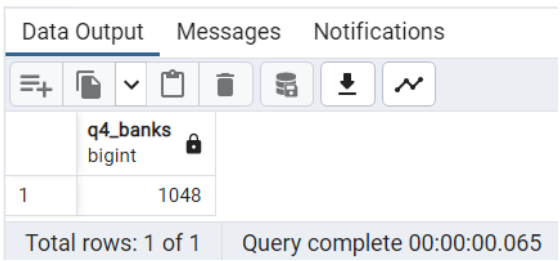

```
Select COUNT(*) as Q2_banks
from banks_total
where (extract(quarter from date) = 2) and security > (0.2 * asset);
```




```
Select COUNT(*) as Q3_banks
from banks_total
where (extract(quarter from date) = 3) and security > (0.2 * asset);
```



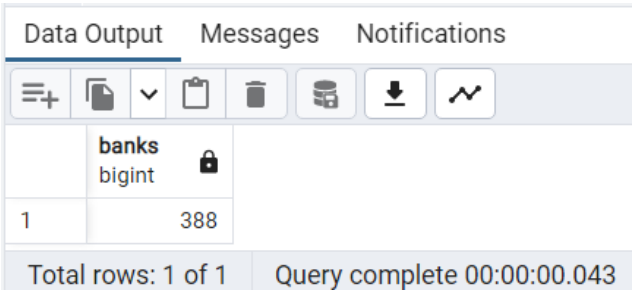

```
Select COUNT(*) as Q4_banks
from banks_total
where (extract(quarter from date) = 4) and security > (0.2 * asset)
```



5. 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:

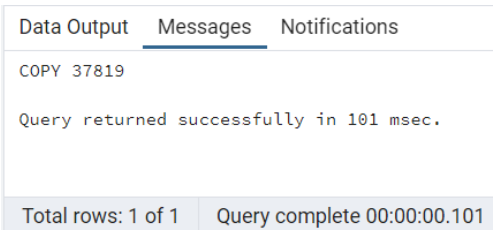
```
SELECT COUNT(*) AS banks FROM
(SELECT id FROM banks_total WHERE (extract(quarter from date) = 1) AND liability > (0.9 * asset)) inc
INNER JOIN
(SELECT id FROM banks_total WHERE (extract(quarter from date) = 2) and liability < (0.9 * asset)) dec
ON dec.id = inc.id;
```

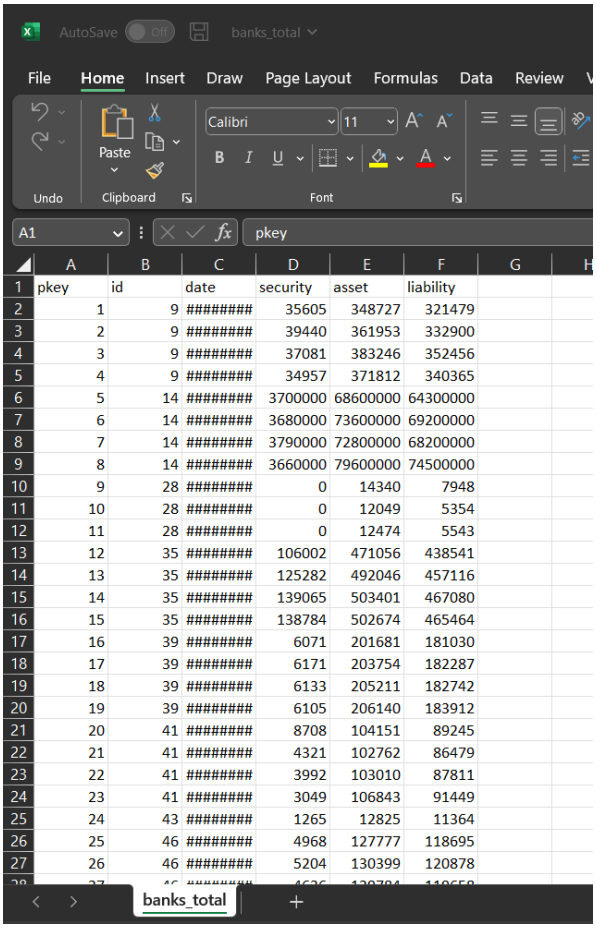


6. Export the joint table (banks total) to a csv file.

Script:

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





	A	B	C	D	E	F
1	pkey	id	date	security	asset	liability
2	1	9	########	35605	348727	321479
3	2	9	########	39440	361953	332900
4	3	9	########	37081	383246	352456
5	4	9	########	34957	371812	340365
6	5	14	########	3700000	68600000	64300000
7	6	14	########	3680000	73600000	69200000
8	7	14	########	3790000	72800000	68200000
9	8	14	########	3660000	79600000	74500000
10	9	28	########	0	14340	7948
11	10	28	########	0	12049	5354
12	11	28	########	0	12474	5543
13	12	35	########	106002	471056	438541
14	13	35	########	125282	492046	457116
15	14	35	########	139065	503401	467080
16	15	35	########	138784	502674	465464
17	16	39	########	6071	201681	181030
18	17	39	########	6171	203754	182287
19	18	39	########	6133	205211	182742
20	19	39	########	6105	206140	183912
21	20	41	########	8708	104151	89245
22	21	41	########	4321	102762	86479
23	22	41	########	3992	103010	87811
24	23	41	########	3049	106843	91449
25	24	43	########	1265	12825	11364
26	25	46	########	4968	127777	118695
27	26	46	########	5204	130399	120878

2. 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")
```

```
> con
<PostgreSQLConnection>
>
```

2. 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)
```

```
1  select * from banks_total_new
```

Data output | Messages | Notifications

	row.names text	pkey integer	id integer	date text	security integer	asset integer	liability integer
1	1	1	9	2002-03-31	35605	348727	321479
2	2	2	9	2002-06-30	39440	361953	332900
3	3	3	9	2002-09-30	37081	383246	352456
4	4	4	9	2002-12-31	34957	371812	340365
5	5	5	14	2002-03-31	3700000	68600000	64300000
6	6	6	14	2002-06-30	3680000	73600000	69200000
7	7	7	14	2002-09-30	3790000	72800000	68200000
8	8	8	14	2002-12-31	3660000	79600000	74500000
9	9	9	28	2002-03-31	0	14340	7948
10	10	10	28	2002-06-30	0	12049	5354
11	11	11	28	2002-09-30	0	12474	5543
12	12	12	35	2002-03-31	106002	471056	438541

Total rows: 1000 of 37819 | Query complete 00:00:00.152

3. 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)
```

```
> nrow(res)
[1] 37819
```

Filter

	pkey	id	date	security	asset	liability
1	1	9	2002-03-31	35605	348727	321479
2	2	9	2002-06-30	39440	361953	332900
3	3	9	2002-09-30	37081	383246	352456
4	4	9	2002-12-31	34957	371812	340365
5	5	14	2002-03-31	3700000	68600000	64300000
6	6	14	2002-06-30	3680000	73600000	69200000
7	7	14	2002-09-30	3790000	72800000	68200000
8	8	14	2002-12-31	3660000	79600000	74500000
9	9	28	2002-03-31	0	14340	7948
10	10	28	2002-06-30	0	12049	5354
11	11	28	2002-09-30	0	12474	5543
12	12	35	2002-03-31	106002	471056	438541
13	13	35	2002-06-30	125282	492046	457116
14	14	35	2002-09-30	139065	503401	467080
15	15	35	2002-12-31	138784	502674	465464
16	16	39	2002-03-31	6071	201681	181030
17	17	39	2002-06-30	6171	203754	182287
18	18	39	2002-09-30	6133	205211	182742

Showing 1 to 19 of 37,819 entries, 6 total columns