PES UNIVERSITY

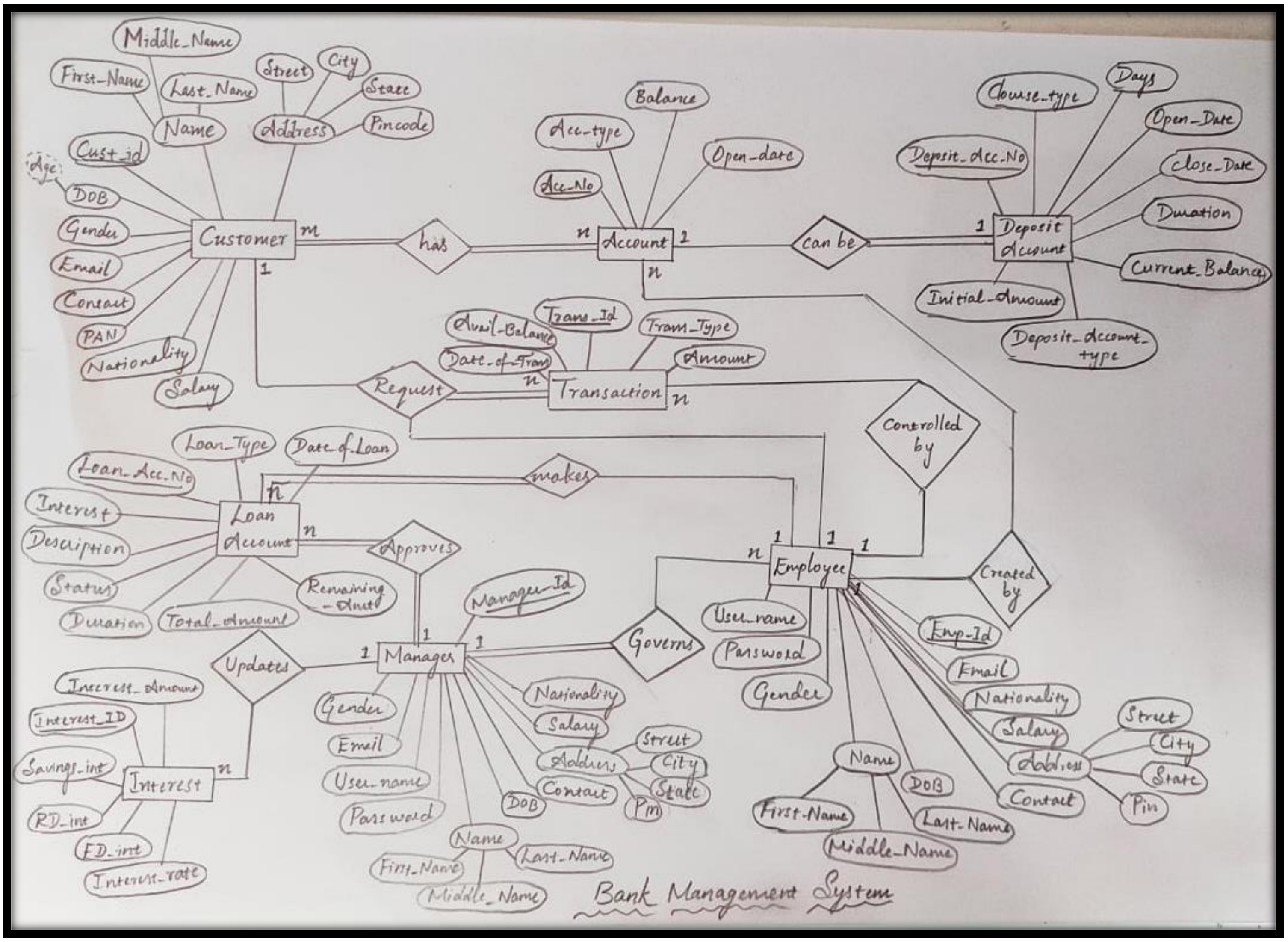
(Established under Karnataka Act No. 16 of 2013)
100 Feet Ring Road, BSK III Stage, Bengaluru-560 085
Department of Computer Science and Engineering
Session: Aug – Dec, 2021
SEMESTER – 5

DBMS PROJECT ASSIGNMENT 3

BANK MANAGEMENT SYSTEM

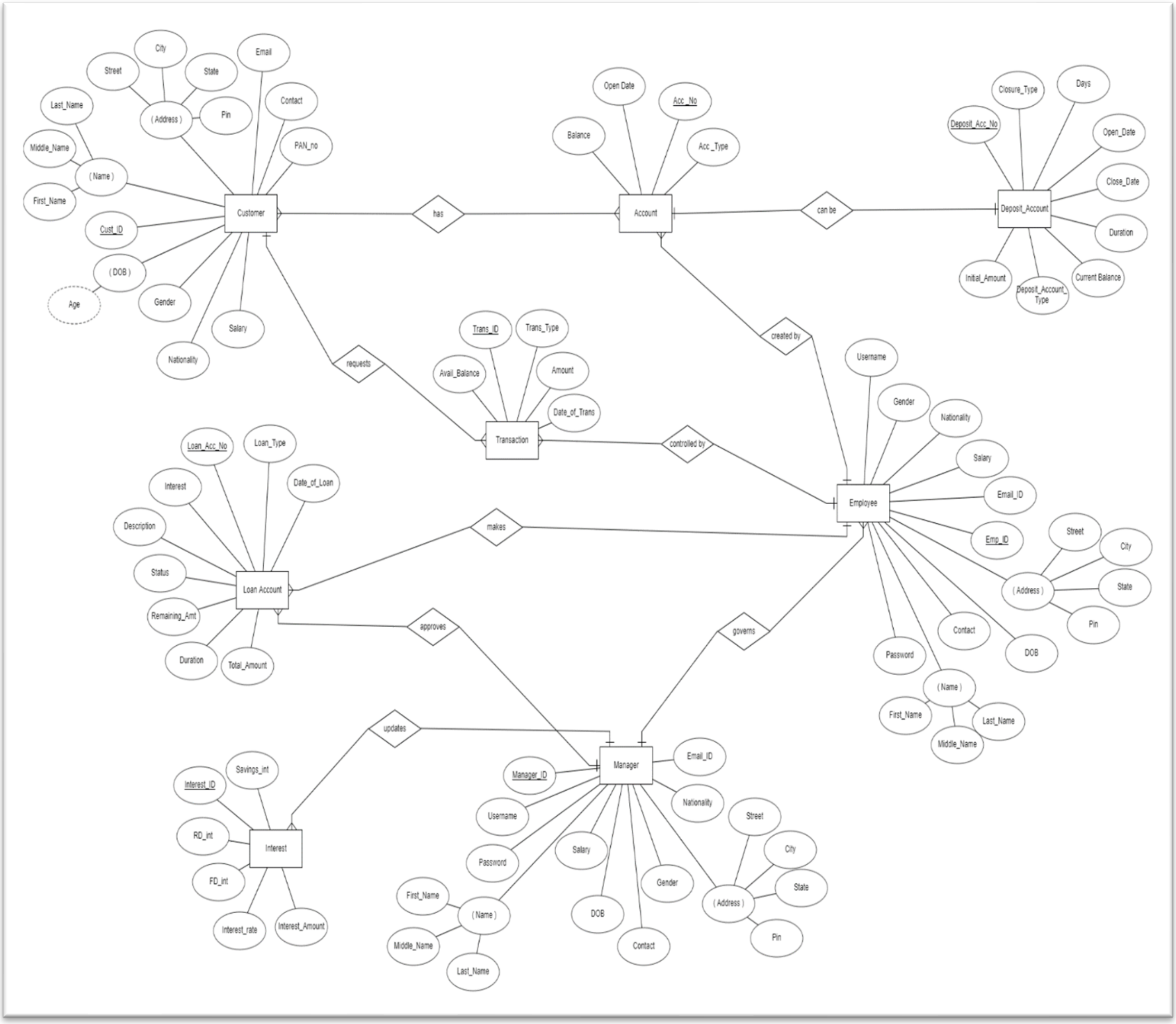
UPDATED ERD AND RELATIONAL SCHEMA

ER DIAGRAM (HAND -DRAWN):



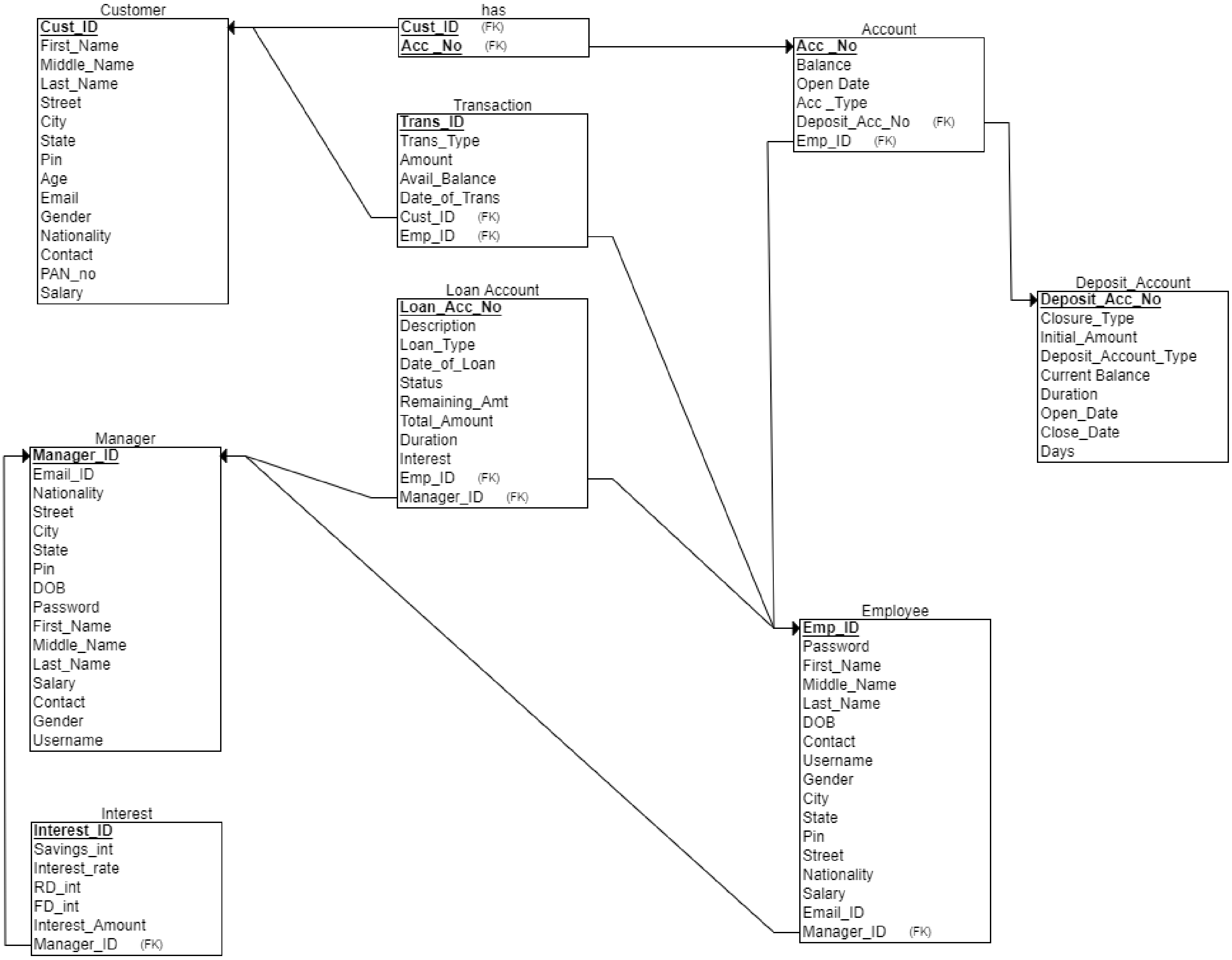
BANK MANAGEMENT SYSTEM

ER DIAGRAM (ERDPlus):



BANK MANAGEMENT SYSTEM

RELATIONAL SCHEMA (ERDPlus):



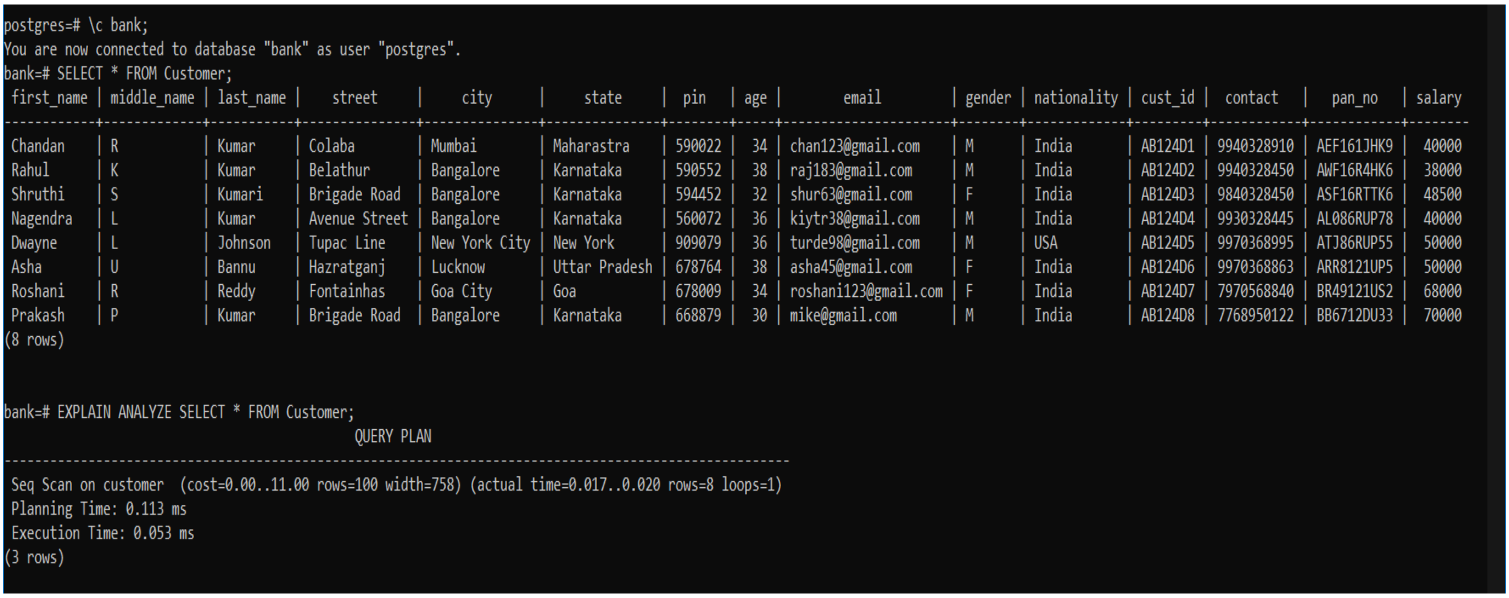
BANK MANAGEMENT SYSTEM

Complete working model of the Database Application

Queries Execution and Performance Analysis with Query Execution Plan

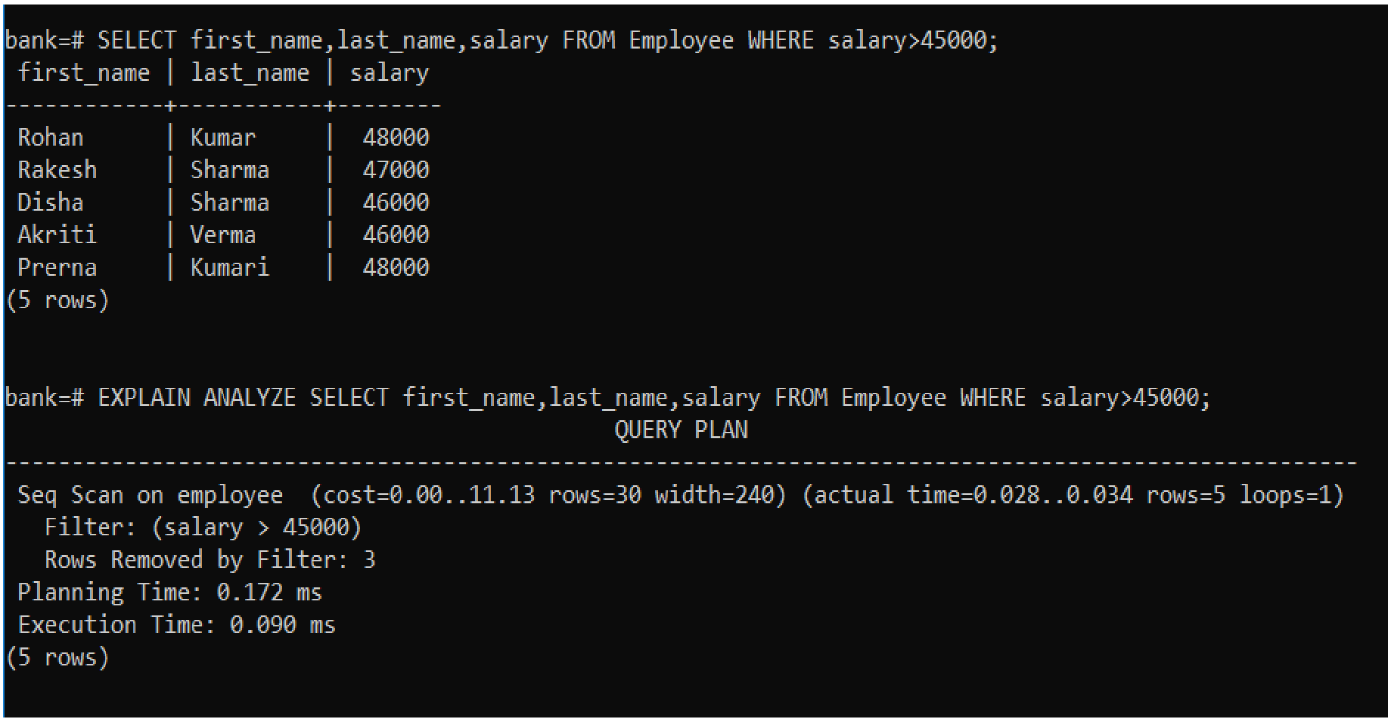
Simple Queries:

1. Retrieve the complete details of the customers.



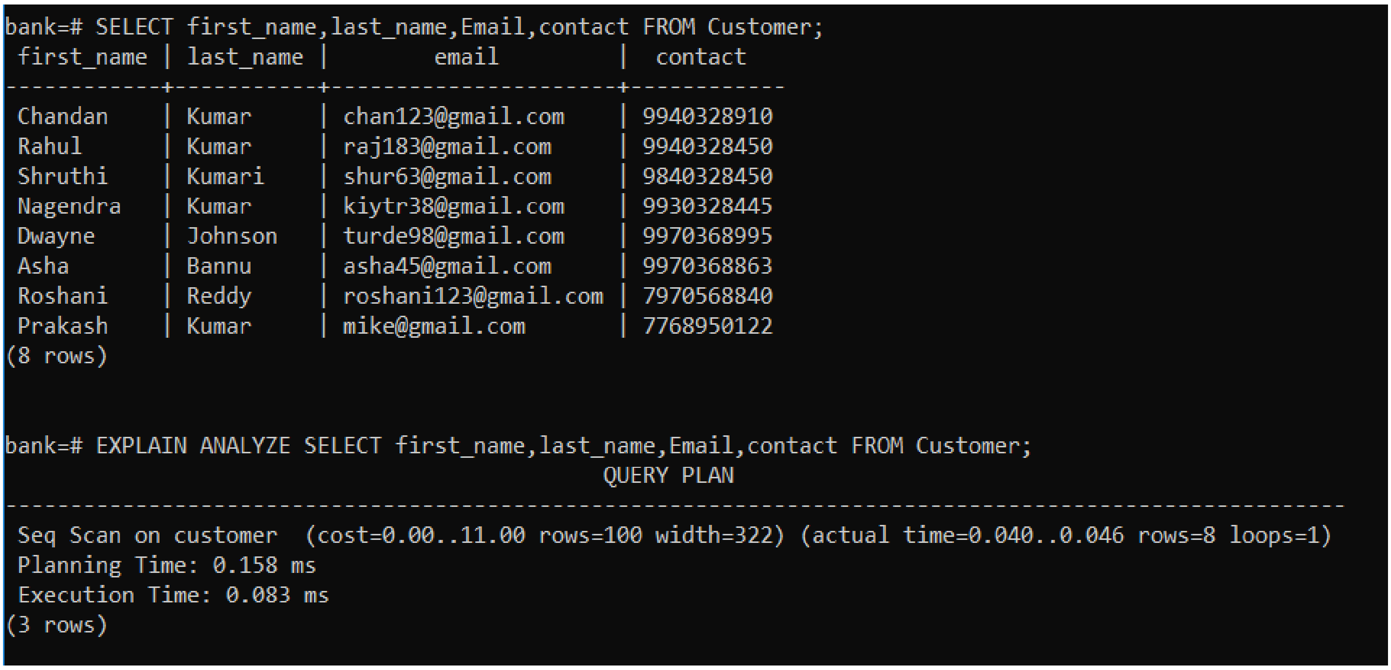
```
postgres=# \c bank;
You are now connected to database "bank" as user "postgres".
bank=# SELECT * FROM Customer;
 first_name | middle_name | last_name |    street     |     city      |     state      |  pin   | age |        email         | gender | nationality | cust_id |  contact   |   pan_no   | salary
------------+-------------+-----------+---------------+---------------+----------------+--------+-----+----------------------+--------+-------------+---------+------------+------------+--------
 Chandan    | R           | Kumar     | Colaba        | Mumbai        | Maharastra     | 590022 |  34 | chan123@gmail.com    | M      | India       | AB124D1 | 9940328910 | AEF161JHK9 |  40000
 Rahul      | K           | Kumar     | Belathur      | Bangalore     | Karnataka      | 590552 |  38 | raj183@gmail.com     | M      | India       | AB124D2 | 9940328450 | AWF16R4HK6 |  38000
 Shruthi    | S           | Kumari    | Brigade Road  | Bangalore     | Karnataka      | 594452 |  32 | shur63@gmail.com     | F      | India       | AB124D3 | 9840328450 | ASF16RTTK6 |  48500
 Nagendra   | L           | Kumar     | Avenue Street | Bangalore     | Karnataka      | 560072 |  36 | kiytr38@gmail.com    | M      | India       | AB124D4 | 9930328445 | AL086RUP78 |  40000
 Dwayne     | L           | Johnson   | Tupac Line    | New York City | New York       | 909079 |  36 | turde98@gmail.com    | M      | USA         | AB124D5 | 9970368995 | ATJ86RUP55 |  50000
 Asha       | U           | Bannu     | Hazratganj    | Lucknow       | Uttar Pradesh  | 678764 |  38 | asha45@gmail.com     | F      | India       | AB124D6 | 9970368863 | ARR8121UP5 |  50000
 Roshani    | R           | Reddy     | Fontainhas    | Goa City      | Goa            | 678009 |  34 | roshani123@gmail.com | F      | India       | AB124D7 | 7970568840 | BR49121US2 |  68000
 Prakash    | P           | Kumar     | Brigade Road  | Bangalore     | Karnataka      | 668879 |  30 | mike@gmail.com       | M      | India       | AB124D8 | 7768950122 | BB6712DU33 |  70000
(8 rows)


bank=# EXPLAIN ANALYZE SELECT * FROM Customer;
                                                QUERY PLAN
---------------------------------------------------------------------------------------------------------
 Seq Scan on customer  (cost=0.00..11.00 rows=100 width=758) (actual time=0.017..0.020 rows=8 loops=1)
 Planning Time: 0.113 ms
 Execution Time: 0.053 ms
(3 rows)
```

2. Retrieve the Names of the Employee whose salary is greater than 45000.



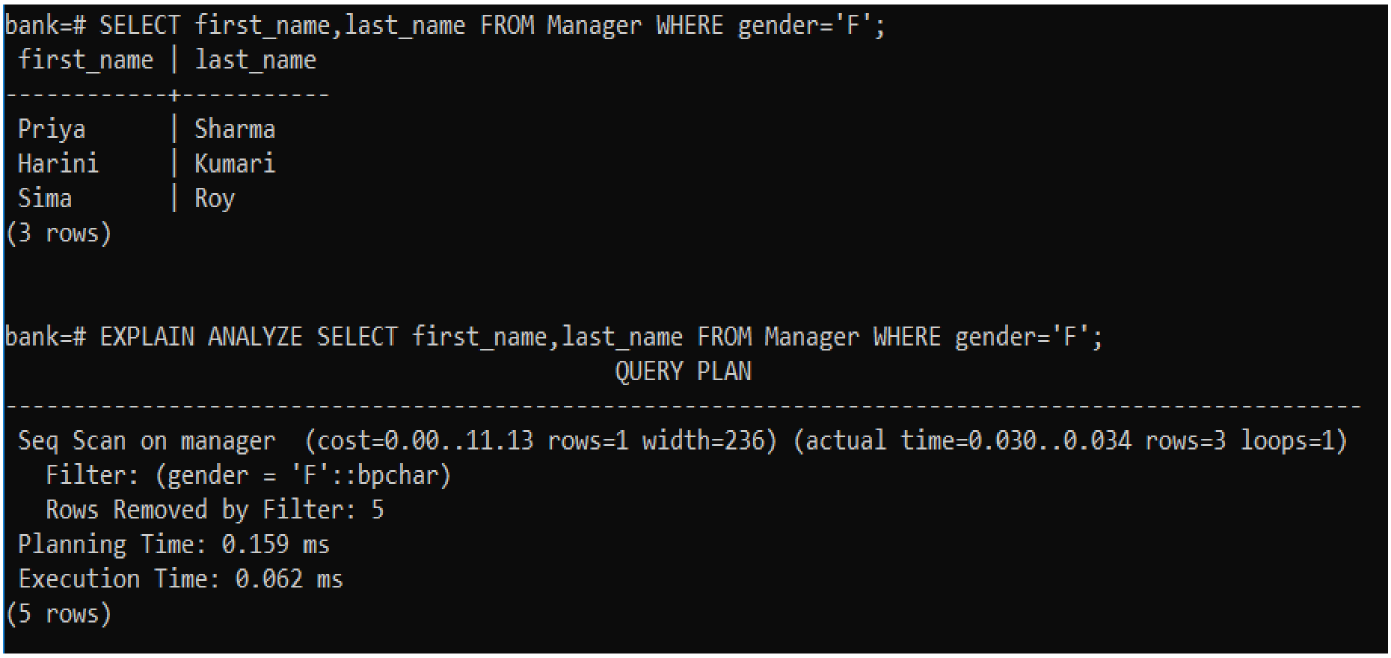
```
bank=# SELECT first_name,last_name,salary FROM Employee WHERE salary>45000;
 first_name | last_name | salary
------------+-----------+--------
 Rohan      | Kumar     |  48000
 Rakesh     | Sharma    |  47000
 Disha      | Sharma    |  46000
 Akriti     | Verma     |  46000
 Prerna     | Kumari    |  48000
(5 rows)


bank=# EXPLAIN ANALYZE SELECT first_name,last_name,salary FROM Employee WHERE salary>45000;
                                               QUERY PLAN
---------------------------------------------------------------------------------------------------------
 Seq Scan on employee  (cost=0.00..11.13 rows=30 width=240) (actual time=0.028..0.034 rows=5 loops=1)
   Filter: (salary > 45000)
   Rows Removed by Filter: 3
 Planning Time: 0.172 ms
 Execution Time: 0.090 ms
(5 rows)
```

3. Retrieve the E-mail and contact details of the customers of the Bank.



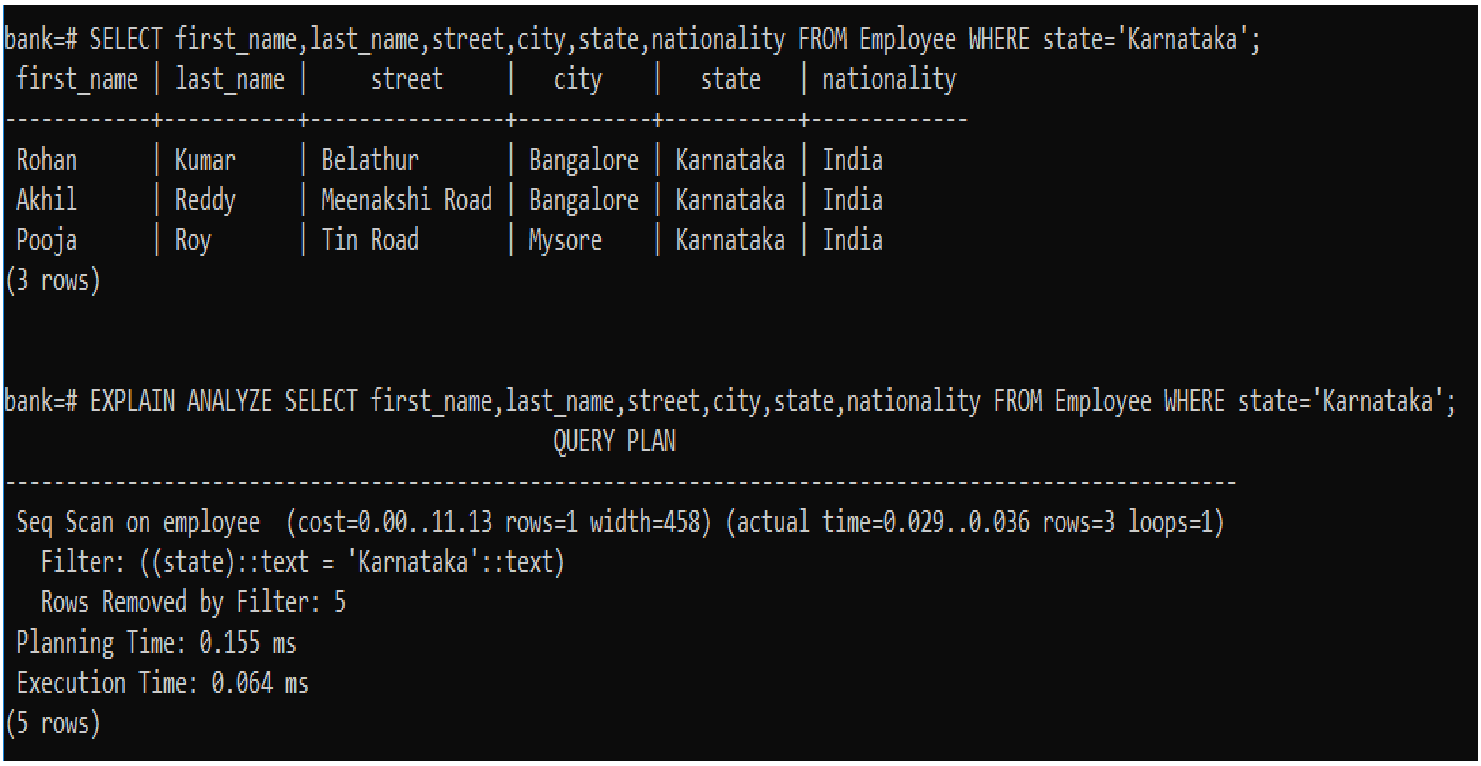
```
bank=# SELECT first_name,last_name,Email,contact FROM Customer;
 first_name | last_name |         email         |  contact
------------+-----------+-----------------------+------------
 Chandan    | Kumar     | chan123@gmail.com     | 9940328910
 Rahul      | Kumar     | raj183@gmail.com      | 9940328450
 Shruthi    | Kumari    | shur63@gmail.com      | 9840328450
 Nagendra   | Kumar     | kiytr38@gmail.com     | 9930328445
 Dwayne     | Johnson   | turde98@gmail.com     | 9970368995
 Asha       | Bannu     | asha45@gmail.com      | 9970368863
 Roshani    | Reddy     | roshani123@gmail.com  | 7970568840
 Prakash    | Kumar     | mike@gmail.com        | 7768950122
(8 rows)


bank=# EXPLAIN ANALYZE SELECT first_name,last_name,Email,contact FROM Customer;
                                              QUERY PLAN
-----------------------------------------------------------------------------------------------------------
 Seq Scan on customer  (cost=0.00..11.00 rows=100 width=322) (actual time=0.040..0.046 rows=8 loops=1)
 Planning Time: 0.158 ms
 Execution Time: 0.083 ms
(3 rows)
```

4. List all the Female Managers of the Bank.



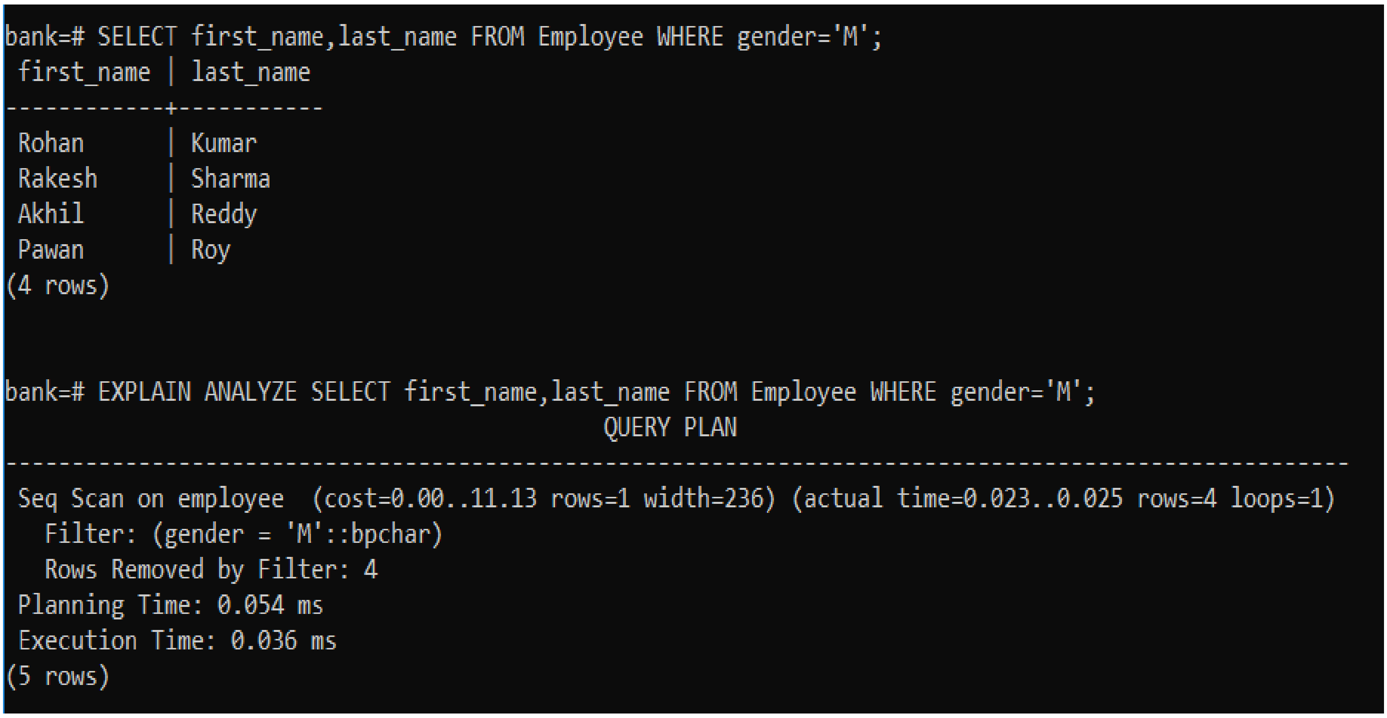
```
bank=# SELECT first_name,last_name FROM Manager WHERE gender='F';
 first_name | last_name
------------+-----------
 Priya      | Sharma
 Harini     | Kumari
 Sima       | Roy
(3 rows)


bank=# EXPLAIN ANALYZE SELECT first_name,last_name FROM Manager WHERE gender='F';
                                              QUERY PLAN
-----------------------------------------------------------------------------------------------------------
 Seq Scan on manager  (cost=0.00..11.13 rows=1 width=236) (actual time=0.030..0.034 rows=3 loops=1)
   Filter: (gender = 'F'::bpchar)
   Rows Removed by Filter: 5
 Planning Time: 0.159 ms
 Execution Time: 0.062 ms
(5 rows)
```

5. Retrieve the Name and Address of all the Employee's whose is from Karnataka State.



```
bank=# SELECT first_name,last_name,street,city,state,nationality FROM Employee WHERE state='Karnataka';
 first_name | last_name |     street     |   city    |   state   | nationality
------------+-----------+----------------+-----------+-----------+-------------
 Rohan      | Kumar     | Belathur       | Bangalore | Karnataka | India
 Akhil      | Reddy     | Meenakshi Road | Bangalore | Karnataka | India
 Pooja      | Roy       | Tin Road       | Mysore    | Karnataka | India
(3 rows)


bank=# EXPLAIN ANALYZE SELECT first_name,last_name,street,city,state,nationality FROM Employee WHERE state='Karnataka';
                                             QUERY PLAN
-----------------------------------------------------------------------------------------------------
 Seq Scan on employee  (cost=0.00..11.13 rows=1 width=458) (actual time=0.029..0.036 rows=3 loops=1)
   Filter: ((state)::text = 'Karnataka'::text)
   Rows Removed by Filter: 5
 Planning Time: 0.155 ms
 Execution Time: 0.064 ms
(5 rows)
```

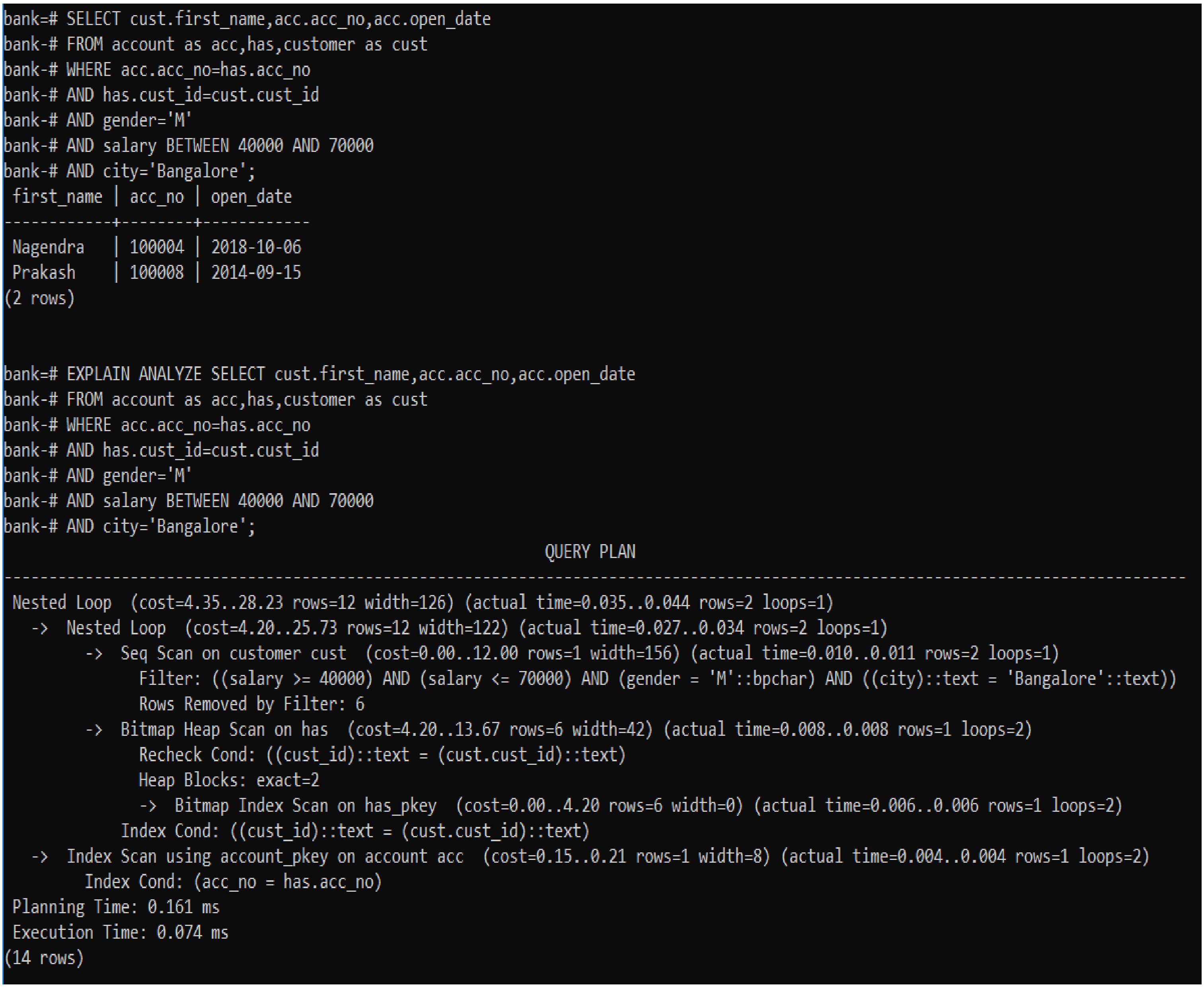
6. List all the Male Employee's of the Bank.



```
bank=# SELECT first_name,last_name FROM Employee WHERE gender='M';
 first_name | last_name
------------+-----------
 Rohan      | Kumar
 Rakesh     | Sharma
 Akhil      | Reddy
 Pawan      | Roy
(4 rows)


bank=# EXPLAIN ANALYZE SELECT first_name,last_name FROM Employee WHERE gender='M';
                                             QUERY PLAN
-----------------------------------------------------------------------------------------------------
 Seq Scan on employee  (cost=0.00..11.13 rows=1 width=236) (actual time=0.023..0.025 rows=4 loops=1)
   Filter: (gender = 'M'::bpchar)
   Rows Removed by Filter: 4
 Planning Time: 0.054 ms
 Execution Time: 0.036 ms
(5 rows)
```

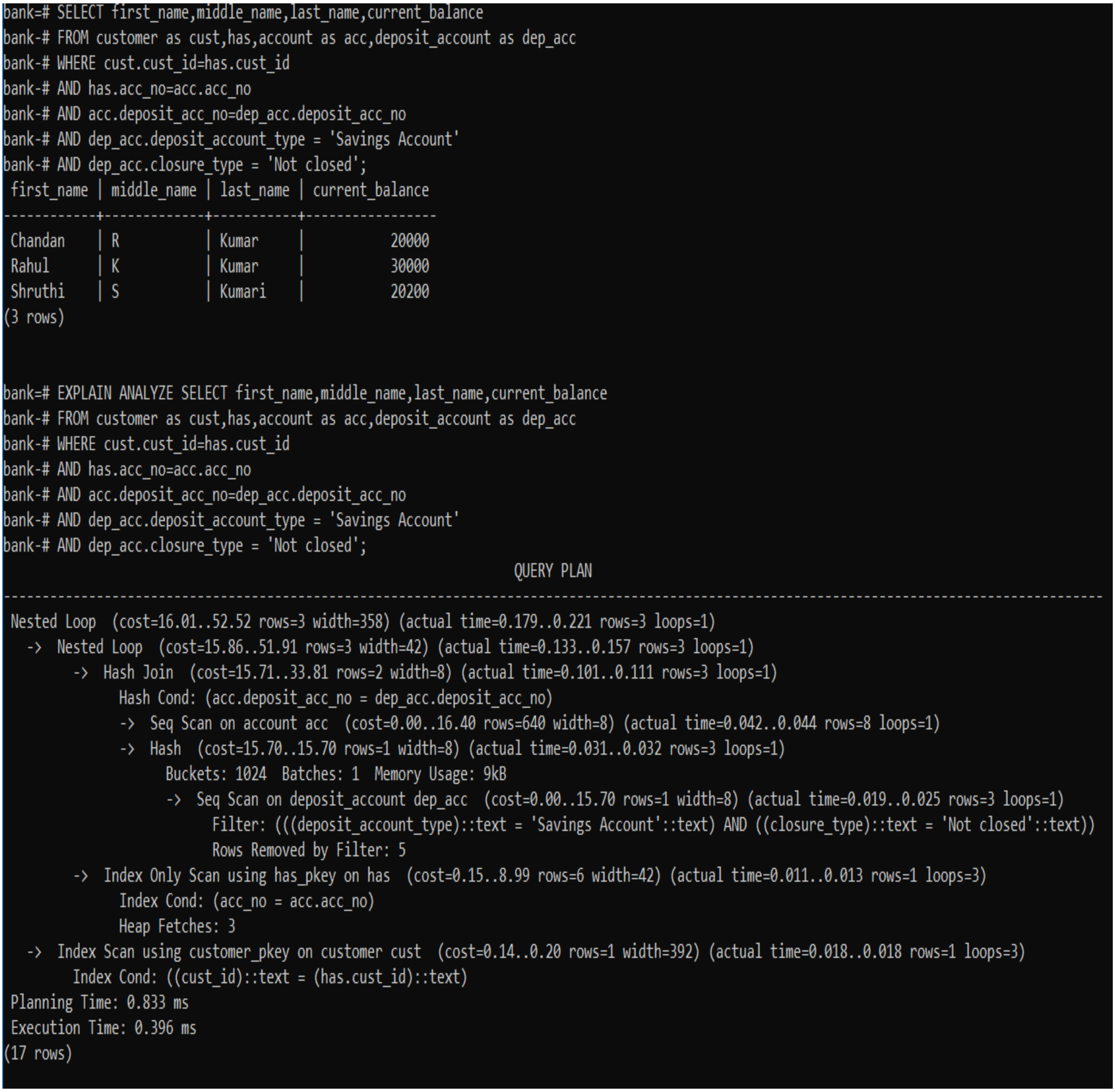
Complex Queries:

1. Retrieve the Opening dates of the accounts of Male Customers staying in Bangalore and having salaries in the range of 40000-70000.



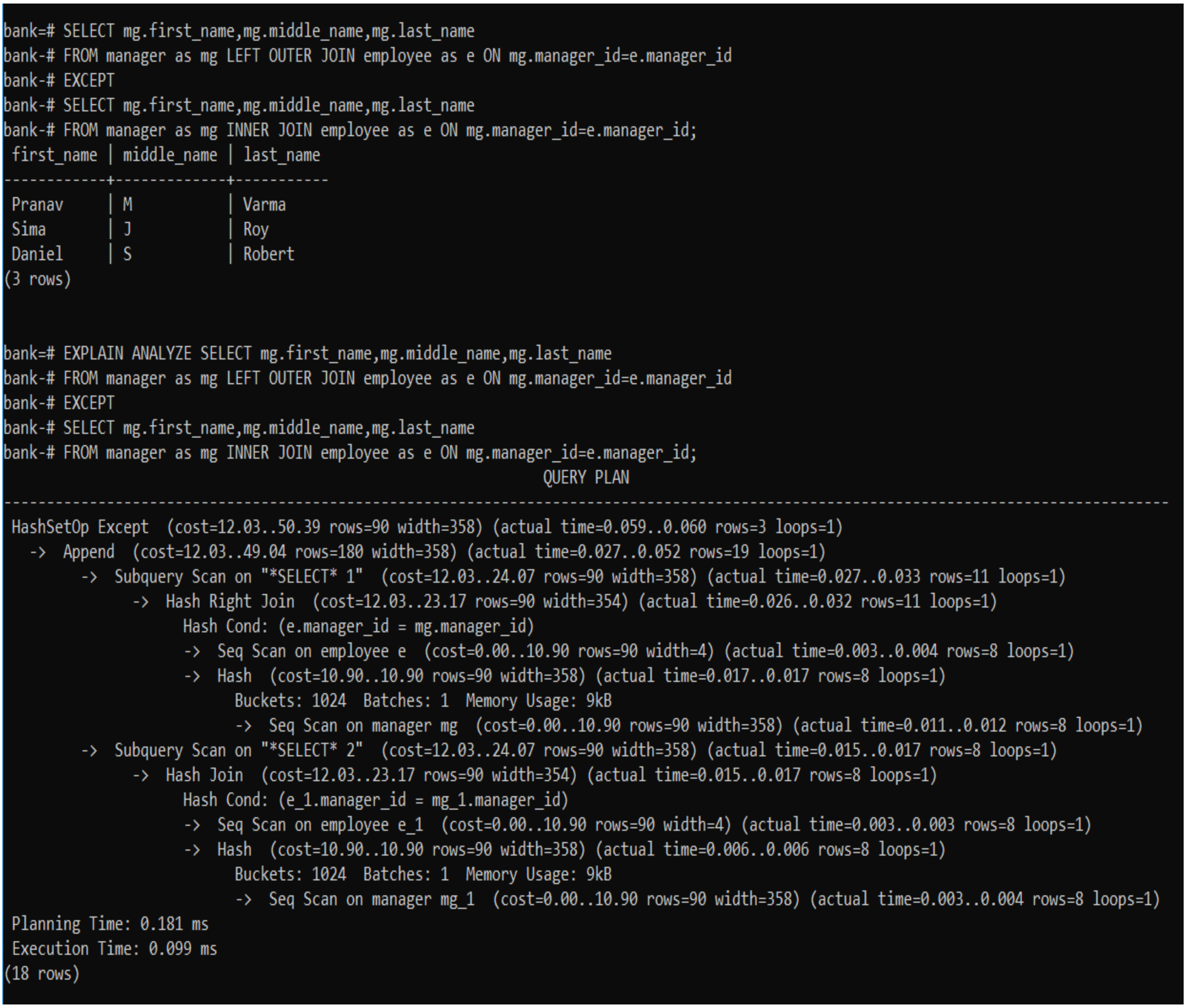
```
bank=# SELECT cust.first_name,acc.acc_no,acc.open_date
bank-# FROM account as acc,has,customer as cust
bank-# WHERE acc.acc_no=has.acc_no
bank-# AND has.cust_id=cust.cust_id
bank-# AND gender='M'
bank-# AND salary BETWEEN 40000 AND 70000
bank-# AND city='Bangalore';
 first_name | acc_no | open_date
------------+--------+------------
 Nagendra   | 100004 | 2018-10-06
 Prakash    | 100008 | 2014-09-15
(2 rows)


bank=# EXPLAIN ANALYZE SELECT cust.first_name,acc.acc_no,acc.open_date
bank-# FROM account as acc,has,customer as cust
bank-# WHERE acc.acc_no=has.acc_no
bank-# AND has.cust_id=cust.cust_id
bank-# AND gender='M'
bank-# AND salary BETWEEN 40000 AND 70000
bank-# AND city='Bangalore';
                                                            QUERY PLAN
-----------------------------------------------------------------------------------------------------------------------------------
 Nested Loop  (cost=4.35..28.23 rows=12 width=126) (actual time=0.035..0.044 rows=2 loops=1)
   ->  Nested Loop  (cost=4.20..25.73 rows=12 width=122) (actual time=0.027..0.034 rows=2 loops=1)
         ->  Seq Scan on customer cust  (cost=0.00..12.00 rows=1 width=156) (actual time=0.010..0.011 rows=2 loops=1)
               Filter: ((salary >= 40000) AND (salary <= 70000) AND (gender = 'M'::bpchar) AND ((city)::text = 'Bangalore'::text))
               Rows Removed by Filter: 6
         ->  Bitmap Heap Scan on has  (cost=4.20..13.67 rows=6 width=42) (actual time=0.008..0.008 rows=1 loops=2)
               Recheck Cond: ((cust_id)::text = (cust.cust_id)::text)
               Heap Blocks: exact=2
               ->  Bitmap Index Scan on has_pkey  (cost=0.00..4.20 rows=6 width=0) (actual time=0.006..0.006 rows=1 loops=2)
             Index Cond: ((cust_id)::text = (cust.cust_id)::text)
   ->  Index Scan using account_pkey on account acc  (cost=0.15..0.21 rows=1 width=8) (actual time=0.004..0.004 rows=1 loops=2)
         Index Cond: (acc_no = has.acc_no)
 Planning Time: 0.161 ms
 Execution Time: 0.074 ms
(14 rows)
```

2. Retrieve the Name and Current Balance of the customers whose Savings Account are open.



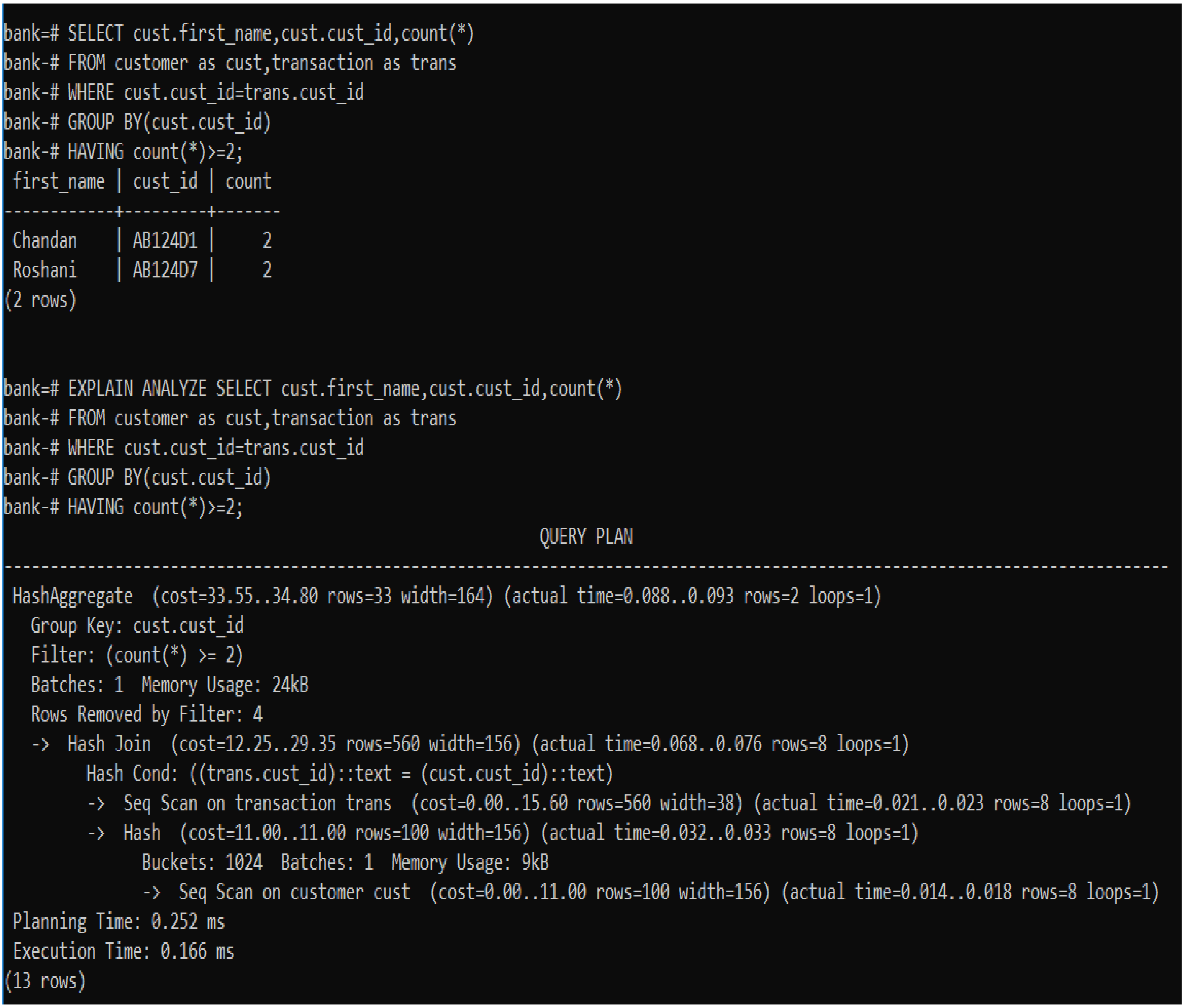
```
bank=# SELECT first_name,middle_name,last_name,current_balance
bank-# FROM customer as cust,has,account as acc,deposit_account as dep_acc
bank-# WHERE cust.cust_id=has.cust_id
bank-# AND has.acc_no=acc.acc_no
bank-# AND acc.deposit_acc_no=dep_acc.deposit_acc_no
bank-# AND dep_acc.deposit_account_type = 'Savings Account'
bank-# AND dep_acc.closure_type = 'Not closed';
 first_name | middle_name | last_name | current_balance
------------+-------------+-----------+-----------------
 Chandan    | R           | Kumar     |           20000
 Rahul      | K           | Kumar     |           30000
 Shruthi    | S           | Kumari    |           20200
(3 rows)


bank=# EXPLAIN ANALYZE SELECT first_name,middle_name,last_name,current_balance
bank-# FROM customer as cust,has,account as acc,deposit_account as dep_acc
bank-# WHERE cust.cust_id=has.cust_id
bank-# AND has.acc_no=acc.acc_no
bank-# AND acc.deposit_acc_no=dep_acc.deposit_acc_no
bank-# AND dep_acc.deposit_account_type = 'Savings Account'
bank-# AND dep_acc.closure_type = 'Not closed';
                                                                 QUERY PLAN
--------------------------------------------------------------------------------------------------------------------------------------------
 Nested Loop  (cost=16.01..52.52 rows=3 width=358) (actual time=0.179..0.221 rows=3 loops=1)
   ->  Nested Loop  (cost=15.86..51.91 rows=3 width=42) (actual time=0.133..0.157 rows=3 loops=1)
         ->  Hash Join  (cost=15.71..33.81 rows=2 width=8) (actual time=0.101..0.111 rows=3 loops=1)
               Hash Cond: (acc.deposit_acc_no = dep_acc.deposit_acc_no)
               ->  Seq Scan on account acc  (cost=0.00..16.40 rows=640 width=8) (actual time=0.042..0.044 rows=8 loops=1)
               ->  Hash  (cost=15.70..15.70 rows=1 width=8) (actual time=0.031..0.032 rows=3 loops=1)
                     Buckets: 1024  Batches: 1  Memory Usage: 9kB
                     ->  Seq Scan on deposit_account dep_acc  (cost=0.00..15.70 rows=1 width=8) (actual time=0.019..0.025 rows=3 loops=1)
                           Filter: (((deposit_account_type)::text = 'Savings Account'::text) AND ((closure_type)::text = 'Not closed'::text))
                           Rows Removed by Filter: 5
         ->  Index Only Scan using has_pkey on has  (cost=0.15..8.99 rows=6 width=42) (actual time=0.011..0.013 rows=1 loops=3)
               Index Cond: (acc_no = acc.acc_no)
               Heap Fetches: 3
   ->  Index Scan using customer_pkey on customer cust  (cost=0.14..0.20 rows=1 width=392) (actual time=0.018..0.018 rows=1 loops=3)
         Index Cond: ((cust_id)::text = (has.cust_id)::text)
 Planning Time: 0.833 ms
 Execution Time: 0.396 ms
(17 rows)
```

3. Find the Names of all the Managers who do not Supervise any Employee (without using nested query).



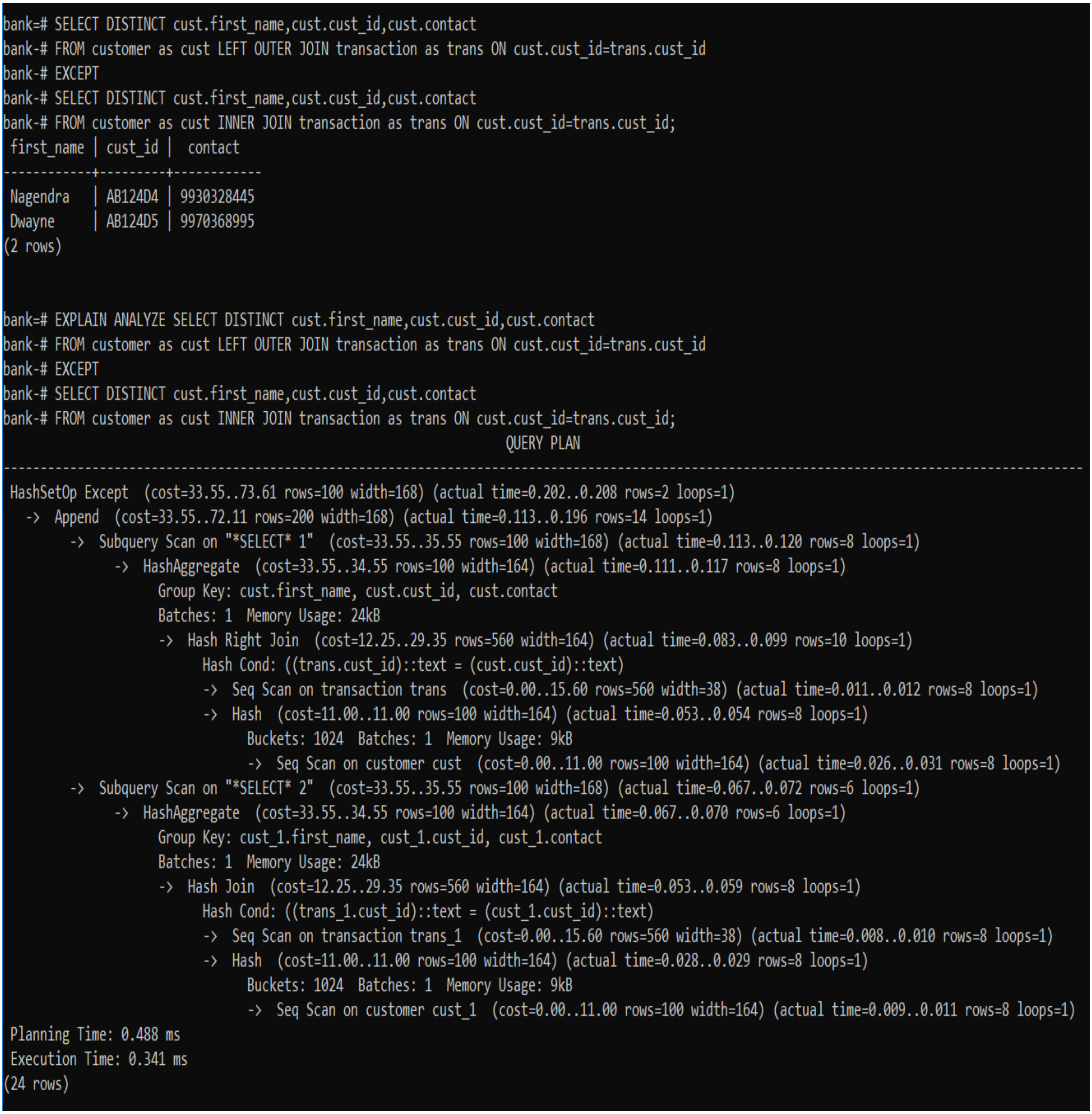
```
bank=# SELECT mg.first_name,mg.middle_name,mg.last_name
bank-# FROM manager as mg LEFT OUTER JOIN employee as e ON mg.manager_id=e.manager_id
bank-# EXCEPT
bank-# SELECT mg.first_name,mg.middle_name,mg.last_name
bank-# FROM manager as mg INNER JOIN employee as e ON mg.manager_id=e.manager_id;
 first_name | middle_name | last_name
------------+-------------+-----------
 Pranav     | M           | Varma
 Sima       | J           | Roy
 Daniel     | S           | Robert
(3 rows)


bank=# EXPLAIN ANALYZE SELECT mg.first_name,mg.middle_name,mg.last_name
bank-# FROM manager as mg LEFT OUTER JOIN employee as e ON mg.manager_id=e.manager_id
bank-# EXCEPT
bank-# SELECT mg.first_name,mg.middle_name,mg.last_name
bank-# FROM manager as mg INNER JOIN employee as e ON mg.manager_id=e.manager_id;
                                                               QUERY PLAN
-------------------------------------------------------------------------------------------------------------------------------------------
 HashSetOp Except  (cost=12.03..50.39 rows=90 width=358) (actual time=0.059..0.060 rows=3 loops=1)
   ->  Append  (cost=12.03..49.04 rows=180 width=358) (actual time=0.027..0.052 rows=19 loops=1)
         ->  Subquery Scan on "*SELECT* 1"  (cost=12.03..24.07 rows=90 width=358) (actual time=0.027..0.033 rows=11 loops=1)
               ->  Hash Right Join  (cost=12.03..23.17 rows=90 width=354) (actual time=0.026..0.032 rows=11 loops=1)
                     Hash Cond: (e.manager_id = mg.manager_id)
                     ->  Seq Scan on employee e  (cost=0.00..10.90 rows=90 width=4) (actual time=0.003..0.004 rows=8 loops=1)
                     ->  Hash  (cost=10.90..10.90 rows=90 width=358) (actual time=0.017..0.017 rows=8 loops=1)
                           Buckets: 1024  Batches: 1  Memory Usage: 9kB
                           ->  Seq Scan on manager mg  (cost=0.00..10.90 rows=90 width=358) (actual time=0.011..0.012 rows=8 loops=1)
         ->  Subquery Scan on "*SELECT* 2"  (cost=12.03..24.07 rows=90 width=358) (actual time=0.015..0.017 rows=8 loops=1)
               ->  Hash Join  (cost=12.03..23.17 rows=90 width=354) (actual time=0.015..0.017 rows=8 loops=1)
                     Hash Cond: (e_1.manager_id = mg_1.manager_id)
                     ->  Seq Scan on employee e_1  (cost=0.00..10.90 rows=90 width=4) (actual time=0.003..0.003 rows=8 loops=1)
                     ->  Hash  (cost=10.90..10.90 rows=90 width=358) (actual time=0.006..0.006 rows=8 loops=1)
                           Buckets: 1024  Batches: 1  Memory Usage: 9kB
                           ->  Seq Scan on manager mg_1  (cost=0.00..10.90 rows=90 width=358) (actual time=0.003..0.004 rows=8 loops=1)
 Planning Time: 0.181 ms
 Execution Time: 0.099 ms
(18 rows)
```

4. Find the total no. of Transactions that have been committed by the customers of the bank, who have committed atleast two transactions.



```
bank=# SELECT cust.first_name,cust.cust_id,count(*)
bank-# FROM customer as cust,transaction as trans
bank-# WHERE cust.cust_id=trans.cust_id
bank-# GROUP BY(cust.cust_id)
bank-# HAVING count(*)>=2;
 first_name | cust_id | count
------------+---------+-------
 Chandan    | AB124D1 |     2
 Roshani    | AB124D7 |     2
(2 rows)



bank=# EXPLAIN ANALYZE SELECT cust.first_name,cust.cust_id,count(*)
bank-# FROM customer as cust,transaction as trans
bank-# WHERE cust.cust_id=trans.cust_id
bank-# GROUP BY(cust.cust_id)
bank-# HAVING count(*)>=2;
                                                          QUERY PLAN
-------------------------------------------------------------------------------------------------------------------------------
 HashAggregate  (cost=33.55..34.80 rows=33 width=164) (actual time=0.088..0.093 rows=2 loops=1)
   Group Key: cust.cust_id
   Filter: (count(*) >= 2)
   Batches: 1  Memory Usage: 24kB
   Rows Removed by Filter: 4
   ->  Hash Join  (cost=12.25..29.35 rows=560 width=156) (actual time=0.068..0.076 rows=8 loops=1)
         Hash Cond: ((trans.cust_id)::text = (cust.cust_id)::text)
         ->  Seq Scan on transaction trans  (cost=0.00..15.60 rows=560 width=38) (actual time=0.021..0.023 rows=8 loops=1)
         ->  Hash  (cost=11.00..11.00 rows=100 width=156) (actual time=0.032..0.033 rows=8 loops=1)
               Buckets: 1024  Batches: 1  Memory Usage: 9kB
               ->  Seq Scan on customer cust  (cost=0.00..11.00 rows=100 width=156) (actual time=0.014..0.018 rows=8 loops=1)
 Planning Time: 0.252 ms
 Execution Time: 0.166 ms
(13 rows)
```

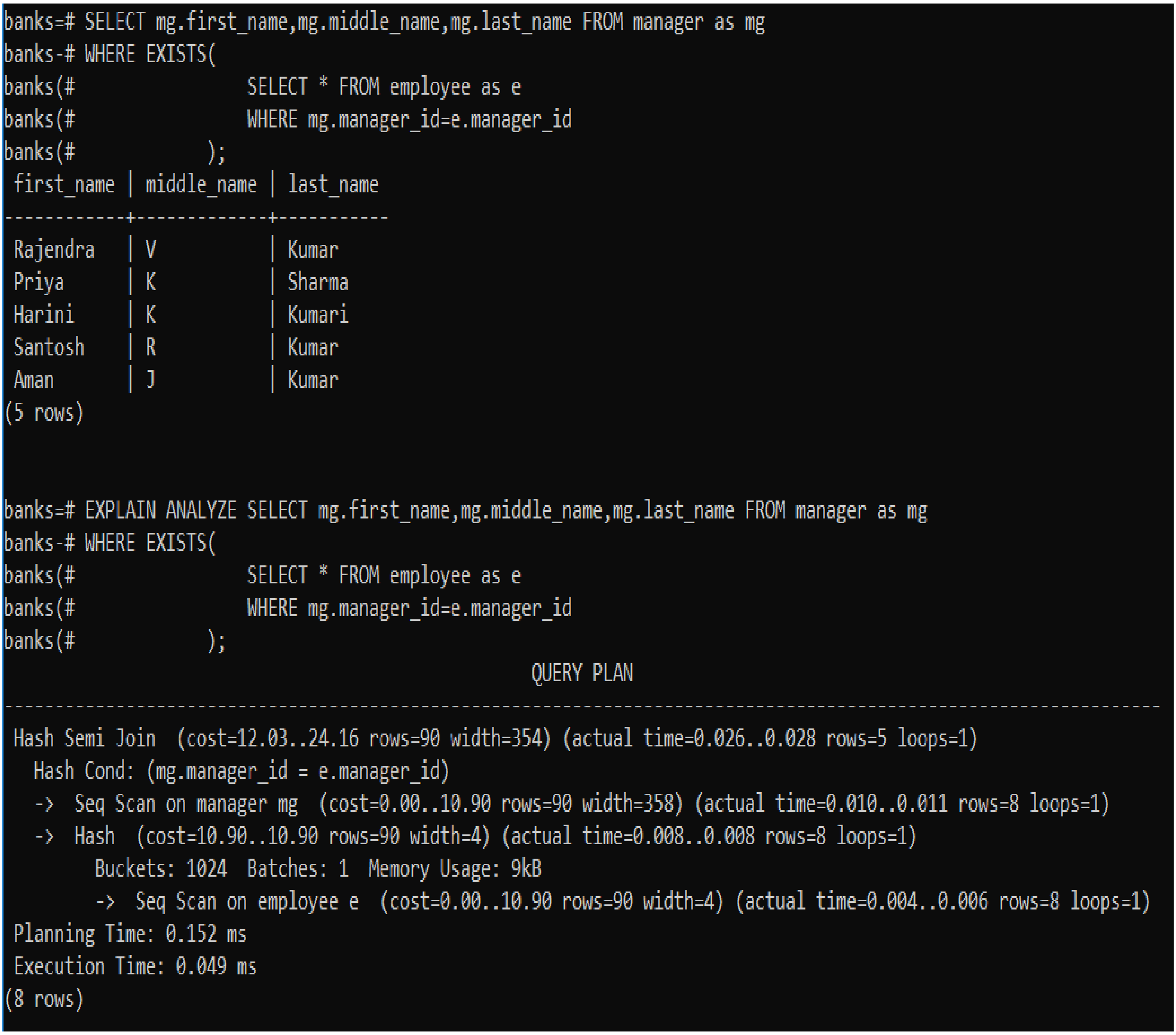
5. Obtain the Contact details of all the Customers who are yet to commit a transaction.



```
bank=# SELECT DISTINCT cust.first_name,cust.cust_id,cust.contact
bank-# FROM customer as cust LEFT OUTER JOIN transaction as trans ON cust.cust_id=trans.cust_id
bank-# EXCEPT
bank-# SELECT DISTINCT cust.first_name,cust.cust_id,cust.contact
bank-# FROM customer as cust INNER JOIN transaction as trans ON cust.cust_id=trans.cust_id;
 first_name | cust_id |  contact
------------+---------+------------
 Nagendra   | AB124D4 | 9930328445
 Dwayne     | AB124D5 | 9970368995
(2 rows)



bank=# EXPLAIN ANALYZE SELECT DISTINCT cust.first_name,cust.cust_id,cust.contact
bank-# FROM customer as cust LEFT OUTER JOIN transaction as trans ON cust.cust_id=trans.cust_id
bank-# EXCEPT
bank-# SELECT DISTINCT cust.first_name,cust.cust_id,cust.contact
bank-# FROM customer as cust INNER JOIN transaction as trans ON cust.cust_id=trans.cust_id;
                                                                    QUERY PLAN
-------------------------------------------------------------------------------------------------------------------------------------------------
 HashSetOp Except  (cost=33.55..73.61 rows=100 width=168) (actual time=0.202..0.208 rows=2 loops=1)
   ->  Append  (cost=33.55..72.11 rows=200 width=168) (actual time=0.113..0.196 rows=14 loops=1)
         ->  Subquery Scan on "*SELECT* 1"  (cost=33.55..35.55 rows=100 width=168) (actual time=0.113..0.120 rows=8 loops=1)
               ->  HashAggregate  (cost=33.55..34.55 rows=100 width=164) (actual time=0.111..0.117 rows=8 loops=1)
                     Group Key: cust.first_name, cust.cust_id, cust.contact
                     Batches: 1  Memory Usage: 24kB
                     ->  Hash Right Join  (cost=12.25..29.35 rows=560 width=164) (actual time=0.083..0.099 rows=10 loops=1)
                           Hash Cond: ((trans.cust_id)::text = (cust.cust_id)::text)
                           ->  Seq Scan on transaction trans  (cost=0.00..15.60 rows=560 width=38) (actual time=0.011..0.012 rows=8 loops=1)
                           ->  Hash  (cost=11.00..11.00 rows=100 width=164) (actual time=0.053..0.054 rows=8 loops=1)
                                 Buckets: 1024  Batches: 1  Memory Usage: 9kB
                                 ->  Seq Scan on customer cust  (cost=0.00..11.00 rows=100 width=164) (actual time=0.026..0.031 rows=8 loops=1)
         ->  Subquery Scan on "*SELECT* 2"  (cost=33.55..35.55 rows=100 width=168) (actual time=0.067..0.072 rows=6 loops=1)
               ->  HashAggregate  (cost=33.55..34.55 rows=100 width=164) (actual time=0.067..0.070 rows=6 loops=1)
                     Group Key: cust_1.first_name, cust_1.cust_id, cust_1.contact
                     Batches: 1  Memory Usage: 24kB
                     ->  Hash Join  (cost=12.25..29.35 rows=560 width=164) (actual time=0.053..0.059 rows=8 loops=1)
                           Hash Cond: ((trans_1.cust_id)::text = (cust_1.cust_id)::text)
                           ->  Seq Scan on transaction trans_1  (cost=0.00..15.60 rows=560 width=38) (actual time=0.008..0.010 rows=8 loops=1)
                           ->  Hash  (cost=11.00..11.00 rows=100 width=164) (actual time=0.028..0.029 rows=8 loops=1)
                                 Buckets: 1024  Batches: 1  Memory Usage: 9kB
                                 ->  Seq Scan on customer cust_1  (cost=0.00..11.00 rows=100 width=164) (actual time=0.009..0.011 rows=8 loops=1)
 Planning Time: 0.488 ms
 Execution Time: 0.341 ms
(24 rows)
```

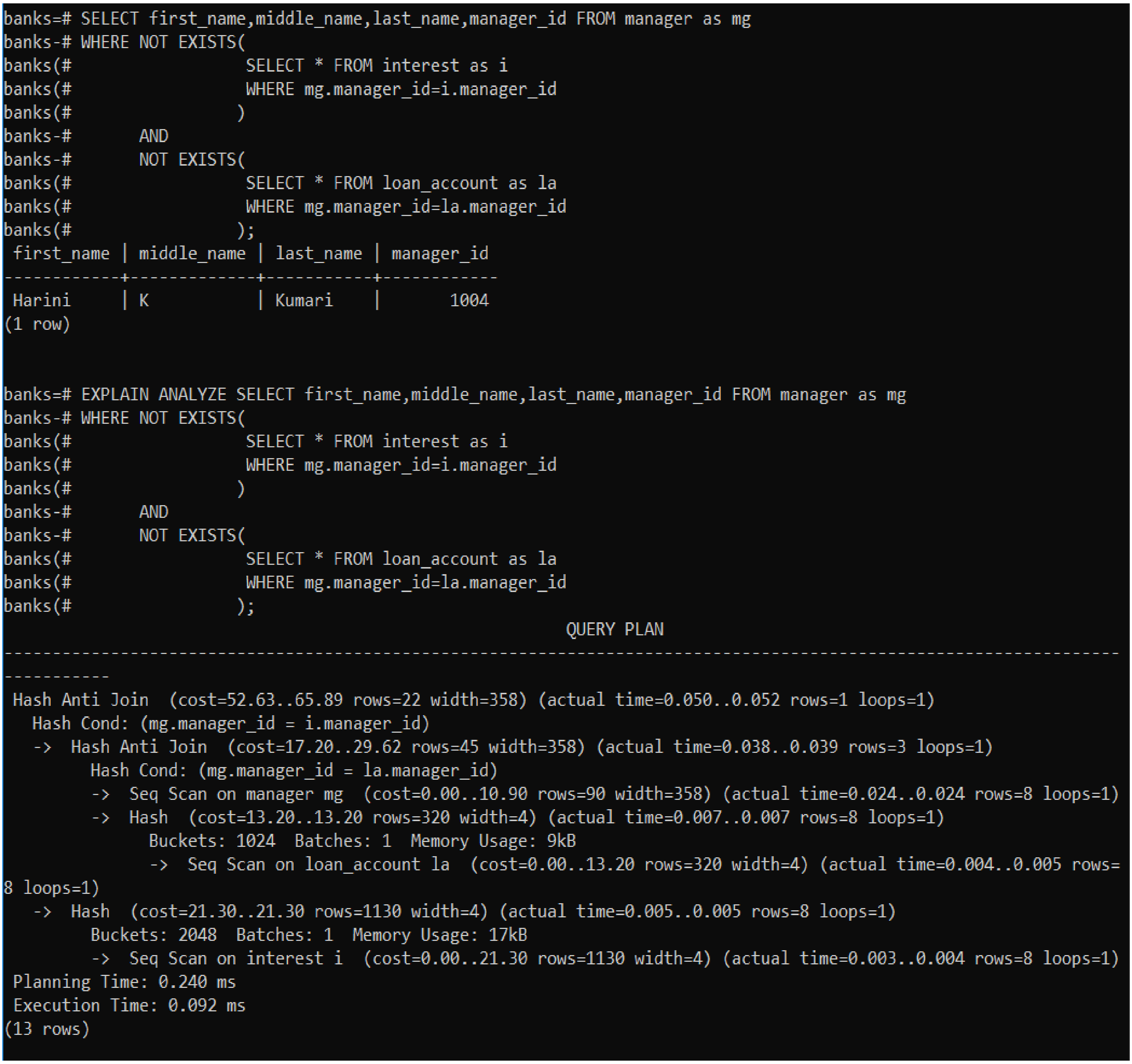
Nested Queries:

1. List the Names of all the Managers who supervise atleast one Employee.



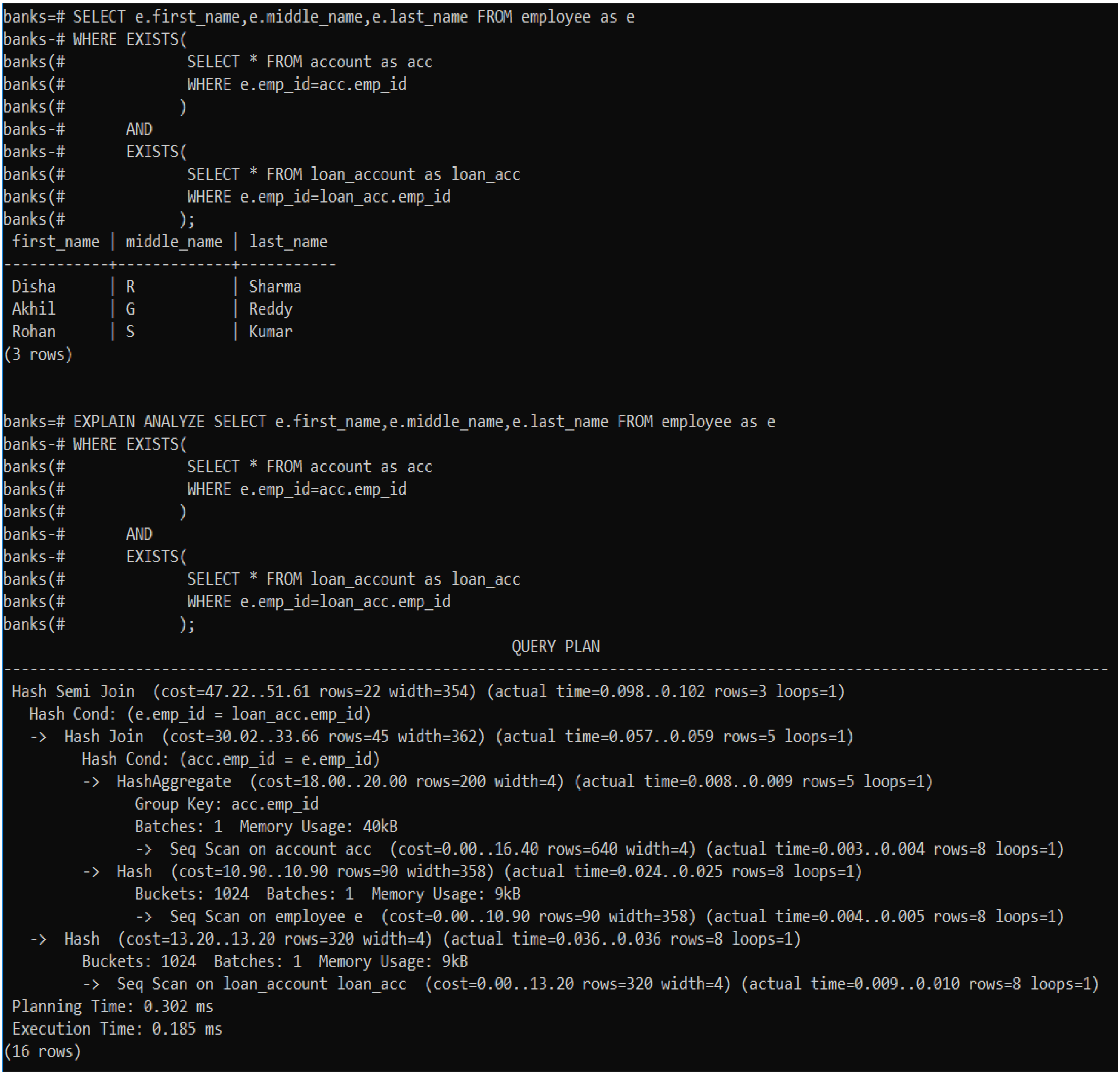
```
banks=# SELECT mg.first_name,mg.middle_name,mg.last_name FROM manager as mg
banks-# WHERE EXISTS(
banks(#                 SELECT * FROM employee as e
banks(#                 WHERE mg.manager_id=e.manager_id
banks(#             );
 first_name | middle_name | last_name
------------+-------------+-----------
 Rajendra   | V           | Kumar
 Priya      | K           | Sharma
 Harini     | K           | Kumari
 Santosh    | R           | Kumar
 Aman       | J           | Kumar
(5 rows)



banks=# EXPLAIN ANALYZE SELECT mg.first_name,mg.middle_name,mg.last_name FROM manager as mg
banks-# WHERE EXISTS(
banks(#                 SELECT * FROM employee as e
banks(#                 WHERE mg.manager_id=e.manager_id
banks(#             );
                                                    QUERY PLAN
------------------------------------------------------------------------------------------------------------------
 Hash Semi Join  (cost=12.03..24.16 rows=90 width=354) (actual time=0.026..0.028 rows=5 loops=1)
   Hash Cond: (mg.manager_id = e.manager_id)
   ->  Seq Scan on manager mg  (cost=0.00..10.90 rows=90 width=358) (actual time=0.010..0.011 rows=8 loops=1)
   ->  Hash  (cost=10.90..10.90 rows=90 width=4) (actual time=0.008..0.008 rows=8 loops=1)
         Buckets: 1024  Batches: 1  Memory Usage: 9kB
         ->  Seq Scan on employee e  (cost=0.00..10.90 rows=90 width=4) (actual time=0.004..0.006 rows=8 loops=1)
 Planning Time: 0.152 ms
 Execution Time: 0.049 ms
(8 rows)
```

2. List the Names and ID of the Managers who do not manage any Loan or Interest Accounts.



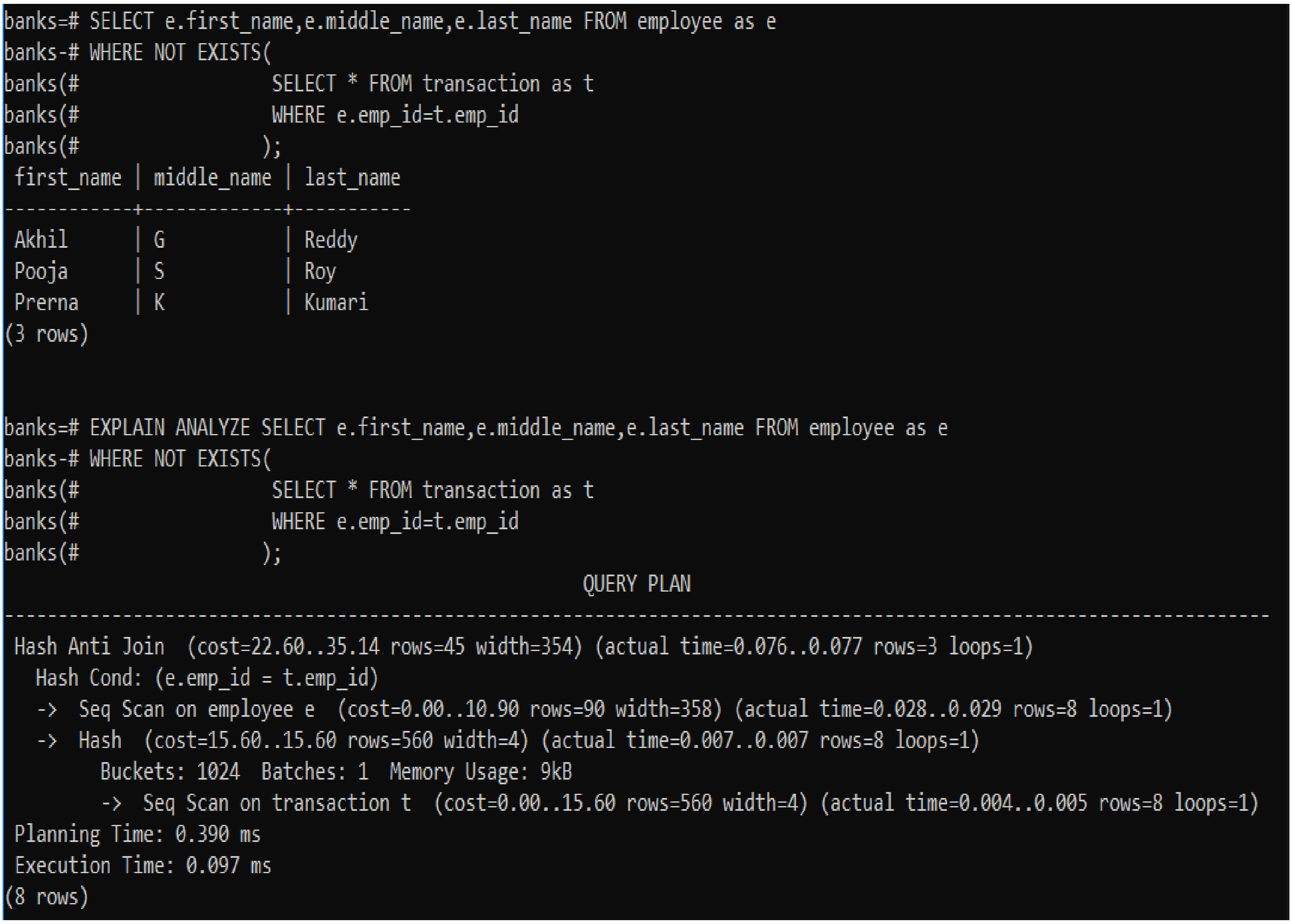
```
banks=# SELECT first_name,middle_name,last_name,manager_id FROM manager as mg
banks-# WHERE NOT EXISTS(
banks(#                  SELECT * FROM interest as i
banks(#                  WHERE mg.manager_id=i.manager_id
banks(#                 )
banks-#       AND
banks-#       NOT EXISTS(
banks(#                  SELECT * FROM loan_account as la
banks(#                  WHERE mg.manager_id=la.manager_id
banks(#                 );
 first_name | middle_name | last_name | manager_id
------------+-------------+-----------+------------
 Harini     | K           | Kumari    |       1004
(1 row)


banks=# EXPLAIN ANALYZE SELECT first_name,middle_name,last_name,manager_id FROM manager as mg
banks-# WHERE NOT EXISTS(
banks(#                  SELECT * FROM interest as i
banks(#                  WHERE mg.manager_id=i.manager_id
banks(#                 )
banks-#       AND
banks-#       NOT EXISTS(
banks(#                  SELECT * FROM loan_account as la
banks(#                  WHERE mg.manager_id=la.manager_id
banks(#                 );
                                                          QUERY PLAN
-------------------------------------------------------------------------------------------------------------------
-----------
 Hash Anti Join  (cost=52.63..65.89 rows=22 width=358) (actual time=0.050..0.052 rows=1 loops=1)
   Hash Cond: (mg.manager_id = i.manager_id)
   ->  Hash Anti Join  (cost=17.20..29.62 rows=45 width=358) (actual time=0.038..0.039 rows=3 loops=1)
         Hash Cond: (mg.manager_id = la.manager_id)
         ->  Seq Scan on manager mg  (cost=0.00..10.90 rows=90 width=358) (actual time=0.024..0.024 rows=8 loops=1)
         ->  Hash  (cost=13.20..13.20 rows=320 width=4) (actual time=0.007..0.007 rows=8 loops=1)
               Buckets: 1024  Batches: 1  Memory Usage: 9kB
               ->  Seq Scan on loan_account la  (cost=0.00..13.20 rows=320 width=4) (actual time=0.004..0.005 rows=
8 loops=1)
   ->  Hash  (cost=21.30..21.30 rows=1130 width=4) (actual time=0.005..0.005 rows=8 loops=1)
         Buckets: 2048  Batches: 1  Memory Usage: 17kB
         ->  Seq Scan on interest i  (cost=0.00..21.30 rows=1130 width=4) (actual time=0.003..0.004 rows=8 loops=1)
 Planning Time: 0.240 ms
 Execution Time: 0.092 ms
(13 rows)
```

3. List the Names of all the Employees who manage atleast one account as well as Loan Account.



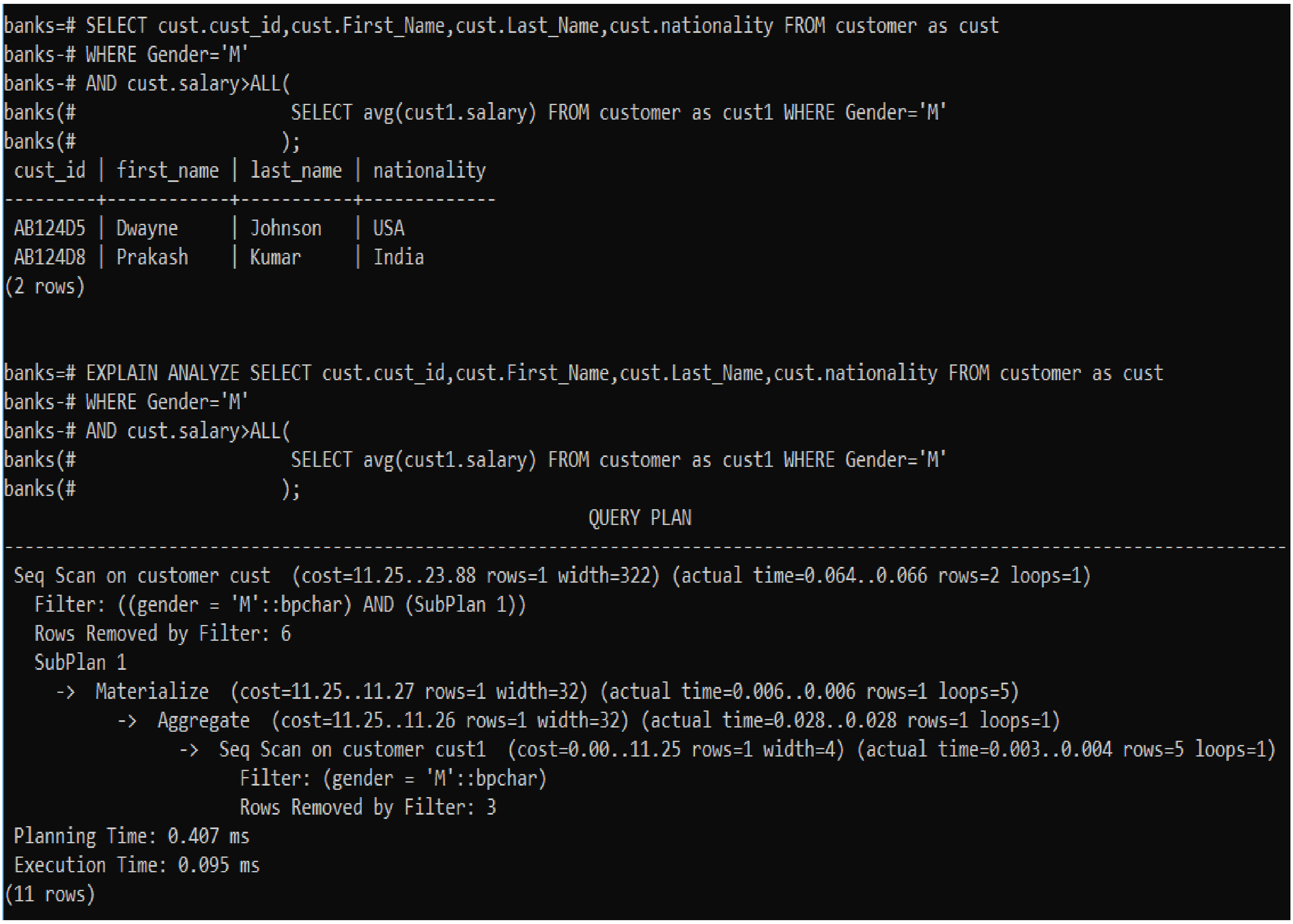
```
banks=# SELECT e.first_name,e.middle_name,e.last_name FROM employee as e
banks-# WHERE EXISTS(
banks(#              SELECT * FROM account as acc
banks(#              WHERE e.emp_id=acc.emp_id
banks(#             )
banks-#       AND
banks-#       EXISTS(
banks(#              SELECT * FROM loan_account as loan_acc
banks(#              WHERE e.emp_id=loan_acc.emp_id
banks(#             );
 first_name | middle_name | last_name
------------+-------------+-----------
 Disha      | R           | Sharma
 Akhil      | G           | Reddy
 Rohan      | S           | Kumar
(3 rows)


banks=# EXPLAIN ANALYZE SELECT e.first_name,e.middle_name,e.last_name FROM employee as e
banks-# WHERE EXISTS(
banks(#              SELECT * FROM account as acc
banks(#              WHERE e.emp_id=acc.emp_id
banks(#             )
banks-#       AND
banks-#       EXISTS(
banks(#              SELECT * FROM loan_account as loan_acc
banks(#              WHERE e.emp_id=loan_acc.emp_id
banks(#             );
                                                         QUERY PLAN
-------------------------------------------------------------------------------------------------------------------------------
 Hash Semi Join  (cost=47.22..51.61 rows=22 width=354) (actual time=0.098..0.102 rows=3 loops=1)
   Hash Cond: (e.emp_id = loan_acc.emp_id)
   ->  Hash Join  (cost=30.02..33.66 rows=45 width=362) (actual time=0.057..0.059 rows=5 loops=1)
         Hash Cond: (acc.emp_id = e.emp_id)
         ->  HashAggregate  (cost=18.00..20.00 rows=200 width=4) (actual time=0.008..0.009 rows=5 loops=1)
               Group Key: acc.emp_id
               Batches: 1  Memory Usage: 40kB
               ->  Seq Scan on account acc  (cost=0.00..16.40 rows=640 width=4) (actual time=0.003..0.004 rows=8 loops=1)
         ->  Hash  (cost=10.90..10.90 rows=90 width=358) (actual time=0.024..0.025 rows=8 loops=1)
               Buckets: 1024  Batches: 1  Memory Usage: 9kB
               ->  Seq Scan on employee e  (cost=0.00..10.90 rows=90 width=358) (actual time=0.004..0.005 rows=8 loops=1)
   ->  Hash  (cost=13.20..13.20 rows=320 width=4) (actual time=0.036..0.036 rows=8 loops=1)
         Buckets: 1024  Batches: 1  Memory Usage: 9kB
         ->  Seq Scan on loan_account loan_acc  (cost=0.00..13.20 rows=320 width=4) (actual time=0.009..0.010 rows=8 loops=1)
 Planning Time: 0.302 ms
 Execution Time: 0.185 ms
(16 rows)
```

4. List the Names of all the Employees who haven't handled any Transaction.



```
banks=# SELECT e.first_name,e.middle_name,e.last_name FROM employee as e
banks-# WHERE NOT EXISTS(
banks(#                  SELECT * FROM transaction as t
banks(#                  WHERE e.emp_id=t.emp_id
banks(#                 );
 first_name | middle_name | last_name
------------+-------------+-----------
 Akhil      | G           | Reddy
 Pooja      | S           | Roy
 Prerna     | K           | Kumari
(3 rows)


banks=# EXPLAIN ANALYZE SELECT e.first_name,e.middle_name,e.last_name FROM employee as e
banks-# WHERE NOT EXISTS(
banks(#                  SELECT * FROM transaction as t
banks(#                  WHERE e.emp_id=t.emp_id
banks(#                 );
                                                    QUERY PLAN
-----------------------------------------------------------------------------------------------------------------------
 Hash Anti Join  (cost=22.60..35.14 rows=45 width=354) (actual time=0.076..0.077 rows=3 loops=1)
   Hash Cond: (e.emp_id = t.emp_id)
   ->  Seq Scan on employee e  (cost=0.00..10.90 rows=90 width=358) (actual time=0.028..0.029 rows=8 loops=1)
   ->  Hash  (cost=15.60..15.60 rows=560 width=4) (actual time=0.007..0.007 rows=8 loops=1)
         Buckets: 1024  Batches: 1  Memory Usage: 9kB
         ->  Seq Scan on transaction t  (cost=0.00..15.60 rows=560 width=4) (actual time=0.004..0.005 rows=8 loops=1)
 Planning Time: 0.390 ms
 Execution Time: 0.097 ms
(8 rows)
```

5. List the Nationality, Name and ID of the Male Customers whose salaries are greater than the average salaries among all the Customers.

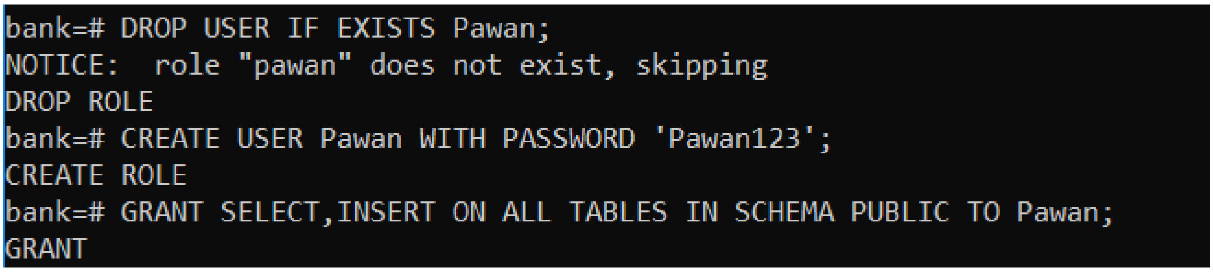


```
banks=# SELECT cust.cust_id,cust.First_Name,cust.Last_Name,cust.nationality FROM customer as cust
banks-# WHERE Gender='M'
banks-# AND cust.salary>ALL(
banks(#                     SELECT avg(cust1.salary) FROM customer as cust1 WHERE Gender='M'
banks(#                    );
 cust_id | first_name | last_name | nationality
---------+------------+-----------+------------
 AB124D5 | Dwayne     | Johnson   | USA
 AB124D8 | Prakash    | Kumar     | India
(2 rows)


banks=# EXPLAIN ANALYZE SELECT cust.cust_id,cust.First_Name,cust.Last_Name,cust.nationality FROM customer as cust
banks-# WHERE Gender='M'
banks-# AND cust.salary>ALL(
banks(#                     SELECT avg(cust1.salary) FROM customer as cust1 WHERE Gender='M'
banks(#                    );
                                                      QUERY PLAN
-------------------------------------------------------------------------------------------------------------------------------
 Seq Scan on customer cust  (cost=11.25..23.88 rows=1 width=322) (actual time=0.064..0.066 rows=2 loops=1)
   Filter: ((gender = 'M'::bpchar) AND (SubPlan 1))
   Rows Removed by Filter: 6
   SubPlan 1
     ->  Materialize  (cost=11.25..11.27 rows=1 width=32) (actual time=0.006..0.006 rows=1 loops=5)
           ->  Aggregate  (cost=11.25..11.26 rows=1 width=32) (actual time=0.028..0.028 rows=1 loops=1)
                 ->  Seq Scan on customer cust1  (cost=0.00..11.25 rows=1 width=4) (actual time=0.003..0.004 rows=5 loops=1)
                       Filter: (gender = 'M'::bpchar)
                       Rows Removed by Filter: 3
 Planning Time: 0.407 ms
 Execution Time: 0.095 ms
(11 rows)
```

Access Privileges:

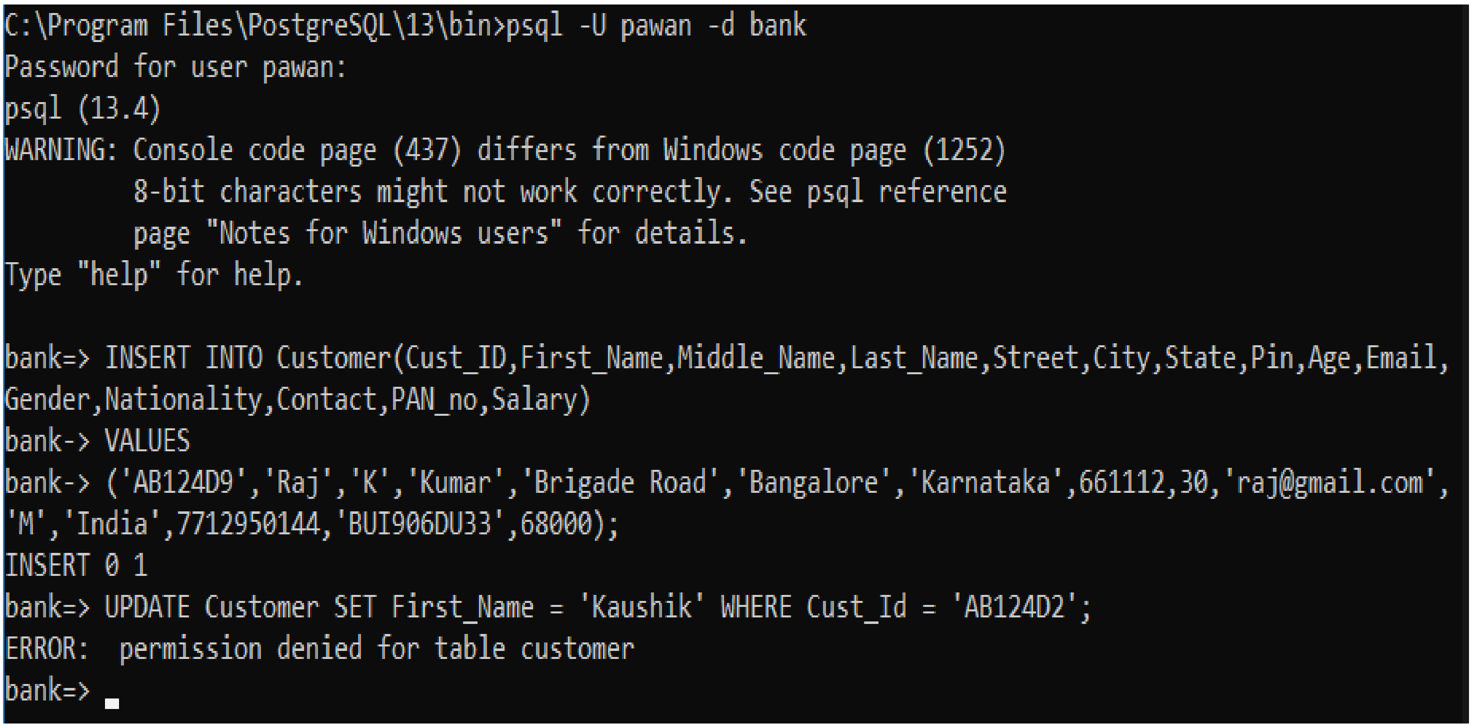
Multiple users with different access privilege levels for different parts of the database should be created.

1. Create a user with permission to the Select and Insert privileges on all Tables for the Bank Database.



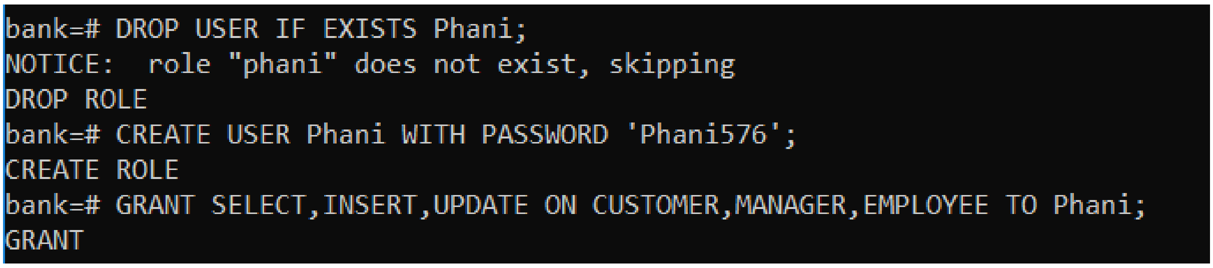
```
bank=# DROP USER IF EXISTS Pawan;
NOTICE:  role "pawan" does not exist, skipping
DROP ROLE
bank=# CREATE USER Pawan WITH PASSWORD 'Pawan123';
CREATE ROLE
bank=# GRANT SELECT,INSERT ON ALL TABLES IN SCHEMA PUBLIC TO Pawan;
GRANT
```

Verification:



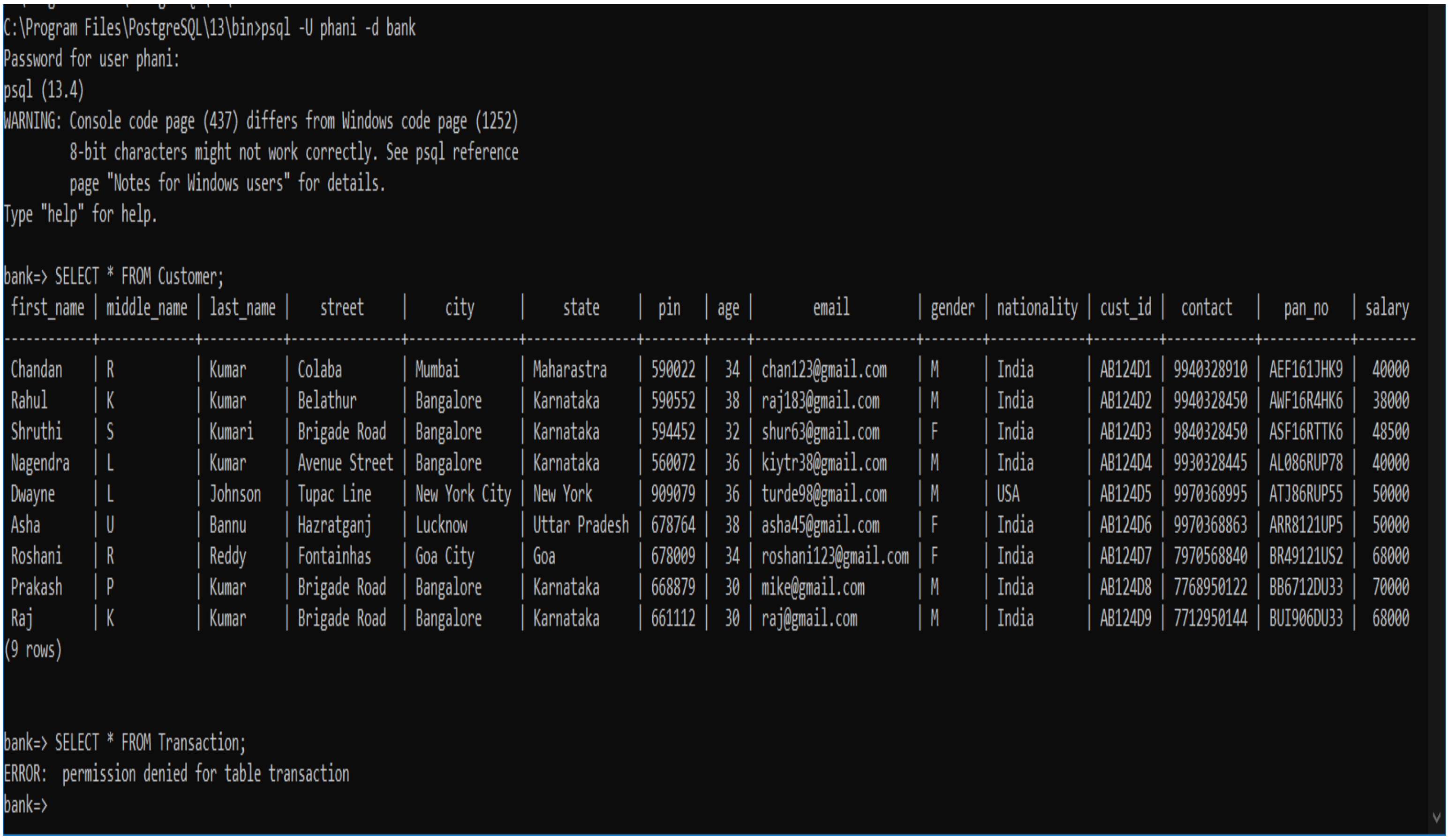
```
C:\Program Files\PostgreSQL\13\bin>psql -U pawan -d bank
Password for user pawan:
psql (13.4)
WARNING: Console code page (437) differs from Windows code page (1252)
         8-bit characters might not work correctly. See psql reference
         page "Notes for Windows users" for details.
Type "help" for help.

bank=> INSERT INTO Customer(Cust_ID,First_Name,Middle_Name,Last_Name,Street,City,State,Pin,Age,Email,
Gender,Nationality,Contact,PAN_no,Salary)
bank-> VALUES
bank-> ('AB124D9','Raj','K','Kumar','Brigade Road','Bangalore','Karnataka',661112,30,'raj@gmail.com',
'M','India',7712950144,'BUI906DU33',68000);
INSERT 0 1
bank=> UPDATE Customer SET First_Name = 'Kaushik' WHERE Cust_Id = 'AB124D2';
ERROR:  permission denied for table customer
bank=>
```

2. Create a user with permission to the Select, Insert and Update privileges on certain Tables for the Bank Database.



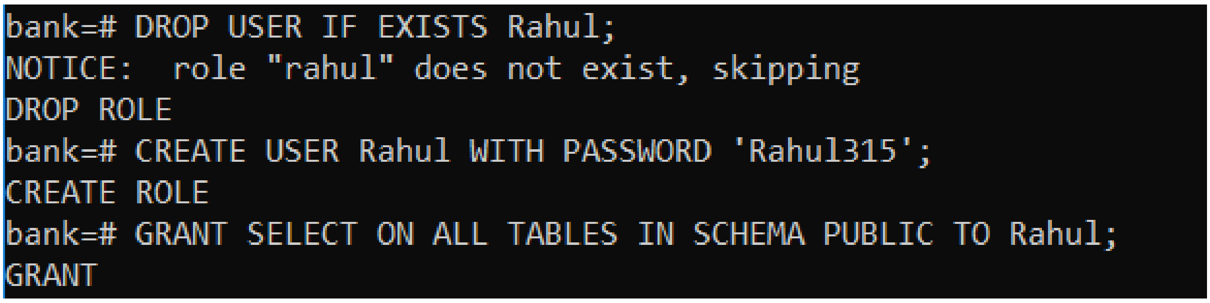
```
bank=# DROP USER IF EXISTS Phani;
NOTICE:  role "phani" does not exist, skipping
DROP ROLE
bank=# CREATE USER Phani WITH PASSWORD 'Phani576';
CREATE ROLE
bank=# GRANT SELECT,INSERT,UPDATE ON CUSTOMER,MANAGER,EMPLOYEE TO Phani;
GRANT
```

Verification:



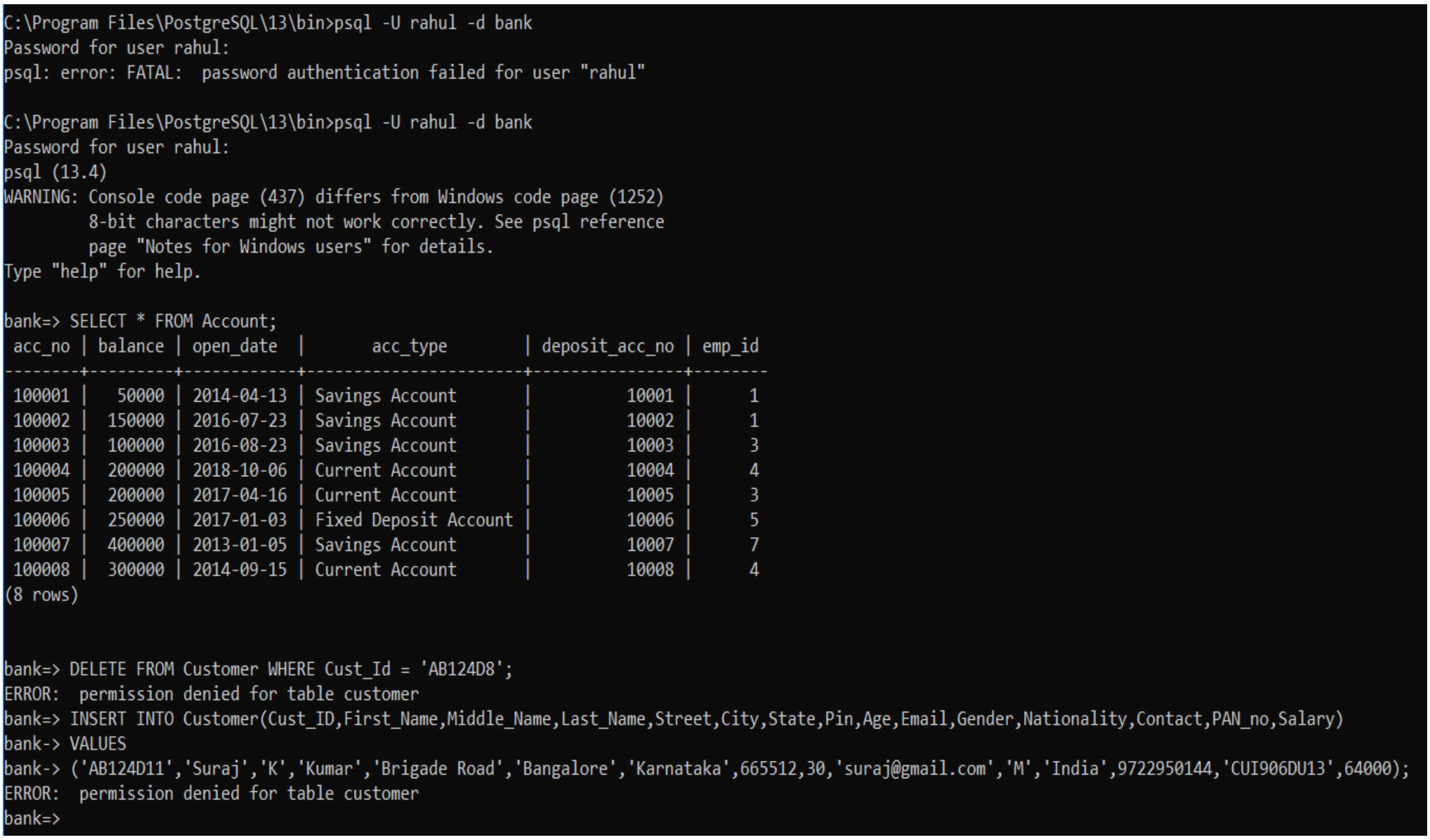
```
C:\Program Files\PostgreSQL\13\bin>psql -U phani -d bank
Password for user phani:
psql (13.4)
WARNING: Console code page (437) differs from Windows code page (1252)
         8-bit characters might not work correctly. See psql reference
         page "Notes for Windows users" for details.
Type "help" for help.

bank=> SELECT * FROM Customer;
 first_name | middle_name | last_name |    street     |     city      |     state     |  pin   | age |        email         | gender | nationality | cust_id |  contact   |   pan_no   | salary
------------+-------------+-----------+---------------+---------------+---------------+--------+-----+----------------------+--------+-------------+---------+------------+------------+--------
 Chandan    | R           | Kumar     | Colaba        | Mumbai        | Maharastra    | 590022 |  34 | chan123@gmail.com    | M      | India       | AB124D1 | 9940328910 | AEF161JHK9 |  40000
 Rahul      | K           | Kumar     | Belathur      | Bangalore     | Karnataka     | 590552 |  38 | raj183@gmail.com     | M      | India       | AB124D2 | 9940328450 | AWF16R4HK6 |  38000
 Shruthi    | S           | Kumari    | Brigade Road  | Bangalore     | Karnataka     | 594452 |  32 | shur63@gmail.com     | F      | India       | AB124D3 | 9840328450 | ASF16RTTK6 |  48500
 Nagendra   | L           | Kumar     | Avenue Street | Bangalore     | Karnataka     | 560072 |  36 | kiytr38@gmail.com    | M      | India       | AB124D4 | 9930328445 | AL086RUP78 |  40000
 Dwayne     | L           | Johnson   | Tupac Line    | New York City | New York      | 909079 |  36 | turde98@gmail.com    | M      | USA         | AB124D5 | 9970368995 | ATJ86RUP55 |  50000
 Asha       | U           | Bannu     | Hazratganj    | Lucknow       | Uttar Pradesh | 678764 |  38 | asha45@gmail.com     | F      | India       | AB124D6 | 9970368863 | ARR8121UP5 |  50000
 Roshani    | R           | Reddy     | Fontainhas    | Goa City      | Goa           | 678009 |  34 | roshani123@gmail.com | F      | India       | AB124D7 | 7970568840 | BR49121US2 |  68000
 Prakash    | P           | Kumar     | Brigade Road  | Bangalore     | Karnataka     | 668879 |  30 | mike@gmail.com       | M      | India       | AB124D8 | 7768950122 | BB6712DU33 |  70000
 Raj        | K           | Kumar     | Brigade Road  | Bangalore     | Karnataka     | 661112 |  30 | raj@gmail.com        | M      | India       | AB124D9 | 7712950144 | BUI906DU33 |  68000
(9 rows)


bank=> SELECT * FROM Transaction;
ERROR:  permission denied for table transaction
bank=>
```

3. Create a user with permission to the Select privilege on all Tables for the Bank Database.



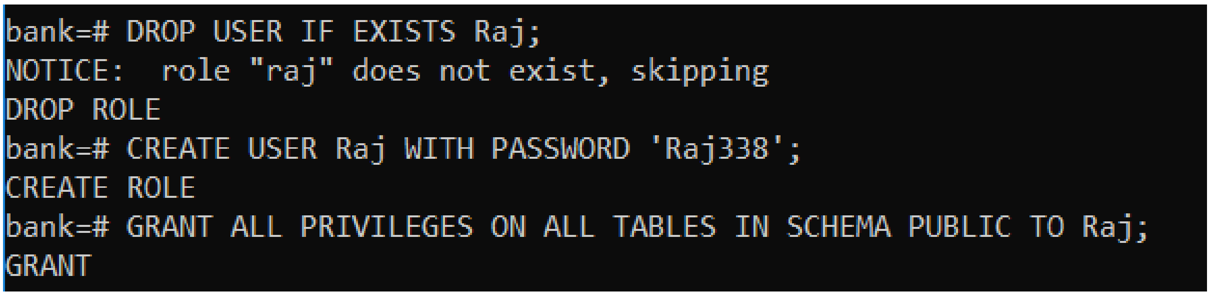
```
bank=# DROP USER IF EXISTS Rahul;
NOTICE:  role "rahul" does not exist, skipping
DROP ROLE
bank=# CREATE USER Rahul WITH PASSWORD 'Rahul315';
CREATE ROLE
bank=# GRANT SELECT ON ALL TABLES IN SCHEMA PUBLIC TO Rahul;
GRANT
```

Verification:



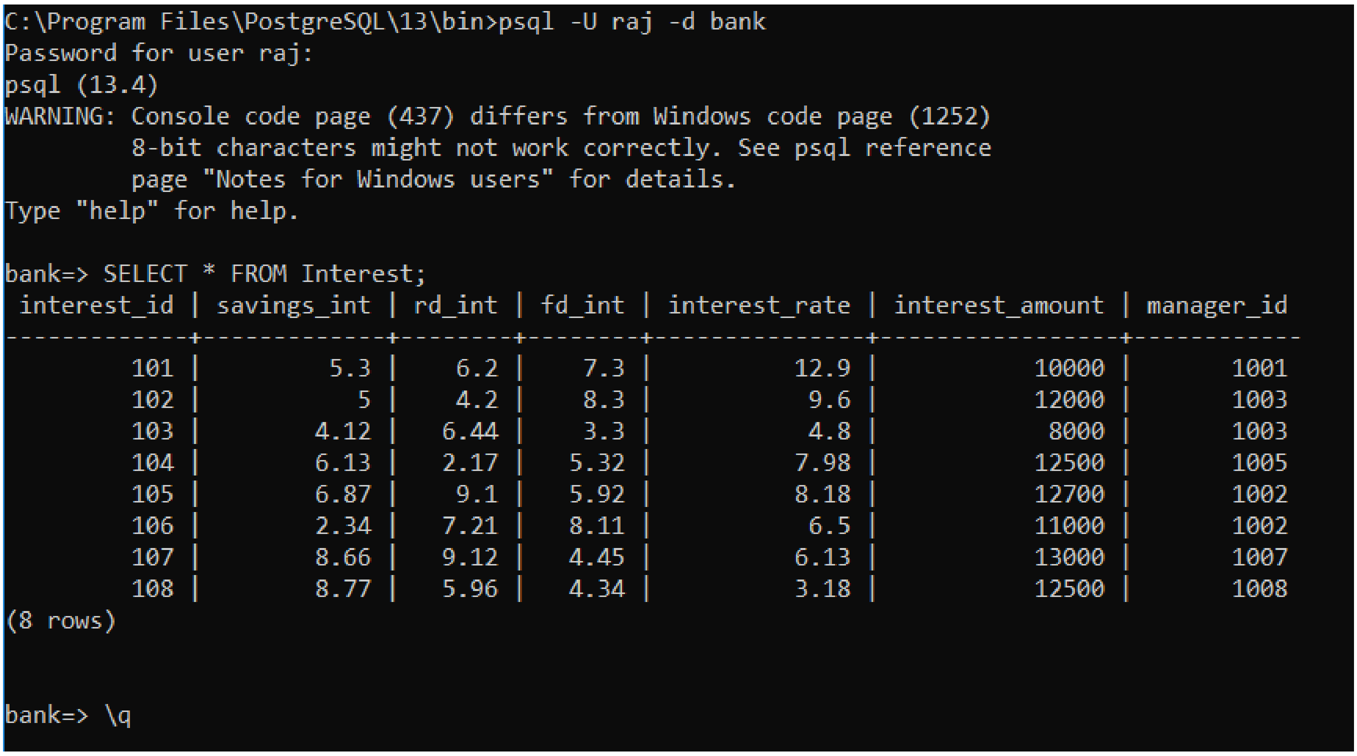
```
C:\Program Files\PostgreSQL\13\bin>psql -U rahul -d bank
Password for user rahul:
psql: error: FATAL:  password authentication failed for user "rahul"

C:\Program Files\PostgreSQL\13\bin>psql -U rahul -d bank
Password for user rahul:
psql (13.4)
WARNING: Console code page (437) differs from Windows code page (1252)
         8-bit characters might not work correctly. See psql reference
         page "Notes for Windows users" for details.
Type "help" for help.

bank=> SELECT * FROM Account;
 acc_no | balance | open_date  |       acc_type        | deposit_acc_no | emp_id
--------+---------+------------+-----------------------+----------------+--------
 100001 |   50000 | 2014-04-13 | Savings Account       |          10001 |      1
 100002 |  150000 | 2016-07-23 | Savings Account       |          10002 |      1
 100003 |  100000 | 2016-08-23 | Savings Account       |          10003 |      3
 100004 |  200000 | 2018-10-06 | Current Account       |          10004 |      4
 100005 |  200000 | 2017-04-16 | Current Account       |          10005 |      3
 100006 |  250000 | 2017-01-03 | Fixed Deposit Account |          10006 |      5
 100007 |  400000 | 2013-01-05 | Savings Account       |          10007 |      7
 100008 |  300000 | 2014-09-15 | Current Account       |          10008 |      4
(8 rows)


bank=> DELETE FROM Customer WHERE Cust_Id = 'AB124D8';
ERROR:  permission denied for table customer
bank=> INSERT INTO Customer(Cust_ID,First_Name,Middle_Name,Last_Name,Street,City,State,Pin,Age,Email,Gender,Nationality,Contact,PAN_no,Salary)
bank-> VALUES
bank-> ('AB124D11','Suraj','K','Kumar','Brigade Road','Bangalore','Karnataka',665512,30,'suraj@gmail.com','M','India',9722950144,'CUI906DU13',64000);
ERROR:  permission denied for table customer
bank=>
```

4. Create a user with permission to ALL privileges on all Tables for the Bank Database.



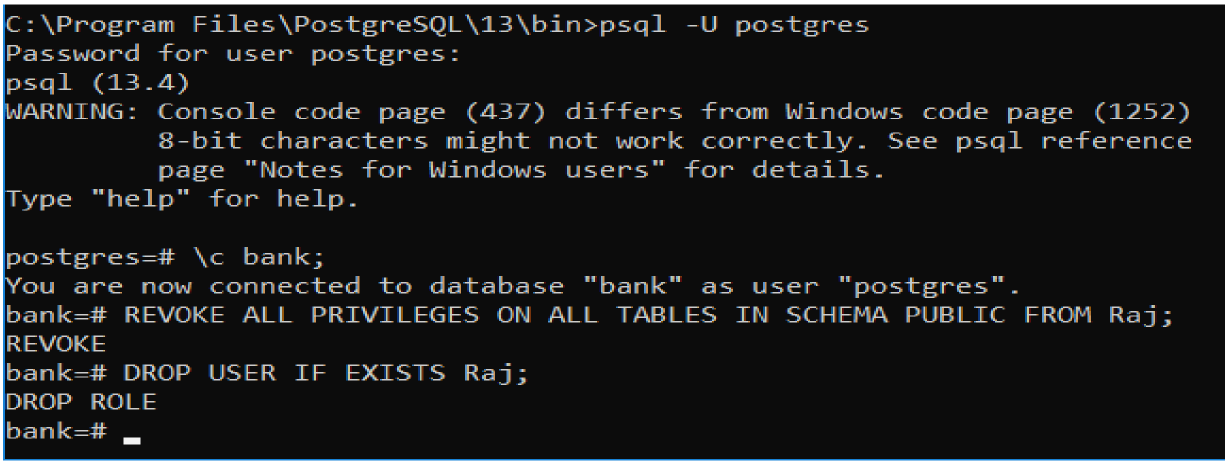
```
bank=# DROP USER IF EXISTS Raj;
NOTICE:  role "raj" does not exist, skipping
DROP ROLE
bank=# CREATE USER Raj WITH PASSWORD 'Raj338';
CREATE ROLE
bank=# GRANT ALL PRIVILEGES ON ALL TABLES IN SCHEMA PUBLIC TO Raj;
GRANT
```

Verification:



```
C:\Program Files\PostgreSQL\13\bin>psql -U raj -d bank
Password for user raj:
psql (13.4)
WARNING: Console code page (437) differs from Windows code page (1252)
         8-bit characters might not work correctly. See psql reference
         page "Notes for Windows users" for details.
Type "help" for help.

bank=> SELECT * FROM Interest;
 interest_id | savings_int | rd_int | fd_int | interest_rate | interest_amount | manager_id
-------------+-------------+--------+--------+---------------+-----------------+------------
         101 |         5.3 |    6.2 |    7.3 |          12.9 |           10000 |       1001
         102 |           5 |    4.2 |    8.3 |           9.6 |           12000 |       1003
         103 |        4.12 |   6.44 |    3.3 |           4.8 |            8000 |       1003
         104 |        6.13 |   2.17 |   5.32 |          7.98 |           12500 |       1005
         105 |        6.87 |    9.1 |   5.92 |          8.18 |           12700 |       1002
         106 |        2.34 |   7.21 |   8.11 |           6.5 |           11000 |       1002
         107 |        8.66 |   9.12 |   4.45 |          6.13 |           13000 |       1007
         108 |        8.77 |   5.96 |   4.34 |          3.18 |           12500 |       1008
(8 rows)


bank=> \q
```

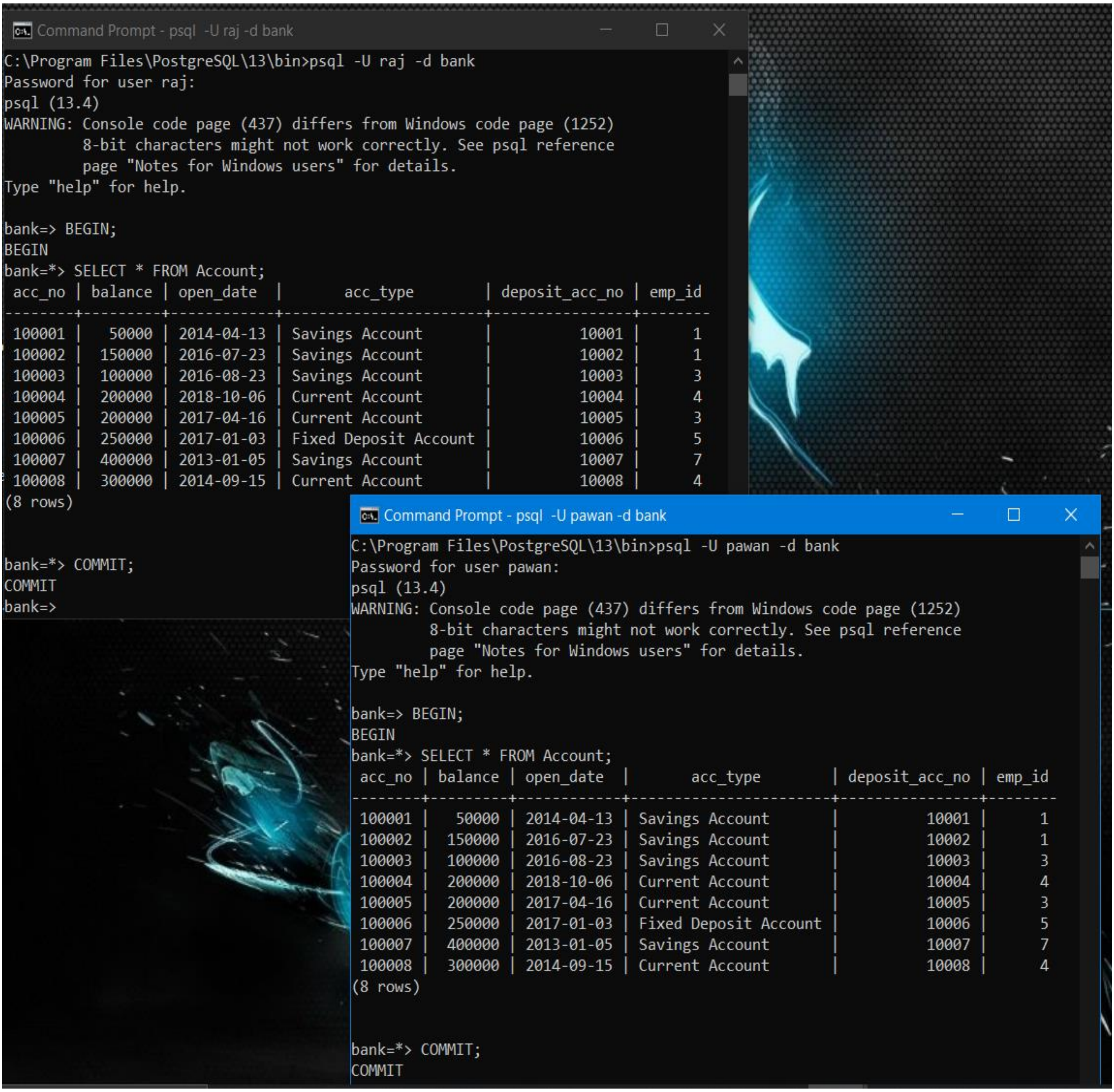
Remove the user Raj



```
C:\Program Files\PostgreSQL\13\bin>psql -U postgres
Password for user postgres:
psql (13.4)
WARNING: Console code page (437) differs from Windows code page (1252)
         8-bit characters might not work correctly. See psql reference
         page "Notes for Windows users" for details.
Type "help" for help.

postgres=# \c bank;
You are now connected to database "bank" as user "postgres".
bank=# REVOKE ALL PRIVILEGES ON ALL TABLES IN SCHEMA PUBLIC FROM Raj;
REVOKE
bank=# DROP USER IF EXISTS Raj;
DROP ROLE
bank=# _
```

Concurrent Transactions and Demonstrate the Concurrency Control for the conflicting actions

Both the users can read the data simultaneously in the transactions.

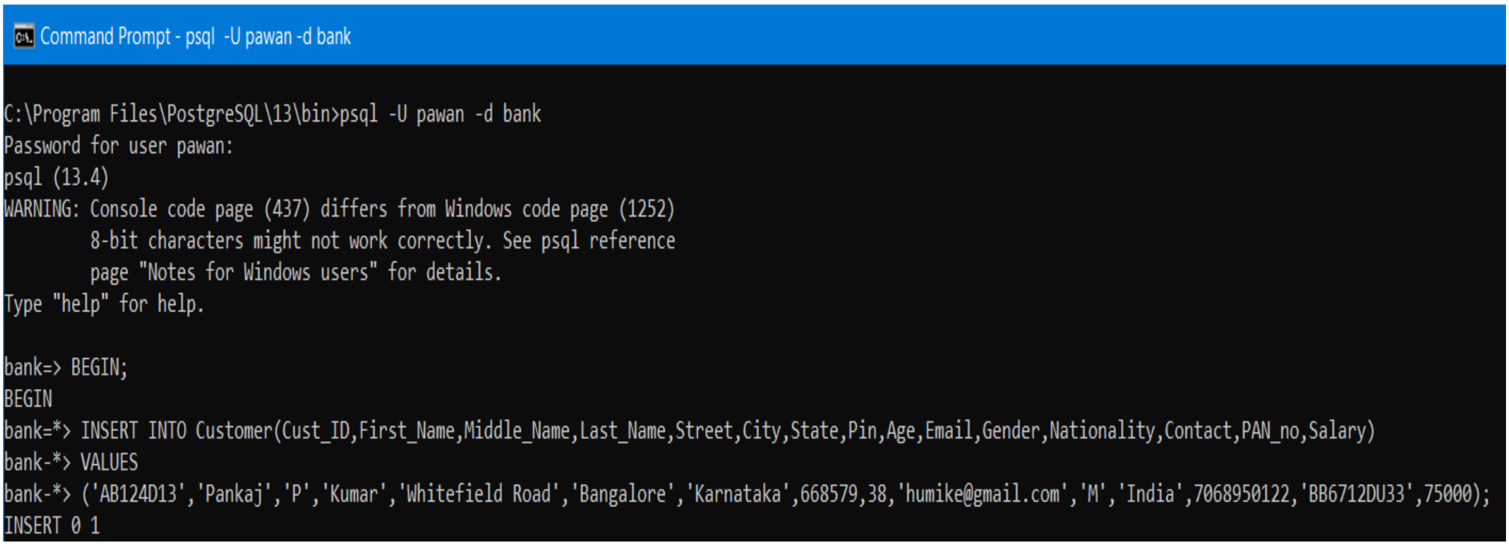


```
C:\Program Files\PostgreSQL\13\bin>psql -U raj -d bank
Password for user raj:
psql (13.4)
WARNING: Console code page (437) differs from Windows code page (1252)
         8-bit characters might not work correctly. See psql reference
         page "Notes for Windows users" for details.
Type "help" for help.

bank=> BEGIN;
BEGIN
bank=*> SELECT * FROM Account;
 acc_no | balance | open_date  |       acc_type        | deposit_acc_no | emp_id
--------+---------+------------+-----------------------+----------------+--------
 100001 |   50000 | 2014-04-13 | Savings Account       |          10001 |      1
 100002 |  150000 | 2016-07-23 | Savings Account       |          10002 |      1
 100003 |  100000 | 2016-08-23 | Savings Account       |          10003 |      3
 100004 |  200000 | 2018-10-06 | Current Account       |          10004 |      4
 100005 |  200000 | 2017-04-16 | Current Account       |          10005 |      3
 100006 |  250000 | 2017-01-03 | Fixed Deposit Account |          10006 |      5
 100007 |  400000 | 2013-01-05 | Savings Account       |          10007 |      7
 100008 |  300000 | 2014-09-15 | Current Account       |          10008 |      4
(8 rows)


bank=*> COMMIT;
COMMIT
bank=>
```

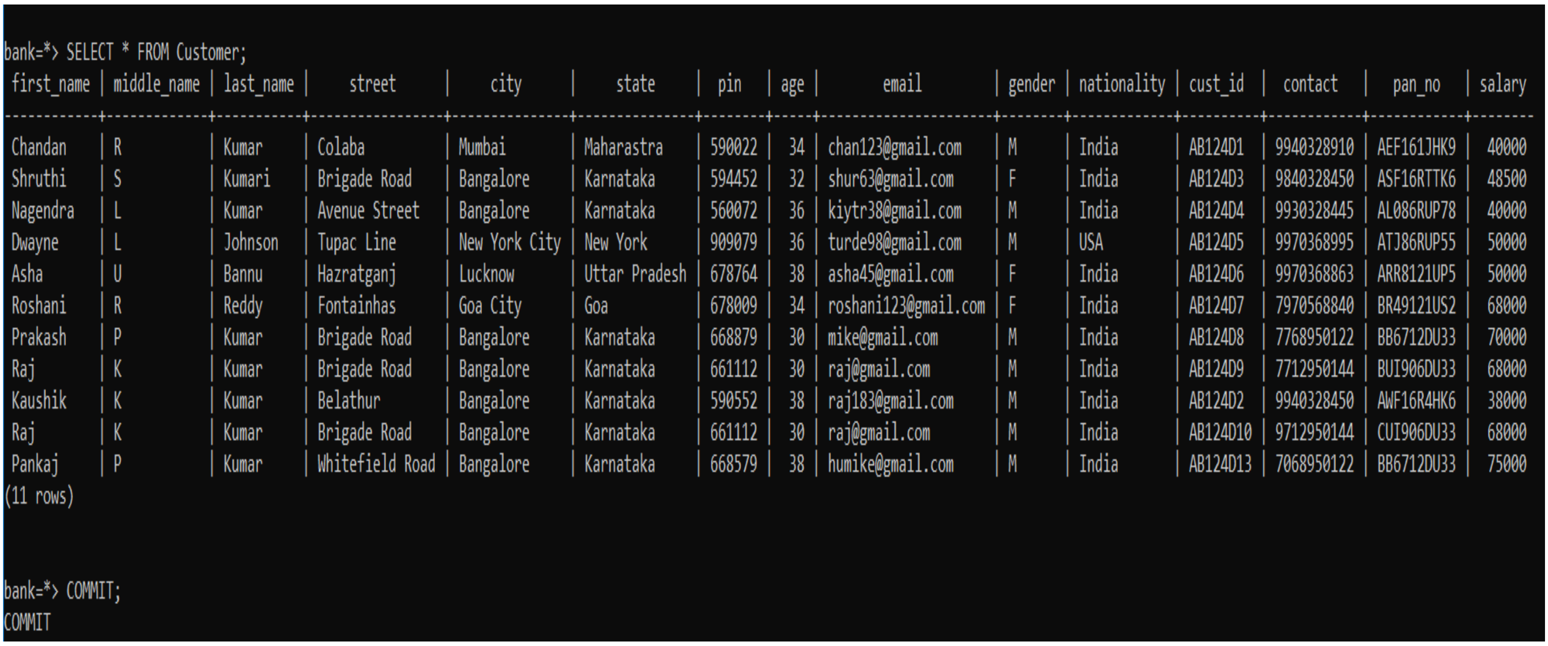
```
C:\Program Files\PostgreSQL\13\bin>psql -U pawan -d bank
Password for user pawan:
psql (13.4)
WARNING: Console code page (437) differs from Windows code page (1252)
         8-bit characters might not work correctly. See psql reference
         page "Notes for Windows users" for details.
Type "help" for help.

bank=> BEGIN;
BEGIN
bank=*> SELECT * FROM Account;
 acc_no | balance | open_date  |       acc_type        | deposit_acc_no | emp_id
--------+---------+------------+-----------------------+----------------+--------
 100001 |   50000 | 2014-04-13 | Savings Account       |          10001 |      1
 100002 |  150000 | 2016-07-23 | Savings Account       |          10002 |      1
 100003 |  100000 | 2016-08-23 | Savings Account       |          10003 |      3
 100004 |  200000 | 2018-10-06 | Current Account       |          10004 |      4
 100005 |  200000 | 2017-04-16 | Current Account       |          10005 |      3
 100006 |  250000 | 2017-01-03 | Fixed Deposit Account |          10006 |      5
 100007 |  400000 | 2013-01-05 | Savings Account       |          10007 |      7
 100008 |  300000 | 2014-09-15 | Current Account       |          10008 |      4
(8 rows)


bank=*> COMMIT;
COMMIT
```

The user Pawan started transaction and added an insert statement and the user Phani also added an insert statement simultaneously in other transaction. But after commit statement the both insert statements was shown in database bank until then it would be seen in the particular buffer memory of the transaction.

User Pawan:

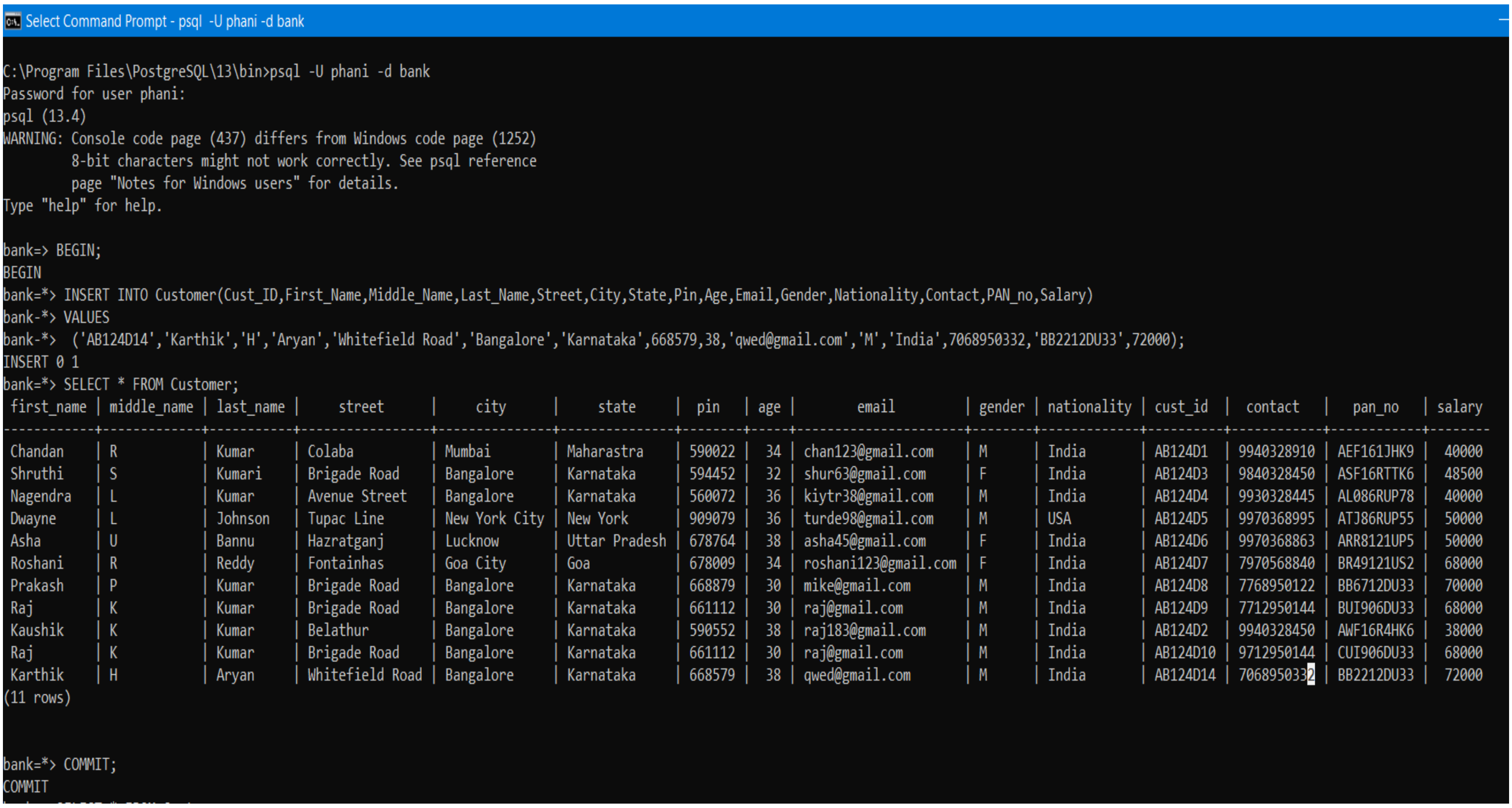


```
C:\Program Files\PostgreSQL\13\bin>psql -U pawan -d bank
Password for user pawan:
psql (13.4)
WARNING: Console code page (437) differs from Windows code page (1252)
         8-bit characters might not work correctly. See psql reference
         page "Notes for Windows users" for details.
Type "help" for help.

bank=> BEGIN;
BEGIN
bank=*> INSERT INTO Customer(Cust_ID,First_Name,Middle_Name,Last_Name,Street,City,State,Pin,Age,Email,Gender,Nationality,Contact,PAN_no,Salary)
bank-*> VALUES
bank-*> ('AB124D13','Pankaj','P','Kumar','Whitefield Road','Bangalore','Karnataka',668579,38,'humike@gmail.com','M','India',7068950122,'BB6712DU33',75000);
INSERT 0 1
```



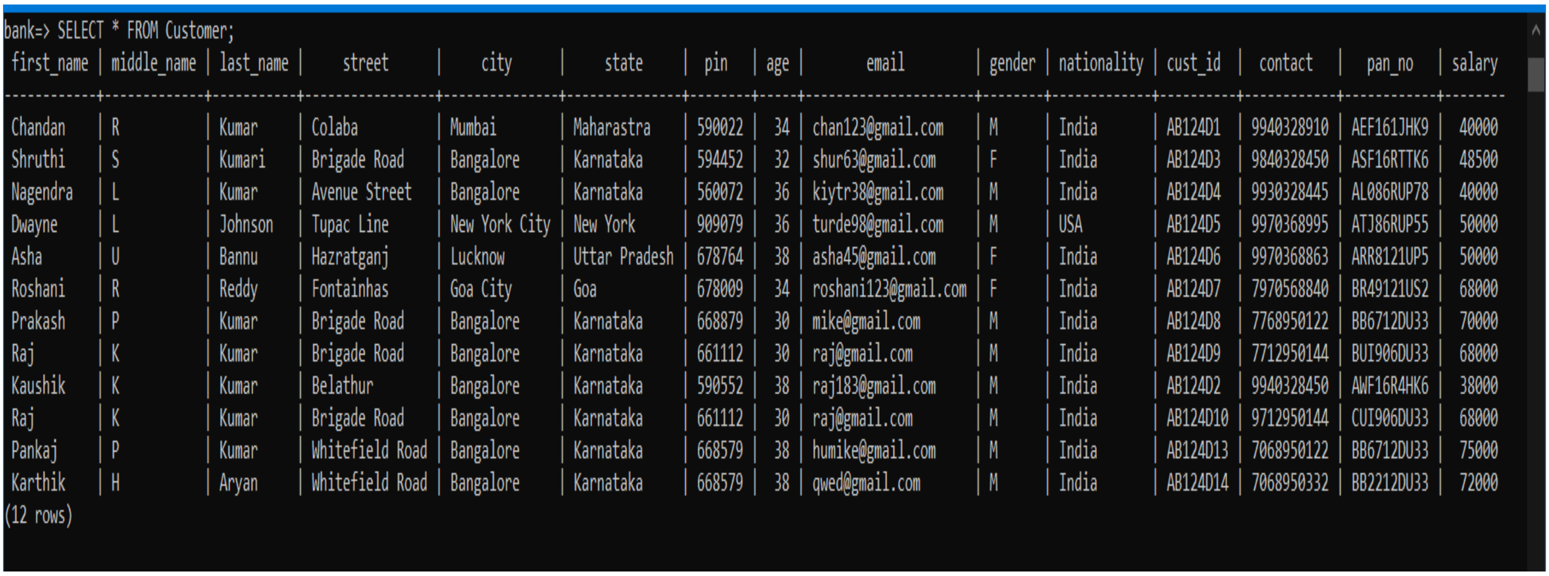
```
bank=*> SELECT * FROM Customer;
 first_name | middle_name | last_name |     street      |     city      |     state     |  pin   | age |        email         | gender | nationality | cust_id  |  contact   |   pan_no   | salary
------------+-------------+-----------+-----------------+---------------+---------------+--------+-----+----------------------+--------+-------------+----------+------------+------------+--------
 Chandan    | R           | Kumar     | Colaba          | Mumbai        | Maharastra    | 590022 |  34 | chan123@gmail.com    | M      | India       | AB124D1  | 9940328910 | AEF161JHK9 |  40000
 Shruthi    | S           | Kumari    | Brigade Road    | Bangalore     | Karnataka     | 594452 |  32 | shur63@gmail.com     | F      | India       | AB124D3  | 9840328450 | ASF16RTTK6 |  48500
 Nagendra   | L           | Kumar     | Avenue Street   | Bangalore     | Karnataka     | 560072 |  36 | kiytr38@gmail.com    | M      | India       | AB124D4  | 9930328445 | AL086RUP78 |  40000
 Dwayne     | L           | Johnson   | Tupac Line      | New York City | New York      | 909079 |  36 | turde98@gmail.com    | M      | USA         | AB124D5  | 9970368995 | ATJ86RUP55 |  50000
 Asha       | U           | Bannu     | Hazratganj      | Lucknow       | Uttar Pradesh | 678764 |  38 | asha45@gmail.com     | F      | India       | AB124D6  | 9970368863 | ARR8121UP5 |  50000
 Roshani    | R           | Reddy     | Fontainhas      | Goa City      | Goa           | 678009 |  34 | roshani123@gmail.com | F      | India       | AB124D7  | 7970568840 | BR49121US2 |  68000
 Prakash    | P           | Kumar     | Brigade Road    | Bangalore     | Karnataka     | 668879 |  30 | mike@gmail.com       | M      | India       | AB124D8  | 7768950122 | BB6712DU33 |  70000
 Raj        | K           | Kumar     | Brigade Road    | Bangalore     | Karnataka     | 661112 |  30 | raj@gmail.com        | M      | India       | AB124D9  | 7712950144 | BUI906DU33 |  68000
 Kaushik    | K           | Kumar     | Belathur        | Bangalore     | Karnataka     | 590552 |  38 | raj183@gmail.com     | M      | India       | AB124D2  | 9940328450 | AWF16R4HK6 |  38000
 Raj        | K           | Kumar     | Brigade Road    | Bangalore     | Karnataka     | 661112 |  30 | raj@gmail.com        | M      | India       | AB124D10 | 9712950144 | CUI906DU33 |  68000
 Pankaj     | P           | Kumar     | Whitefield Road | Bangalore     | Karnataka     | 668579 |  38 | humike@gmail.com     | M      | India       | AB124D13 | 7068950122 | BB6712DU33 |  75000
(11 rows)


bank=*> COMMIT;
COMMIT
```

User Phani:



```
C:\Program Files\PostgreSQL\13\bin>psql -U phani -d bank
Password for user phani:
psql (13.4)
WARNING: Console code page (437) differs from Windows code page (1252)
         8-bit characters might not work correctly. See psql reference
         page "Notes for Windows users" for details.
Type "help" for help.

bank=> BEGIN;
BEGIN
bank=*> INSERT INTO Customer(Cust_ID,First_Name,Middle_Name,Last_Name,Street,City,State,Pin,Age,Email,Gender,Nationality,Contact,PAN_no,Salary)
bank-*> VALUES
bank-*>  ('AB124D14','Karthik','H','Aryan','Whitefield Road','Bangalore','Karnataka',668579,38,'qwed@gmail.com','M','India',7068950332,'BB2212DU33',72000);
INSERT 0 1
bank=*> SELECT * FROM Customer;
 first_name | middle_name | last_name |     street      |     city      |     state     |  pin   | age |        email         | gender | nationality | cust_id  |  contact   |   pan_no   | salary
------------+-------------+-----------+-----------------+---------------+---------------+--------+-----+----------------------+--------+-------------+----------+------------+------------+--------
 Chandan    | R           | Kumar     | Colaba          | Mumbai        | Maharastra    | 590022 |  34 | chan123@gmail.com    | M      | India       | AB124D1  | 9940328910 | AEF161JHK9 |  40000
 Shruthi    | S           | Kumari    | Brigade Road    | Bangalore     | Karnataka     | 594452 |  32 | shur63@gmail.com     | F      | India       | AB124D3  | 9840328450 | ASF16RTTK6 |  48500
 Nagendra   | L           | Kumar     | Avenue Street   | Bangalore     | Karnataka     | 560072 |  36 | kiytr38@gmail.com    | M      | India       | AB124D4  | 9930328445 | AL086RUP78 |  40000
 Dwayne     | L           | Johnson   | Tupac Line      | New York City | New York      | 909079 |  36 | turde98@gmail.com    | M      | USA         | AB124D5  | 9970368995 | ATJ86RUP55 |  50000
 Asha       | U           | Bannu     | Hazratganj      | Lucknow       | Uttar Pradesh | 678764 |  38 | asha45@gmail.com     | F      | India       | AB124D6  | 9970368863 | ARR8121UP5 |  50000
 Roshani    | R           | Reddy     | Fontainhas      | Goa City      | Goa           | 678009 |  34 | roshani123@gmail.com | F      | India       | AB124D7  | 7970568840 | BR49121US2 |  68000
 Prakash    | P           | Kumar     | Brigade Road    | Bangalore     | Karnataka     | 668879 |  30 | mike@gmail.com       | M      | India       | AB124D8  | 7768950122 | BB6712DU33 |  70000
 Raj        | K           | Kumar     | Brigade Road    | Bangalore     | Karnataka     | 661112 |  30 | raj@gmail.com        | M      | India       | AB124D9  | 7712950144 | BUI906DU33 |  68000
 Kaushik    | K           | Kumar     | Belathur        | Bangalore     | Karnataka     | 590552 |  38 | raj183@gmail.com     | M      | India       | AB124D2  | 9940328450 | AWF16R4HK6 |  38000
 Raj        | K           | Kumar     | Brigade Road    | Bangalore     | Karnataka     | 661112 |  30 | raj@gmail.com        | M      | India       | AB124D10 | 9712950144 | CUI906DU33 |  68000
 Karthik    | H           | Aryan     | Whitefield Road | Bangalore     | Karnataka     | 668579 |  38 | qwed@gmail.com       | M      | India       | AB124D14 | 7068950332 | BB2212DU33 |  72000
(11 rows)


bank=*> COMMIT;
COMMIT
```

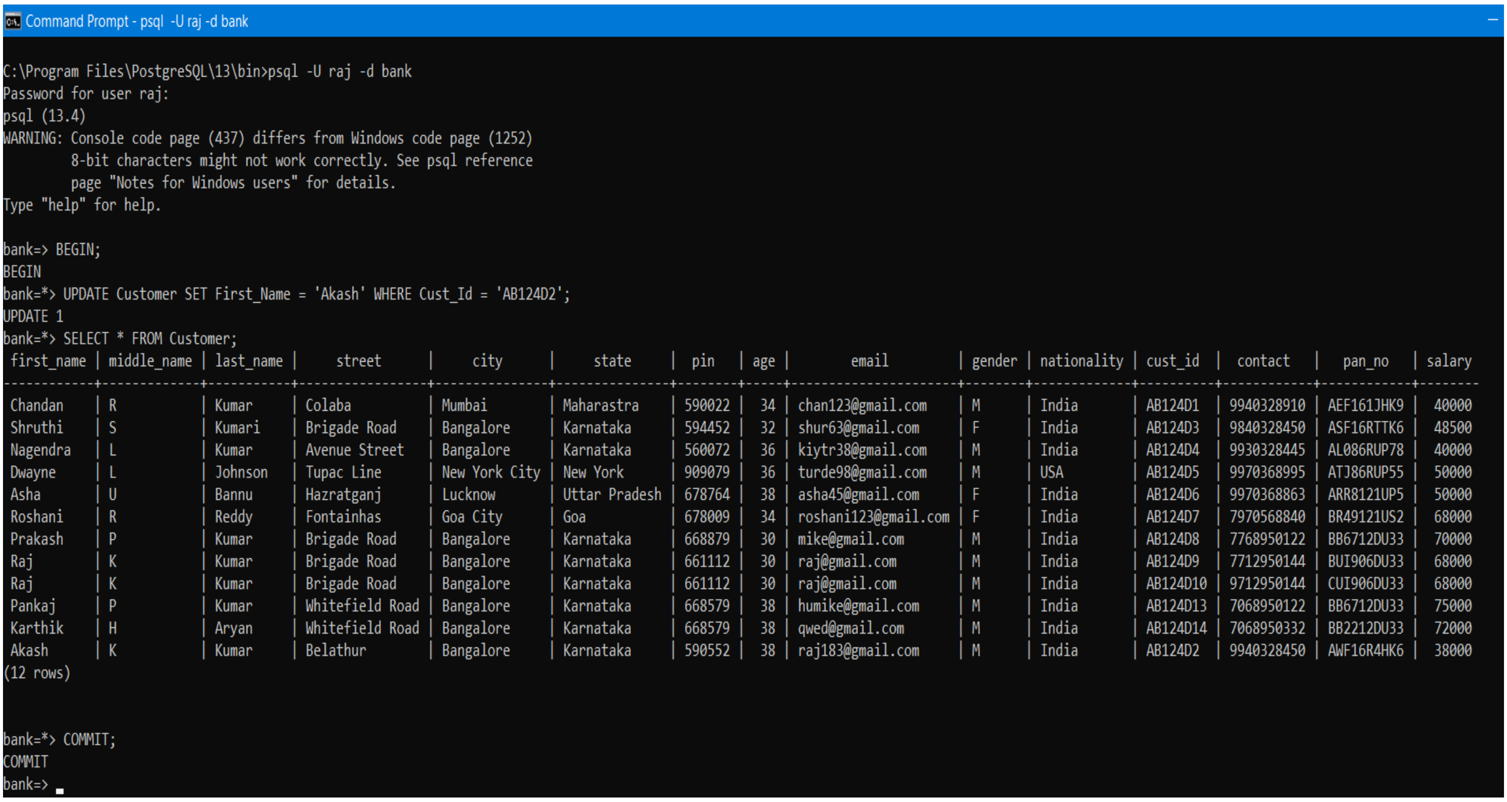
Both the insert values are updated in the main database after commit statement.



```
bank=> SELECT * FROM Customer;
 first_name | middle_name | last_name |     street      |     city      |     state     |  pin   | age |        email         | gender | nationality | cust_id  |  contact   |   pan_no   | salary
------------+-------------+-----------+-----------------+---------------+---------------+--------+-----+----------------------+--------+-------------+----------+------------+------------+--------
 Chandan    | R           | Kumar     | Colaba          | Mumbai        | Maharastra    | 590022 |  34 | chan123@gmail.com    | M      | India       | AB124D1  | 9940328910 | AEF161JHK9 |  40000
 Shruthi    | S           | Kumari    | Brigade Road    | Bangalore     | Karnataka     | 594452 |  32 | shur63@gmail.com     | F      | India       | AB124D3  | 9840328450 | ASF16RTTK6 |  48500
 Nagendra   | L           | Kumar     | Avenue Street   | Bangalore     | Karnataka     | 560072 |  36 | kiytr38@gmail.com    | M      | India       | AB124D4  | 9930328445 | AL086RUP78 |  40000
 Dwayne     | L           | Johnson   | Tupac Line      | New York City | New York      | 909079 |  36 | turde98@gmail.com    | M      | USA         | AB124D5  | 9970368995 | ATJ86RUP55 |  50000
 Asha       | U           | Bannu     | Hazratganj      | Lucknow       | Uttar Pradesh | 678764 |  38 | asha45@gmail.com     | F      | India       | AB124D6  | 9970368863 | ARR8121UP5 |  50000
 Roshani    | R           | Reddy     | Fontainhas      | Goa City      | Goa           | 678009 |  34 | roshani123@gmail.com | F      | India       | AB124D7  | 7970568840 | BR49121US2 |  68000
 Prakash    | P           | Kumar     | Brigade Road    | Bangalore     | Karnataka     | 668879 |  30 | mike@gmail.com       | M      | India       | AB124D8  | 7768950122 | BB6712DU33 |  70000
 Raj        | K           | Kumar     | Brigade Road    | Bangalore     | Karnataka     | 661112 |  30 | raj@gmail.com        | M      | India       | AB124D9  | 7712950144 | BUI906DU33 |  68000
 Kaushik    | K           | Kumar     | Belathur        | Bangalore     | Karnataka     | 590552 |  38 | raj183@gmail.com     | M      | India       | AB124D2  | 9940328450 | AWF16R4HK6 |  38000
 Raj        | K           | Kumar     | Brigade Road    | Bangalore     | Karnataka     | 661112 |  30 | raj@gmail.com        | M      | India       | AB124D10 | 9712950144 | CUI906DU33 |  68000
 Pankaj     | P           | Kumar     | Whitefield Road | Bangalore     | Karnataka     | 668579 |  38 | humike@gmail.com     | M      | India       | AB124D13 | 7068950122 | BB6712DU33 |  75000
 Karthik    | H           | Aryan     | Whitefield Road | Bangalore     | Karnataka     | 668579 |  38 | qwed@gmail.com       | M      | India       | AB124D14 | 7068950332 | BB2212DU33 |  72000
(12 rows)
```

Updating the values of the same data in transactions of both user Raj and Phani simultaneously and we can see that second transaction of user Phani wait until the Raj commit the transaction and then the user Phani’s Update statement is executed and then he can commit. This shows conflicting actions on concurrent control.

User Raj:

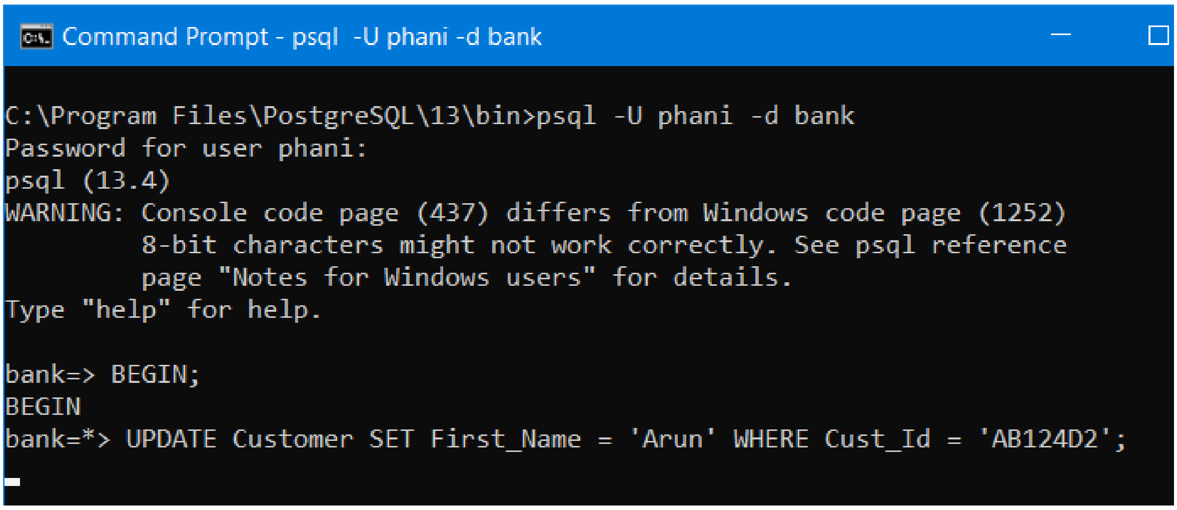


Command Prompt - psql -U raj -d bank

```
C:\Program Files\PostgreSQL\13\bin>psql -U raj -d bank
Password for user raj:
psql (13.4)
WARNING: Console code page (437) differs from Windows code page (1252)
         8-bit characters might not work correctly. See psql reference
         page "Notes for Windows users" for details.
Type "help" for help.

bank=> BEGIN;
BEGIN
bank=*> UPDATE Customer SET First_Name = 'Akash' WHERE Cust_Id = 'AB124D2';
UPDATE 1
bank=*> SELECT * FROM Customer;
 first_name | middle_name | last_name |     street      |     city      |     state     |  pin   | age |        email         | gender | nationality | cust_id  |  contact   |   pan_no   | salary
------------+-------------+-----------+-----------------+---------------+---------------+--------+-----+----------------------+--------+-------------+----------+------------+------------+--------
 Chandan    | R           | Kumar     | Colaba          | Mumbai        | Maharastra    | 590022 |  34 | chan123@gmail.com    | M      | India       | AB124D1  | 9940328910 | AEF161JHK9 |  40000
 Shruthi    | S           | Kumari    | Brigade Road    | Bangalore     | Karnataka     | 594452 |  32 | shur63@gmail.com     | F      | India       | AB124D3  | 9840328450 | ASF16RTTK6 |  48500
 Nagendra   | L           | Kumar     | Avenue Street   | Bangalore     | Karnataka     | 560072 |  36 | kiytr38@gmail.com    | M      | India       | AB124D4  | 9930328445 | AL086RUP78 |  40000
 Dwayne     | L           | Johnson   | Tupac Line      | New York City | New York      | 909079 |  36 | turde98@gmail.com    | M      | USA         | AB124D5  | 9970368995 | ATJ86RUP55 |  50000
 Asha       | U           | Bannu     | Hazratganj      | Lucknow       | Uttar Pradesh | 678764 |  38 | asha45@gmail.com     | F      | India       | AB124D6  | 9970368863 | ARR8121UP5 |  50000
 Roshani    | R           | Reddy     | Fontainhas      | Goa City      | Goa           | 678009 |  34 | roshani123@gmail.com | F      | India       | AB124D7  | 7970568840 | BR49121US2 |  68000
 Prakash    | P           | Kumar     | Brigade Road    | Bangalore     | Karnataka     | 668879 |  30 | mike@gmail.com       | M      | India       | AB124D8  | 7768950122 | BB6712DU33 |  70000
 Raj        | K           | Kumar     | Brigade Road    | Bangalore     | Karnataka     | 661112 |  30 | raj@gmail.com        | M      | India       | AB124D9  | 7712950144 | BUI906DU33 |  68000
 Raj        | K           | Kumar     | Brigade Road    | Bangalore     | Karnataka     | 661112 |  30 | raj@gmail.com        | M      | India       | AB124D10 | 9712950144 | CUI906DU33 |  68000
 Pankaj     | P           | Kumar     | Whitefield Road | Bangalore     | Karnataka     | 668579 |  38 | humike@gmail.com     | M      | India       | AB124D13 | 7068950122 | BB6712DU33 |  75000
 Karthik    | H           | Aryan     | Whitefield Road | Bangalore     | Karnataka     | 668579 |  38 | qwed@gmail.com       | M      | India       | AB124D14 | 7068950332 | BB2212DU33 |  72000
 Akash      | K           | Kumar     | Belathur        | Bangalore     | Karnataka     | 590552 |  38 | raj183@gmail.com     | M      | India       | AB124D2  | 9940328450 | AWF16R4HK6 |  38000
(12 rows)


bank=*> COMMIT;
COMMIT
bank=>
```

User Phani:

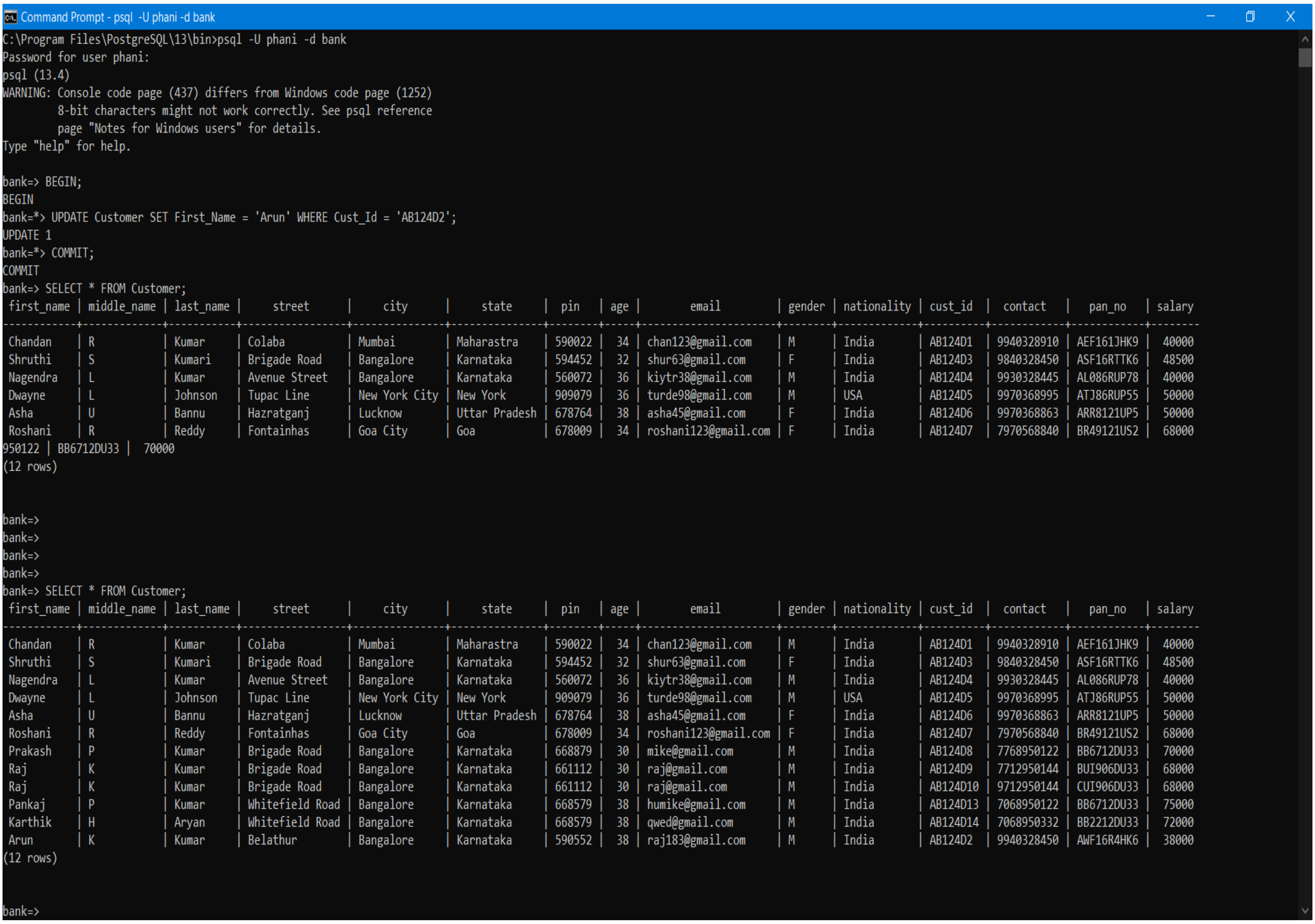
User Phani is waiting until the transaction by user Raj is committed.



Command Prompt - psql -U phani -d bank

```
C:\Program Files\PostgreSQL\13\bin>psql -U phani -d bank
Password for user phani:
psql (13.4)
WARNING: Console code page (437) differs from Windows code page (1252)
         8-bit characters might not work correctly. See psql reference
         page "Notes for Windows users" for details.
Type "help" for help.

bank=> BEGIN;
BEGIN
bank=*> UPDATE Customer SET First_Name = 'Arun' WHERE Cust_Id = 'AB124D2';
```

Since both User’s were working on same data of the database the update statement of last transaction is reflected on main database.



```
C:\Program Files\PostgreSQL\13\bin>psql -U phani -d bank
Password for user phani:
psql (13.4)
WARNING: Console code page (437) differs from Windows code page (1252)
         8-bit characters might not work correctly. See psql reference
         page "Notes for Windows users" for details.
Type "help" for help.

bank=> BEGIN;
BEGIN
bank=*> UPDATE Customer SET First_Name = 'Arun' WHERE Cust_Id = 'AB124D2';
UPDATE 1
bank=*> COMMIT;
COMMIT
bank=> SELECT * FROM Customer;
 first_name | middle_name | last_name |     street      |     city      |     state     |  pin   | age |        email         | gender | nationality | cust_id  |  contact   |   pan_no   | salary
------------+-------------+-----------+-----------------+---------------+---------------+--------+-----+----------------------+--------+-------------+----------+------------+------------+--------
 Chandan    | R           | Kumar     | Colaba          | Mumbai        | Maharastra    | 590022 |  34 | chan123@gmail.com    | M      | India       | AB124D1  | 9940328910 | AEF161JHK9 |  40000
 Shruthi    | S           | Kumari    | Brigade Road    | Bangalore     | Karnataka     | 594452 |  32 | shur63@gmail.com     | F      | India       | AB124D3  | 9840328450 | ASF16RTTK6 |  48500
 Nagendra   | L           | Kumar     | Avenue Street   | Bangalore     | Karnataka     | 560072 |  36 | kiytr38@gmail.com    | M      | India       | AB124D4  | 9930328445 | AL086RUP78 |  40000
 Dwayne     | L           | Johnson   | Tupac Line      | New York City | New York      | 909079 |  36 | turde98@gmail.com    | M      | USA         | AB124D5  | 9970368995 | ATJ86RUP55 |  50000
 Asha       | U           | Bannu     | Hazratganj      | Lucknow       | Uttar Pradesh | 678764 |  38 | asha45@gmail.com     | F      | India       | AB124D6  | 9970368863 | ARR8121UP5 |  50000
 Roshani    | R           | Reddy     | Fontainhas      | Goa City      | Goa           | 678009 |  34 | roshani123@gmail.com | F      | India       | AB124D7  | 7970568840 | BR49121US2 |  68000
950122 | BB6712DU33 |  70000
(12 rows)


bank=>
bank=>
bank=>
bank=>
bank=> SELECT * FROM Customer;
 first_name | middle_name | last_name |     street      |     city      |     state     |  pin   | age |        email         | gender | nationality | cust_id  |  contact   |   pan_no   | salary
------------+-------------+-----------+-----------------+---------------+---------------+--------+-----+----------------------+--------+-------------+----------+------------+------------+--------
 Chandan    | R           | Kumar     | Colaba          | Mumbai        | Maharastra    | 590022 |  34 | chan123@gmail.com    | M      | India       | AB124D1  | 9940328910 | AEF161JHK9 |  40000
 Shruthi    | S           | Kumari    | Brigade Road    | Bangalore     | Karnataka     | 594452 |  32 | shur63@gmail.com     | F      | India       | AB124D3  | 9840328450 | ASF16RTTK6 |  48500
 Nagendra   | L           | Kumar     | Avenue Street   | Bangalore     | Karnataka     | 560072 |  36 | kiytr38@gmail.com    | M      | India       | AB124D4  | 9930328445 | AL086RUP78 |  40000
 Dwayne     | L           | Johnson   | Tupac Line      | New York City | New York      | 909079 |  36 | turde98@gmail.com    | M      | USA         | AB124D5  | 9970368995 | ATJ86RUP55 |  50000
 Asha       | U           | Bannu     | Hazratganj      | Lucknow       | Uttar Pradesh | 678764 |  38 | asha45@gmail.com     | F      | India       | AB124D6  | 9970368863 | ARR8121UP5 |  50000
 Roshani    | R           | Reddy     | Fontainhas      | Goa City      | Goa           | 678009 |  34 | roshani123@gmail.com | F      | India       | AB124D7  | 7970568840 | BR49121US2 |  68000
 Prakash    | P           | Kumar     | Brigade Road    | Bangalore     | Karnataka     | 668879 |  30 | mike@gmail.com       | M      | India       | AB124D8  | 7768950122 | BB6712DU33 |  70000
 Raj        | K           | Kumar     | Brigade Road    | Bangalore     | Karnataka     | 661112 |  30 | raj@gmail.com        | M      | India       | AB124D9  | 7712950144 | BUI906DU33 |  68000
 Raj        | K           | Kumar     | Brigade Road    | Bangalore     | Karnataka     | 661112 |  30 | raj@gmail.com        | M      | India       | AB124D10 | 9712950144 | CUI906DU33 |  68000
 Pankaj     | P           | Kumar     | Whitefield Road | Bangalore     | Karnataka     | 668579 |  38 | humike@gmail.com     | M      | India       | AB124D13 | 7068950122 | BB6712DU33 |  75000
 Karthik    | H           | Aryan     | Whitefield Road | Bangalore     | Karnataka     | 668579 |  38 | qwed@gmail.com       | M      | India       | AB124D14 | 7068950332 | BB2212DU33 |  72000
 Arun       | K           | Kumar     | Belathur        | Bangalore     | Karnataka     | 590552 |  38 | raj183@gmail.com     | M      | India       | AB124D2  | 9940328450 | AWF16R4HK6 |  38000
(12 rows)


bank=>
```

CONTRIBUTIONS:

NAME	SRN	CONTRIBUTION	TIME SPENT
Pawan Prasad P	PES2UG19CS280	Updated Statements and ER Diagram, Access Privileges and Concurrent Transactions	100 min
Phani Kumar Vedurumudi	PES2UG19CS281	Complex Queries and Nested Queries	90 min
Rahul S Bhat	PES2UG19CS315	Simple Queries and Performance Analysis	80 min