1. In total, how many transactions have been carried out at the bank?

SELECT COUNT(*) AS total_transactions FROM acc_transaction;

```
bank_management=# SELECT COUNT(*) AS total_transactions FROM acc_transaction;
 total_transactions
--------------------
                 19
(1 行记录)
```

2. How many accounts of type ‘CHK’ are there at this bank?

SELECT COUNT(*) AS total_chk_accounts FROM account WHERE product_cd = 'CHK';

```
bank_management=# SELECT COUNT(*) AS total_chk_accounts FROM account WHERE product_cd = 'CHK';
 total_chk_accounts
--------------------
                 10
(1 行记录)
```

3. Produce a list of job titles and how many employees hold this position.

SELECT title, COUNT(*) AS num_employees FROM employee GROUP BY title;

```
bank_management=# SELECT title, COUNT(*) AS num_employees FROM employee GROUP BY title;
       title        | num_employees
--------------------+---------------
 Vice President     |             1
 Head Teller        |             4
 Teller             |             9
 Treasurer          |             1
 President          |             1
 Operations Manager |             1
 Loan Manager       |             1
(7 行记录)
```

4. Produce a list of Customers and the number of accounts they have.

```
SELECT c.cust_id,
        COUNT(a.account_id) AS num_accounts
FROM customer c
LEFT JOIN account a ON c.cust_id = a.cust_id
GROUP BY c.cust_id
ORDER BY c.cust_id;
```

```
bank_management=# SELECT c.cust_id,
bank_management-#        COUNT(a.account_id) AS num_accounts
bank_management-# FROM customer c
bank_management-# LEFT JOIN account a ON c.cust_id = a.cust_id
bank_management-# GROUP BY c.cust_id
bank_management-# ORDER BY c.cust_id;
 cust_id | num_accounts
---------+--------------
       1 |            3
       2 |            2
       3 |            2
       4 |            3
       5 |            1
       6 |            2
       7 |            1
       8 |            2
       9 |            3
      10 |            2
      11 |            0
      12 |            1
      13 |            1
(13 行记录)
```

5. What is the total available balance for the customer; James Hadley (cust_id = 1)?

```
SELECT SUM(avail_balance) AS total_balance
FROM account
WHERE cust_id = 1;
```

```
bank_management=# SELECT SUM(avail_balance) AS total_balance
bank_management-# FROM account
bank_management-# WHERE cust_id = 1;
 total_balance
---------------
       4648.91
(1 行记录)
```

6. Produce a list of all customers and their total available balance.

```
SELECT c.cust_id,
        SUM(a.avail_balance) AS total_balance
FROM customer c
LEFT JOIN account a ON c.cust_id = a.cust_id
GROUP BY c.cust_id
ORDER BY c.cust_id;
```

```
bank_management=# SELECT c.cust_id,
bank_management-#        SUM(a.avail_balance) AS total_balance
bank_management-# FROM customer c
bank_management-# LEFT JOIN account a ON c.cust_id = a.cust_id
bank_management-# GROUP BY c.cust_id
bank_management-# ORDER BY c.cust_id;
 cust_id | total_balance
---------+---------------
       1 |       4648.91
       2 |       2458.02
       3 |       3270.25
       4 |       6788.98
       5 |       2237.97
       6 |      10122.37
       7 |       5000.00
       8 |       3875.18
       9 |      10971.22
      10 |      23575.12
      11 |
      12 |      38552.05
      13 |      50000.00
(13 行记录)
```

7. Write a query to list all account product types and the average available balance for each type.

SELECT product_cd, AVG(avail_balance) AS avg_balance
FROM account
GROUP BY product_cd;

```
bank_management=# SELECT product_cd, AVG(avail_balance) AS avg_balance
bank_management-# FROM account
bank_management-# GROUP BY product_cd;
 product_cd |      avg_balance
------------+------------------------
 CHK        |  7302.9170000000000000
 SBL        |    50000.000000000000
 MM         |  5681.7133333333333333
 SAV        |   466.4400000000000000
 CD         |  4890.0000000000000000
 BUS        | 0.00000000000000000000
(6 行记录)
```

8. Find the total available balance in customers' accounts where the opening branch was the Woburn Branch.

SELECT SUM(avail_balance) AS total_balance
FROM account
WHERE open_branch_id = (SELECT branch_id FROM branch WHERE name = 'Woburn Branch');

```
bank_management=# SELECT SUM(avail_balance) AS total_balance
bank_management-# FROM account
bank_management-# WHERE open_branch_id = (SELECT branch_id FROM branch WHERE name = 'Woburn Branch');
 total_balance
---------------
      12106.93
(1 行记录)
```

9. Produce a list of account product types and the highest available balance for each.

SELECT product_cd, MAX(avail_balance) AS max_balance
FROM account
GROUP BY product_cd;

```
bank_management=#
bank_management=#
bank_management=#
bank_management=# SELECT product_cd, MAX(avail_balance) AS max_balance
bank_management-# FROM account
bank_management-# GROUP BY product_cd;
 product_cd | max_balance
------------+-------------
 CHK        |    38552.05
 SBL        |    50000.00
 MM         |     9345.55
 SAV        |      767.77
 CD         |    10000.00
 BUS        |        0.00
(6 行记录)
```

10. What is the minimum available balance?
SELECT MIN(avail_balance) AS min_balance FROM account;

```
bank_management=# SELECT MIN(avail_balance) AS min_balance FROM account;
 min_balance
-------------
        0.00
(1 行记录)
```

11. Produce a list of the total available balance per customer. The balance displayed should be rounded down.

```
SELECT
    c.cust_id,
    FLOOR(SUM(a.avail_balance)) AS total_available_balance
FROM
    customer c
LEFT JOIN
    account a ON c.cust_id = a.cust_id
GROUP BY
    c.cust_id
ORDER BY
    c.cust_id;
```

```
bank_management=# SELECT
bank_management-#     c.cust_id,
bank_management-#     FLOOR(SUM(a.avail_balance)) AS total_available_balance
bank_management-# FROM
bank_management-#     customer c
bank_management-# LEFT JOIN
bank_management-#     account a ON c.cust_id = a.cust_id
bank_management-# GROUP BY
bank_management-#     c.cust_id
bank_management-# ORDER BY
bank_management-#     c.cust_id;
 cust_id | total_available_balance
---------+-------------------------
       1 |                    4648
       2 |                    2458
       3 |                    3270
       4 |                    6788
       5 |                    2237
       6 |                   10122
       7 |                    5000
       8 |                    3875
       9 |                   10971
      10 |                   23575
      11 |
      12 |                   38552
      13 |                   50000
(13 行记录)
```

12. The Output lists of EMPLOYEE details in the following formats.

a. Employees Name; [LAST_NAME], [FIRST_NAME] e.g. Smith, Michael

SELECT CONCAT(last_name, ', ', first_name) AS employee_name FROM employee;

```
bank_management=# SELECT CONCAT(last_name, ', ', first_name) AS employee_name FROM employee;
   employee_name
-------------------
 Smith, Michael
 Barker, Susan
 Tyler, Robert
 Hawthorne, Susan
 Gooding, John
 Fleming, Helen
 Tucker, Chris
 Parker, Sarah
 Grossman, Jane
 Roberts, Paula
 Jameson, Samantha
 Blake, John
 Mason, Cindy
 Portman, Frank
 Markham, Theresa
 Fowler, Beth
 Tulman, Rick
 Ziegler, Thomas
(18 行记录)
```

b. Employee Position; [FIRST_NAME] [LAST_NAME] Position: [TITLE] e.g. Michael Smith Position: President

SELECT CONCAT(first_name, ' ', last_name, ' Position: ', title) AS employee_position FROM employee;

```
bank_management=# SELECT CONCAT(first_name, ' ', last_name, ' Position: ', title) AS employee_position FROM employee;
          employee_position
-------------------------------------------
 Michael Smith Position: President
 Susan Barker Position: Vice President
 Robert Tyler Position: Treasurer
 Susan Hawthorne Position: Operations Manager
 John Gooding Position: Loan Manager
 Helen Fleming Position: Head Teller
 Chris Tucker Position: Teller
 Sarah Parker Position: Teller
 Jane Grossman Position: Teller
 Paula Roberts Position: Head Teller
 Samantha Jameson Position: Teller
 John Blake Position: Head Teller
 Cindy Mason Position: Teller
 Frank Portman Position: Teller
 Theresa Markham Position: Head Teller
 Beth Fowler Position: Teller
 Rick Tulman Position: Teller
 Thomas Ziegler Position: Teller
(18 行记录)
```

13. Swap the word ‘anger’ for ‘panic buying’.

SELECT REPLACE('Fear leads to anger; anger leads to hatred; hatred leads to conflict; conflict leads to suffering', 'anger', 'panic buying') AS modified_text;

```
bank_management=# SELECT REPLACE('Fear leads to anger; anger leads to hatred; hatred leads to conflict; conflict leads
to suffering', 'anger', 'panic buying') AS modified_text;
                                                modified_text
-----------------------------------------------------------------------------------------------------------------------
 Fear leads to panic buying; panic buying leads to hatred; hatred leads to conflict; conflict leads to suffering
(1 行记录)
```

14. Standardise the format of FED_ID to nnnnnnnn.

UPDATE customer SET fed_id = REPLACE(REPLACE(fed_id, '-', ''), 'nn', '') WHERE fed_id LIKE '%-%';

```
bank_management=# UPDATE customer SET fed_id = REPLACE(REPLACE(fed_id, '-', ''), 'nn', '') WHERE fed_id LIKE '%-%';
UPDATE 9
bank_management=#
```

15. Return the year portion of the account transaction date and the number of transactions that took place in each year.

SELECT EXTRACT(YEAR FROM txn_date) AS year, COUNT(*) AS count

FROM acc_transaction

GROUP BY year;

```
bank_management=# SELECT EXTRACT(YEAR FROM txn_date) AS year, COUNT(*) AS count
bank_management-# FROM acc_transaction
bank_management-# GROUP BY year;
 year | count
------+-------
 2001 |     4
 2004 |     7
 2000 |     3
 2002 |     4
 2003 |     1
(5 行记录)
```

16. Update the EMPLOYEE table to store everyone’s job title in Uppercase.

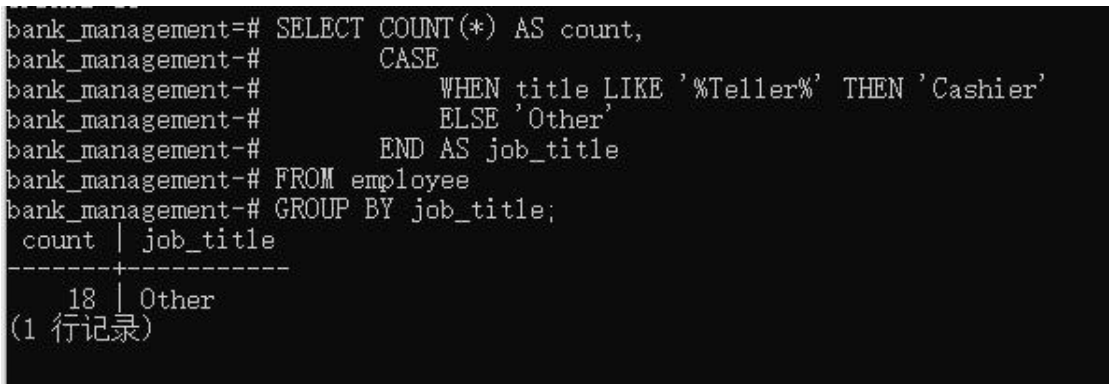
UPDATE employee SET title = UPPER(title);

```
bank_management=# UPDATE employee SET title = UPPER(title);
UPDATE 18
bank_management=#
```

Report showing the number of employees that have Teller as part of their job title and a count of all other employees.

SELECT COUNT(*) AS count,

```
        CASE
            WHEN title LIKE '%Teller%' THEN 'Cashier'
            ELSE 'Other'
        END AS job_title
FROM employee
GROUP BY job_title;
```



17. Produce a list of customers whose accumulated available balance is less than £5000.

```
SELECT
    c.cust_id,
    SUM(a.avail_balance) AS total_balance
FROM
    customer c
LEFT JOIN
    account a ON c.cust_id = a.cust_id
GROUP BY
    c.cust_id
HAVING
    SUM(a.avail_balance) < 5000;
```

```
bank_management=# SELECT
bank_management-#     c.cust_id,
bank_management-#     SUM(a.avail_balance) AS total_balance
bank_management-# FROM
bank_management-#     customer c
bank_management-# LEFT JOIN
bank_management-#     account a ON c.cust_id = a.cust_id
bank_management-# GROUP BY
bank_management-#     c.cust_id
bank_management-# HAVING
bank_management-#     SUM(a.avail_balance) < 5000;
 cust_id | total_balance
---------+---------------
       3 |       3270.25
       1 |       4648.91
       8 |       3875.18
       2 |       2458.02
       5 |       2237.97
(5 行记录)
```

18. Produce a report showing the total number of staff assigned to each branch.

```
SELECT b.name AS branch_name, COUNT(e.emp_id) AS staff_count
FROM branch b
LEFT JOIN employee e ON b.branch_id = e.assigned_branch_id
GROUP BY b.branch_id;
```

```
bank_management=# SELECT b.name AS branch_name, COUNT(e.emp_id) AS staff_count
bank_management-# FROM branch b
bank_management-# LEFT JOIN employee e ON b.branch_id = e.assigned_branch_id
bank_management-# GROUP BY b.branch_id;
  branch_name   | staff_count
----------------+-------------
 Quincy Branch  |           3
 Headquarters   |           9
 So. NH Branch  |           3
 Woburn Branch  |           3
(4 行记录)
```

19. Produce a report showing the total number of accounts which have the product codes CHK and SAV.

```
SELECT
    COUNT(*) AS total_accounts
FROM
    account
WHERE
    product_cd IN ('CHK', 'SAV');
```

```
bank_management=# SELECT
bank_management-#     COUNT(*) AS total_accounts
bank_management-# FROM
bank_management-#     account
bank_management-# WHERE
bank_management-#     product_cd IN ('CHK', 'SAV');
 total_accounts
----------------
             14
(1 行记录)
```