**Qiu Yuyue, Li Xudong, Cao Kuangzheng**

1. **Basic Tasks**

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

**SELECT COUNT(\*) AS Total\_Transactions FROM ACC\_TRANSACTION;**

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

**SELECT COUNT(\*) AS CHK\_Accounts FROM ACCOUNT WHERE productCode = 'CHK';**

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

**SELECT title, COUNT(\*) AS Number\_Of\_Employees FROM EMPLOYEE GROUP BY title;**

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

**SELECT custID, COUNT(\*) AS Number\_Of\_Accounts FROM ACCOUNT GROUP BY custID;**

**5. What is the total available balance for the customer; James Hadley (cust\_id = 1)?**

**SELECT SUM(availableBalance) AS Total\_Balance FROM ACCOUNT WHERE custID = 1;**

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

**SELECT custID, SUM(availableBalance) AS Total\_Balance FROM ACCOUNT GROUP BY custID;**

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

**SELECT productCode, AVG(availableBalance) AS Average\_Balance FROM ACCOUNT GROUP BY productCode;**

1. **Medium Tasks**

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

**SELECT SUM(availableBalance) AS Total\_Balance FROM ACCOUNT WHERE branchID = (SELECT branchID FROM BRANCH WHERE branchName = 'Woburn');**

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

**SELECT productCode, MAX(availableBalance) AS Highest\_Balance FROM ACCOUNT GROUP BY productCode;**

**10. What is the minimum available balance?**

**SELECT MIN(availableBalance) AS Minimum\_Balance FROM ACCOUNT;**

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

**SELECT custID, FLOOR(SUM(availableBalance)) AS Total\_Balance FROM ACCOUNT GROUP BY custID;**

**12. The Output lists of EMPLOYEE details in the following formats (only one column should be output):**

**a. Employees Name; [LAST\_NAME], [FIRST\_NAME] e.g. Smith, Michael**

**SELECT CONCAT(lastName, ', ', firstName) AS Employee\_Name FROM EMPLOYEE;**

**b. Employee Position; [FIRST\_NAME] [LAST\_NAME] Position: [TITLE] e.g. Michael Smith Position: President**

**SELECT CONCAT(firstName, ' ', lastName, ' Position: ', title) AS Employee\_Details FROM EMPLOYEE;**

1. **Advanced Tasks**

**13. Consider this text “Fear leads to anger; anger leads to hatred; hatred leads to conflict; conflict leads to suffering”. Write a statement to return the same text but 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;**

**14. The data in the CUSTOMER table that holds the FED\_ID is currently held in two different formats. Either nnn-nnnnnn or nn-nnnnnn (where n is a number). The bank wishes to standardise the format so that all values in FED\_ID are stored as nnnnnnnn. Write an update statement to do this.**

**UPDATE CUSTOMER SET fedID = REPLACE(REPLACE(fedID, '-', ''), ' ', '');**

**15. Write a query to return the year portion of the account transaction date and the number of transactions that took place in each year. The output should look similar to:**

**SELECT YEAR(transDate) AS Year, COUNT(\*) AS Number\_Of\_Transactions FROM ACC\_TRANSACTION GROUP BY YEAR(transDate);**

**16. There have been some inconsistencies in the way data has been entered into the system. As SQL is case sensitive this has caused problems when searching for data. To prevent problems, update the EMPLOYEE table to store everyone’s job title in Uppercase.**

**UPDATE EMPLOYEE SET title = UPPER(title);**

**SELECT COUNT(\*) AS Count,**

**CASE**

**WHEN title LIKE '%TELLER%' THEN 'Cashier'**

**ELSE 'Other'**

**END AS JobTitle**

**FROM EMPLOYEE**

**GROUP BY JobTitle;**

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

**SELECT custID, SUM(availableBalance) AS Total\_Balance FROM ACCOUNT GROUP BY custID HAVING SUM(availableBalance) < 5000;**

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

**SELECT branchID, COUNT(\*) AS Total\_Staff FROM EMPLOYEE GROUP BY branchID;**

**19. Using only the ACCOUNTS Table; produce a report showing the total number of accounts which have the product codes CHK and SAV. Your output should display the following text for each product code:**

**SELECT COUNT(\*) AS Number\_Of\_Accounts,**

**CASE**

**WHEN productCode = 'CHK' THEN 'Checking Account'**

**WHEN productCode = 'SAV' THEN 'Savings Account'**

**END AS Product\_Type**

**FROM ACCOUNT**

**WHERE productCode IN ('CHK', 'SAV')**

**GROUP BY productCode;**