|  |  |  |
| --- | --- | --- |
|  | Pimpri Chinchwad Education Trust’s  **Pimpri Chinchwad College of Engineering**  An Autonomous Institute  (Permanently affiliated to Savitribai Phule Pune University) |  |
| SEMESTER-IV |
| Assignment 5 | | |

**Assignment 5: SQL Grouping and database views**

**Database Schema (Banking System)**

**Customers**

**customer\_id (INT, PRIMARY KEY)**

**name (VARCHAR)**

**dob (DATE)**

**city (VARCHAR)**

**account\_type (VARCHAR)**

**Accounts**

**account\_id (INT, PRIMARY KEY)**

**customer\_id (INT, FOREIGN KEY REFERENCES Customers(customer\_id))**

**balance (DECIMAL)**

**account\_type (VARCHAR)**

**branch\_id (INT, FOREIGN KEY REFERENCES Branches(branch\_id))**

**Transactions**

**transaction\_id (INT, PRIMARY KEY)**

**account\_id (INT, FOREIGN KEY REFERENCES Accounts(account\_id))**

**amount (DECIMAL)**

**transaction\_type (VARCHAR) -- ('Deposit', 'Withdrawal', 'Transfer')**

**transaction\_date (DATE)**

**Branches**

**branch\_id (INT, PRIMARY KEY)**

**branch\_name (VARCHAR)**

**city (VARCHAR)**

**Loans**

**loan\_id (INT, PRIMARY KEY)**

**customer\_id (INT, FOREIGN KEY REFERENCES Customers(customer\_id))**

**amount (DECIMAL)**

**loan\_type (VARCHAR) -- ('Home', 'Car', 'Personal')**

**status (VARCHAR) -- ('Approved', 'Pending', 'Rejected')**

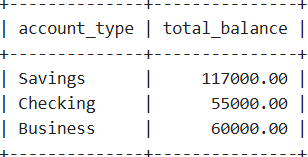
**Solve following queries :**

1. Retrieve the total balance for each account type.

SELECT account\_type, SUM(balance) AS total\_balance

FROM Accounts\_180

GROUP BY account\_type;



1. Count the number of accounts in each branch.

SELECT branch\_id, COUNT(account\_id) AS total\_accounts

FROM Accounts\_180

GROUP BY branch\_id;

A white paper with black text

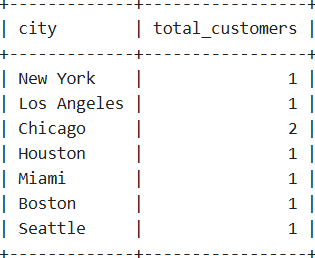
AI-generated content may be incorrect.

1. Find the number of customers in each city.

SELECT city, COUNT(customer\_id) AS total\_customers

FROM Customers\_180

GROUP BY city;



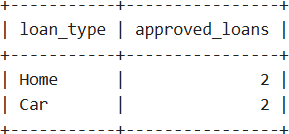
1. Show the number of loans approved per loan type.

SELECT loan\_type, COUNT(loan\_id) AS approved\_loans

FROM Loans\_180

WHERE status = 'Approved'

GROUP BY loan\_type;



1. Find the total number of transactions for each transaction type.

SELECT transaction\_type, COUNT(transaction\_id) AS total\_transactions

FROM Transactions\_180

GROUP BY transaction\_type;

A close-up of a white background

AI-generated content may be incorrect.

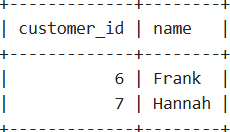
1. Show the customers who do not have a loan.

**SELECT c.customer\_id, c.name**

**FROM Customers\_180 c**

**LEFT JOIN Loans\_180 l ON c.customer\_id = l.customer\_id**

**WHERE l.loan\_id IS NULL;**

****

1. Retrieve customers who have accounts in more than one branch.

**SELECT customer\_id, COUNT(DISTINCT branch\_id) AS branch\_count**

**FROM Accounts\_180**

**GROUP BY customer\_id**

**HAVING COUNT(DISTINCT branch\_id) > 1;**

**A close up of a grid

AI-generated content may be incorrect.**

1. Find all accounts that have not had any deposits in the last 3 months.

SELECT a.account\_id

FROM Accounts\_180 a

LEFT JOIN Transactions\_180 t ON a.account\_id = t.account\_id

AND t.transaction\_type = 'Deposit'

AND t.transaction\_date >= CURRENT\_DATE - INTERVAL 3 MONTH

WHERE t.transaction\_id IS NULL;

A white paper with black numbers

AI-generated content may be incorrect.

1. Display the account types where the total balance is below ₹25,000.

**SELECT account\_type, SUM(balance) AS total\_balance**

**FROM Accounts\_180**

**GROUP BY account\_type**

**HAVING SUM(balance) < 25000;**

**-------> Empty Set**

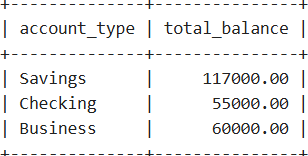
1. Find the total balance per account type where the balance is above ₹50,000.

**SELECT account\_type, SUM(balance) AS total\_balance**

**FROM Accounts\_180**

**GROUP BY account\_type**

**HAVING SUM(balance) > 50000;**

****

1. Retrieve the count of transactions made on dates where more than 5 transactions occurred.

**SELECT transaction\_date, COUNT(transaction\_id) AS total\_transactions**

**FROM Transactions\_180**

**GROUP BY transaction\_date**

**HAVING COUNT(transaction\_id) > 5;**

**------> Empty Set**

1. Find the top 3 transaction days with the highest total transaction amount.

SELECT transaction\_date, SUM(amount) AS total\_amount

FROM Transactions\_180

GROUP BY transaction\_date

ORDER BY total\_amount DESC

LIMIT 3;

A white paper with black text

AI-generated content may be incorrect.

1. Find customers who have a loan but no account in the bank.

SELECT l.customer\_id, c.name

FROM Loans\_180 l

LEFT JOIN Accounts\_180 a ON l.customer\_id = a.customer\_id

JOIN Customers\_180 c ON l.customer\_id = c.customer\_id

WHERE a.account\_id IS NULL;

**A close up of a white background

AI-generated content may be incorrect.**