**Deliverable 5**

|  |  |  |  |
| --- | --- | --- | --- |
| **Individual Contribution** | | | |
| **CWID** | **Name** | **Contribution (description)** | **Percent Contribution** |
| **A20580060** | **Anshul Dani** | Created the Loom video, demonstrating the application's features and functionality. Provided narration for complex queries and implementation. Also contributed to finalizing error handling, documentation, and code organization. | **33.3%** |
| **A20576599** | **Sanjana Waghray** | Responsible for implementing and testing SQL queries, including advanced features like set operations, subqueries, and OLAP queries. Assisted in error handling and ensuring meaningful query results. | **33.3%** |
| **A20545641** | **Snehal Phadatare** | Designed and developed the user interface, including interactive elements like menus, buttons, and icons. Contributed to documentation and ensured the application was intuitive and accessible. | **33.3%** |

**Queries for Expense Manager data**

1. People who are from New York

SELECT Name, Location FROM User WHERE Location = 'New York';

A black and grey background

Description automatically generated with medium confidence

1. Find the category with the most expenses

SELECT c.CategoryName, SUM(e.Amount) AS TotalSpending

FROM Expense e

JOIN Category c ON e.CategoryID = c.CategoryID

GROUP BY c.CategoryName

ORDER BY TotalSpending DESC

LIMIT 1;

A screenshot of a computer

Description automatically generated

1. Transactions from a particular payment method

SELECT \* FROM Transaction WHERE PaymentMethodID = 2;

A black screen with white text

Description automatically generated

1. Amount spent in each category

SELECT c.CategoryName, SUM(e.Amount) AS TotalAmountSpent

FROM Expense e

JOIN Category c ON e.CategoryID = c.CategoryID

GROUP BY c. CategoryName;

A black and white table with white text

Description automatically generated

1. Most expensive expense for each user

SELECT e.UserID, MAX(e.Amount) AS MaxExpense

FROM Expense e

GROUP BY e.UserID;

A screenshot of a black and white screen

Description automatically generated

1. Number of transactions for each payment method

SELECT pm.MethodName, COUNT(t.TransactionID) AS TransactionCount

FROM PaymentMethod pm

JOIN Transaction t ON pm.PaymentMethodID = t.PaymentMethodID

GROUP BY pm.MethodName;

A black rectangular sign with white text

Description automatically generated

1. Average expense for each category

SELECT c.CategoryName, AVG(e.Amount) AS AverageExpense

FROM Expense e

JOIN Category c ON e.CategoryID = c.CategoryID

GROUP BY c.CategoryName;

A screenshot of a black and white table

Description automatically generated

1. Most recent expense for each user

SELECT e.UserID, e.Amount, e.ExpenseDate

FROM Expense e

WHERE e.ExpenseDate = (

SELECT MAX(ExpenseDate)

FROM Expense

WHERE UserID = e.UserID

);

A screenshot of a black and white table

Description automatically generated

1. Total spending for each month

SELECT YEAR(e.ExpenseDate) AS Year, MONTH(e.ExpenseDate) AS Month, SUM(e.Amount) AS TotalSpending

FROM Expense e

GROUP BY YEAR(e.ExpenseDate), MONTH(e.ExpenseDate)

ORDER BY Year DESC, Month DESC;

A screenshot of a black and white screen

Description automatically generated

1. Users who have spent on more than 2 categories

SELECT u.Name

FROM User u

JOIN Expense e ON u.UserID = e.UserID

GROUP BY u.UserID

HAVING COUNT(DISTINCT e.CategoryID) > 2;

**NONE**

1. User who spent on a specific category

SELECT UserID, u.Name, u.Location

FROM User u

WHERE u.UserID IN (

SELECT e.UserID

FROM Expense e

WHERE e.CategoryID = (

SELECT CategoryID FROM Category WHERE CategoryName = 'Travel'

)

);

A screenshot of a computer

Description automatically generated

1. Users with above average spending

SELECT Name, TotalExpenses

FROM User

WHERE TotalExpenses > (

SELECT AVG(TotalExpenses)

FROM User

);

A screenshot of a graph

Description automatically generated

1. Total spending per user

WITH UserExpenses AS (

SELECT e.UserID, SUM(e.Amount) AS TotalSpending

FROM Expense e

GROUP BY e.UserID

)

SELECT u.Name, u.Location, ue.TotalSpending

FROM User u

JOIN UserExpenses ue ON u.UserID = ue.UserID

ORDER BY ue.TotalSpending DESC;

A table with numbers and names

Description automatically generated

1. Most popular payment method

SELECT pm.MethodName, COUNT(t.PaymentMethodID) AS UsageCount

FROM PaymentMethod pm

JOIN Transaction t ON pm.PaymentMethodID = t.PaymentMethodID

GROUP BY pm.MethodName

ORDER BY UsageCount DESC

LIMIT 1;

A screen shot of a card

Description automatically generated

1. Cumulative spending for each category

SELECT

c.CategoryName,

SUM(e.Amount) AS TotalSpending,

SUM(SUM(e.Amount)) OVER (ORDER BY SUM(e.Amount) DESC) AS CumulativeSpending

FROM Expense e

JOIN Category c ON e.CategoryID = c.CategoryID

GROUP BY c.CategoryName

ORDER BY TotalSpending DESC;

A screenshot of a black and white table

Description automatically generated

1. Users and categories they have spent on

SELECT u.Name AS UserName, c.CategoryName, SUM(e.Amount) AS TotalCategorySpending

FROM User u

JOIN Expense e ON u.UserID = e.UserID

JOIN Category c ON e.CategoryID = c.CategoryID

GROUP BY u.Name, c.CategoryName

ORDER BY u.Name, TotalCategorySpending DESC;

A black and white table with white text

Description automatically generated

1. Monthly spending per user

SELECT

u.Name AS UserName,

DATE\_FORMAT(e.ExpenseDate, '%Y-%m') AS Month,

SUM(e.Amount) AS MonthlySpending

FROM User u

JOIN Expense e ON u.UserID = e.UserID

GROUP BY u.Name, Month

ORDER BY u.Name, Month;

A black and white screen with white text

Description automatically generated

1. Category spending as a percentage of total expenses

WITH CategorySpending AS (

SELECT c.CategoryName, SUM(e.Amount) AS TotalCategorySpending

FROM Expense e

JOIN Category c ON e.CategoryID = c.CategoryID

GROUP BY c.CategoryName

),

TotalSpending AS (

SELECT SUM(TotalCategorySpending) AS GrandTotal

FROM CategorySpending

)

SELECT cs.CategoryName,

cs.TotalCategorySpending,

ROUND((cs.TotalCategorySpending / ts.GrandTotal) \* 100, 2) AS PercentageOfTotal

FROM CategorySpending cs, TotalSpending ts

ORDER BY PercentageOfTotal DESC;

A screenshot of a computer screen

Description automatically generated

1. Payment method used most often by each user

SELECT u.Name, pm.MethodName, COUNT(t.TransactionID) AS UsageCount

FROM Transaction t

JOIN PaymentMethod pm ON t.PaymentMethodID = pm.PaymentMethodID

JOIN User u ON t.UserID = u.UserID

GROUP BY u.UserID, pm.MethodName

HAVING COUNT(t.TransactionID) = (

SELECT MAX(UsageCount)

FROM (

SELECT COUNT(t1.TransactionID) AS UsageCount

FROM Transaction t1

WHERE t1.UserID = u.UserID

GROUP BY t1.PaymentMethodID

) AS SubQuery

)

ORDER BY u.Name;

A table with numbers and names

Description automatically generated with medium confidence

**THE LOOM VIDEO WAS NOT BEING DOWNLOADED SO HERE IS THE LINK TO THE VIDEO:**

https://www.loom.com/share/7c9451878339429491e6561db40033d9?sid=31c6ebb0-6242-4d8f-b4c6-0ba85c8bd260