**USED THE DATABASE CREATED BY THE week2.txt file**

**Part 1: Retrieving Data with SELECT (30 minutes)**

**1.1 retrieve data**

* Write an SQL query to retrieve all data points (columns) from the "Expenses" table.

**SQL COMMAND**

SELECT \* FROM Expenses;

**Data shown:**

| **# expense\_id** | **amount** | **date** | **category** |
| --- | --- | --- | --- |
| **1** | **90.00** | **2021-02-02** | **Entertainment** |
| **2** | **41.00** | **2021-02-16** | **Transportation** |
| **3** | **92.00** | **2023-02-05** | **Other** |
| **4** | **87.00** | **2024-02-25** | **Groceries** |
| **5** | **34.00** | **2022-06-06** | **Entertainment** |
| **6** | **77.00** | **2024-04-07** | **Transportation** |
| **7** | **77.00** | **2022-08-25** | **Other** |
| **8** | **76.00** | **2023-05-13** | **Groceries** |
| **9** | **67.00** | **2023-07-02** | **Entertainment** |
| **10** | **10.00** | **2023-09-15** | **Transportation** |
| **11** | **107.00** | **2022-05-04** | **Other** |
| **12** | **47.00** | **2022-07-21** | **Groceries** |
| **13** | **52.00** | **2022-11-22** | **Entertainment** |
| **14** | **80.00** | **2023-06-16** | **Transportation** |
| **15** | **67.00** | **2023-01-30** | **Other** |
| **16** | **60.00** | **2022-11-24** | **Groceries** |
| **17** | **57.00** | **2022-10-13** | **Entertainment** |
| **18** | **52.00** | **2022-03-21** | **Transportation** |
| **19** | **96.00** | **2020-08-07** | **Other** |
| **20** | **63.00** | **2022-12-02** | **Groceries** |

**1.2 specific columns**

Modify your query to select only specific columns relevant to your analysis. For example, you might choose "date," "category," and "amount" to analyze spending patterns by category and date

**SQL COMMAND**

SELECT date, category, amount FROM Expenses;

**Data shown:**

| **# date** | **category** | **amount** |
| --- | --- | --- |
| **2021-02-02** | **Entertainment** | **90.00** |
| **2021-02-16** | **Transportation** | **41.00** |
| **2023-02-05** | **Other** | **92.00** |
| **2024-02-25** | **Groceries** | **87.00** |
| **2022-06-06** | **Entertainment** | **34.00** |
| **2024-04-07** | **Transportation** | **77.00** |
| **2022-08-25** | **Other** | **77.00** |
| **2023-05-13** | **Groceries** | **76.00** |
| **2023-07-02** | **Entertainment** | **67.00** |
| **2023-09-15** | **Transportation** | **10.00** |
| **2022-05-04** | **Other** | **107.00** |
| **2022-07-21** | **Groceries** | **47.00** |
| **2022-11-22** | **Entertainment** | **52.00** |
| **2023-06-16** | **Transportation** | **80.00** |
| **2023-01-30** | **Other** | **67.00** |
| **2022-11-24** | **Groceries** | **60.00** |
| **2022-10-13** | **Entertainment** | **57.00** |
| **2022-03-21** | **Transportation** | **52.00** |
| **2020-08-07** | **Other** | **96.00** |
| **2022-12-02** | **Groceries** | **63.00** |

**1.3 filtering by date range**

Filtering by Date Range: Write a query to retrieve expenses charged between a specific date range (e.g., January 1, 2021, to December 15, 2024). Remember to use the appropriate data type for the "date" column when specifying the date range in your query.

**SQL COMMAND**

SELECT \* FROM Expenses

WHERE date BETWEEN

'2021-01-01' AND

'2024-12-15';

**Data shown:**

| **# expense\_id** | **amount** | **date** | **category** |
| --- | --- | --- | --- |
| **1** | **90.00** | **2021-02-02** | **Entertainment** |
| **2** | **41.00** | **2021-02-16** | **Transportation** |
| **3** | **92.00** | **2023-02-05** | **Other** |
| **4** | **87.00** | **2024-02-25** | **Groceries** |
| **5** | **34.00** | **2022-06-06** | **Entertainment** |
| **6** | **77.00** | **2024-04-07** | **Transportation** |
| **7** | **77.00** | **2022-08-25** | **Other** |
| **8** | **76.00** | **2023-05-13** | **Groceries** |
| **9** | **67.00** | **2023-07-02** | **Entertainment** |
| **10** | **10.00** | **2023-09-15** | **Transportation** |
| **11** | **107.00** | **2022-05-04** | **Other** |
| **12** | **47.00** | **2022-07-21** | **Groceries** |
| **13** | **52.00** | **2022-11-22** | **Entertainment** |
| **14** | **80.00** | **2023-06-16** | **Transportation** |
| **15** | **67.00** | **2023-01-30** | **Other** |
| **16** | **60.00** | **2022-11-24** | **Groceries** |
| **17** | **57.00** | **2022-10-13** | **Entertainment** |
| **18** | **52.00** | **2022-03-21** | **Transportation** |
| **20** | **63.00** | **2022-12-02** | **Groceries** |

**Part2: filtering with the where clause**

**2.1:filtering by category**

Filtering by Category: Write a query to find all expenses belonging to a specific category (e.g., "Entertainment").

**SQL COMMAND**

SELECT \* FROM Expenses

WHERE category =

'Entertainment';

**Data shown:**

| **# expense\_id** | **amount** | **date** | **category** |
| --- | --- | --- | --- |
| **1** | **90.00** | **2021-02-02** | **Entertainment** |
| **5** | **34.00** | **2022-06-06** | **Entertainment** |
| **9** | **67.00** | **2023-07-02** | **Entertainment** |
| **13** | **52.00** | **2022-11-22** | **Entertainment** |
| **17** | **57.00** | **2022-10-13** | **Entertainment** |

**2.2 Filtering with Comparison Operators:**

2.2 Filtering with Comparison Operators: Find expenses with an amount greater than a certain value (e.g., $50).

**SQL COMMAND**

SELECT \* FROM Expenses

WHERE amount > 50;

**Data shown:**

| **# expense\_id** | **amount** | **date** | **category** |
| --- | --- | --- | --- |
| **1** | **90.00** | **2021-02-02** | **Entertainment** |
| **3** | **92.00** | **2023-02-05** | **Other** |
| **4** | **87.00** | **2024-02-25** | **Groceries** |
| **6** | **77.00** | **2024-04-07** | **Transportation** |
| **7** | **77.00** | **2022-08-25** | **Other** |
| **8** | **76.00** | **2023-05-13** | **Groceries** |
| **9** | **67.00** | **2023-07-02** | **Entertainment** |
| **11** | **107.00** | **2022-05-04** | **Other** |
| **13** | **52.00** | **2022-11-22** | **Entertainment** |
| **14** | **80.00** | **2023-06-16** | **Transportation** |
| **15** | **67.00** | **2023-01-30** | **Other** |
| **16** | **60.00** | **2022-11-24** | **Groceries** |
| **17** | **57.00** | **2022-10-13** | **Entertainment** |
| **18** | **52.00** | **2022-03-21** | **Transportation** |
| **19** | **96.00** | **2020-08-07** | **Other** |
| **20** | **63.00** | **2022-12-02** | **Groceries** |

**2.3 Combining Filters (AND):**

Combining Filters (AND): Refine your query to find expenses that meet multiple criteria. For example, you might search for expenses greater than $75 AND belonging to the "Food" category.

**SQL COMMAND**

SELECT \* FROM Expenses

WHERE amount > 75 AND

category = 'food';

**Data shown:**

| **# expense\_id** | **amount** | **date** | **category** |
| --- | --- | --- | --- |
| **Null** | **Null** | **Null** | **Null** |

**2.4 Combining Filters (OR):**

Combining Filters (OR): Modify your query to find expenses belonging to one category or another (e.g., "Transportation" OR "Groceries").

**SQL COMMAND**

SELECT \* FROM Expenses

WHERE category =

'Transportation' OR category =

'Groceries';

**Data shown:**

| **# expense\_id** | **amount** | **date** | **category** |
| --- | --- | --- | --- |
| **2** | **41.00** | **2021-02-16** | **Transportation** |
| **4** | **87.00** | **2024-02-25** | **Groceries** |
| **6** | **77.00** | **2024-04-07** | **Transportation** |
| **8** | **76.00** | **2023-05-13** | **Groceries** |
| **10** | **10.00** | **2023-09-15** | **Transportation** |
| **12** | **47.00** | **2022-07-21** | **Groceries** |
| **14** | **80.00** | **2023-06-16** | **Transportation** |
| **16** | **60.00** | **2022-11-24** | **Groceries** |
| **18** | **52.00** | **2022-03-21** | **Transportation** |
| **20** | **63.00** | **2022-12-02** | **Groceries** |

**2.5 Filtering with NOT:**

Filtering with NOT: Write a query to display expenses unrelated to a specific category (e.g., "Rent").

**SQL COMMAND**

SELECT \* FROM Expenses

WHERE category != 'Rent';

**Data shown:**

| **# expense\_id** | **amount** | **date** | **category** |
| --- | --- | --- | --- |
| **1** | **90.00** | **2021-02-02** | **Entertainment** |
| **2** | **41.00** | **2021-02-16** | **Transportation** |
| **3** | **92.00** | **2023-02-05** | **Other** |
| **4** | **87.00** | **2024-02-25** | **Groceries** |
| **5** | **34.00** | **2022-06-06** | **Entertainment** |
| **6** | **77.00** | **2024-04-07** | **Transportation** |
| **7** | **77.00** | **2022-08-25** | **Other** |
| **8** | **76.00** | **2023-05-13** | **Groceries** |
| **9** | **67.00** | **2023-07-02** | **Entertainment** |
| **10** | **10.00** | **2023-09-15** | **Transportation** |
| **11** | **107.00** | **2022-05-04** | **Other** |
| **12** | **47.00** | **2022-07-21** | **Groceries** |
| **13** | **52.00** | **2022-11-22** | **Entertainment** |
| **14** | **80.00** | **2023-06-16** | **Transportation** |
| **15** | **67.00** | **2023-01-30** | **Other** |
| **16** | **60.00** | **2022-11-24** | **Groceries** |
| **17** | **57.00** | **2022-10-13** | **Entertainment** |
| **18** | **52.00** | **2022-03-21** | **Transportation** |
| **19** | **96.00** | **2020-08-07** | **Other** |
| **20** | **63.00** | **2022-12-02** | **Groceries** |

**Part 3: Sorting Retrieved Data (45 minutes)**

**3.1 Sorting by Amount:**

Sorting by Amount: Write a query to display all expenses sorted by amount in a specific order (e.g., descending order for highest to lowest spending).

**SQL COMMAND**

SELECT \* FROM Expenses

ORDER BY amount DESC;

**Data shown:**

| **# expense\_id** | **amount** | **date** | **category** |
| --- | --- | --- | --- |
| **11** | **107.00** | **2022-05-04** | **Other** |
| **19** | **96.00** | **2020-08-07** | **Other** |
| **3** | **92.00** | **2023-02-05** | **Other** |
| **1** | **90.00** | **2021-02-02** | **Entertainment** |
| **4** | **87.00** | **2024-02-25** | **Groceries** |
| **14** | **80.00** | **2023-06-16** | **Transportation** |
| **6** | **77.00** | **2024-04-07** | **Transportation** |
| **7** | **77.00** | **2022-08-25** | **Other** |
| **8** | **76.00** | **2023-05-13** | **Groceries** |
| **9** | **67.00** | **2023-07-02** | **Entertainment** |
| **15** | **67.00** | **2023-01-30** | **Other** |
| **20** | **63.00** | **2022-12-02** | **Groceries** |
| **16** | **60.00** | **2022-11-24** | **Groceries** |
| **17** | **57.00** | **2022-10-13** | **Entertainment** |
| **13** | **52.00** | **2022-11-22** | **Entertainment** |
| **18** | **52.00** | **2022-03-21** | **Transportation** |
| **12** | **47.00** | **2022-07-21** | **Groceries** |
| **2** | **41.00** | **2021-02-16** | **Transportation** |
| **5** | **34.00** | **2022-06-06** | **Entertainment** |
| **10** | **10.00** | **2023-09-15** | **Transportation** |

**3.2 Sorting by Date and Category:**

Sorting by Date and Category: Modify your query to sort expenses based on multiple columns. For example, you might sort first by date (descending order) and then by category (ascending order) to see recent spending trends by category.

**SQL COMMAND**

SELECT \* FROM Expenses

ORDER BY date DESC, category

ASC;

**Data shown:**

| **# expense\_id** | **amount** | **date** | **category** |
| --- | --- | --- | --- |
| **6** | **77.00** | **2024-04-07** | **Transportation** |
| 4 | **87.00** | **2024-02-25** | **Groceries** |
| 10 | **10.00** | **2023-09-15** | **Transportation** |
| 9 | **67.00** | **2023-07-02** | **Entertainment** |
| 14 | **80.00** | **2023-06-16** | **Transportation** |
| 8 | **76.00** | **2023-05-13** | **Groceries** |
| 3 | **92.00** | **2023-02-05** | **Other** |
| 15 | **67.00** | **2023-01-30** | **Other** |
| 20 | **63.00** | **2022-12-02** | **Groceries** |
| 16 | **60.00** | **2022-11-24** | **Groceries** |
| 13 | **52.00** | **2022-11-22** | **Entertainment** |
| 17 | **57.00** | **2022-10-13** | **Entertainment** |
| 7 | **77.00** | **2022-08-25** | **Other** |
| 12 | **47.00** | **2022-07-21** | **Groceries** |
| 5 | **34.00** | **2022-06-06** | **Entertainment** |
| 11 | **107.00** | **2022-05-04** | **Other** |
| 18 | **52.00** | **2022-03-21** | **Transportation** |
| 2 | **41.00** | **2021-02-16** | **Transportation** |
| 1 | **90.00** | **2021-02-02** | **Entertainment** |
| 19 | **96.00** | **2020-08-07** | **Other** |

**Part 4: Database Upgrade**

**4.1 Write SQL commands to achieve the following:**

We don't have a table for income yet. Create a table named "Income" with columns for:

income\_id (INT) - Primary Key (auto-increment)

amount (DECIMAL(10,2)) - NOT NULL

date (DATE) - NOT NULL

source (VARCHAR(50)) - NOT NULL

**SQL COMMAND TO CREATE INCOME TABLE**

CREATE TABLE Income (

income\_id int

AUTO\_INCREMENT PRIMARY KEY,

amount DECIMAL(10,2) NOT NULL,

date DATE NOT NULL,

source VARCHAR(50) NOT NULL

);

**4.2 After creating the "Income" table, you realize you also want to track the income category "source" (e.g., "Salary," "Freelance Work").**

**SQL COMMAND**

ALTER TABLE Income

ADD column category

VARCHAR(50);

**4.3 Let's say you decide tracking the income source isn't necessary for now.**

**SQL COMMAND**

ALTER TABLE Income

DROP COLUMN source;

**-- dropping the Income Table**

**SQL COMMAND**

DROP TABLE Income;