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

**Based on the Expense Tracker table you designed in Week 1, which likely includes columns like "expense\_id," "amount," "date," and "category," complete the following tasks:**

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

SELECT \* FROM db.myexpenses1\_cleaned;

**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.

SELECT date, category, amount

FROM myexpenses1\_cleaned;

**1.3 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.

SELECT date, category, amount

FROM myexpenses1\_cleaned

WHERE date BETWEEN '2023-03-01' AND '2023-03-31';

**Part 2: Filtering with WHERE Clause (45 minutes)**

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

SELECT date, category, amount

FROM myexpenses1\_cleaned

WHERE category = 'Friend';

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

SELECT date, category, amount

FROM myexpenses1\_cleaned

WHERE amount > 50;

**2.3 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.

SELECT date, category, amount

FROM myexpenses1\_cleaned

WHERE amount > 75 AND category = 'Friend';

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

SELECT date, category, amount

FROM myexpenses1\_cleaned

WHERE category = 'Friend' OR category = 'alone';

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

SELECT date, category, amount

FROM myexpenses1\_cleaned

WHERE category <> 'Friend';

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

**3.1 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).

SELECT date, category, amount

FROM myexpenses1\_cleaned

ORDER BY amount DESC;

**3.2 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.

SELECT date, category, amount

FROM myexpenses1\_cleaned

ORDER BY date DESC, amount ASC;

**Part 4: Database Upgrade:** Imagine you're tasked by the CIO to expand your Expense Tracker database. Practice creating, modifying, and removing a table to manage spending habits.

* CREATE QUERY

CREATE TABLE expenses (

id INT AUTO\_INCREMENT PRIMARY KEY,

date DATE NOT NULL,

category VARCHAR(50) NOT NULL,

amount DECIMAL(10, 2) NOT NULL,

description VARCHAR(255)

);

* MODIFYING QUERY

ALTER TABLE expenses

ADD COLUMN payment\_method VARCHAR(50);

* REMOVE QUERY

ALTER TABLE expenses

DROP COLUMN payment\_method;

**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

**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").**

* Use ALTER TABLE to add a new column named "category" of type VARCHAR(50).

ALTER TABLE Income

ADD COLUMN category VARCHAR(50);

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

* Use ALTER TABLE again to remove the "source" column from the "Income" table.

Imagine you no longer need the "Income" table entirely. Experiment how to Use DROP TABLE to permanently remove it from your database.

**Alter table**:

ALTER TABLE Income

DROP COLUMN source;

**Drop Table**:

DROP TABLE Income;