WEEK 2 DATABASE ASSIGNMENT.

```
-- Create the database (adjust the name if needed)
CREATE DATABASE IF NOT EXISTS expense_tracker;

-- Use the expense_tracker database
USE expense_tracker;

-- Create the Expenses table
CREATE TABLE IF NOT EXISTS Expenses (
  expense_id INT PRIMARY KEY AUTO_INCREMENT,
  amount DECIMAL(10,2) NOT NULL,
  date DATE NOT NULL,
  category VARCHAR(50) NOT NULL
);

-- Function to generate random date within a specific range (modify as needed)
DELIMITER //

CREATE FUNCTION GetRandomDate(startDate DATE, endDate DATE)
RETURNS DATE
READS SQL DATA
DETERMINISTIC
BEGIN
  DECLARE randomDays INT;
  SET randomDays = FLOOR(RAND() * (DATEDIFF(endDate, startDate) + 1));

  RETURN DATE_ADD(startDate, INTERVAL randomDays DAY);

END; //


DELIMITER ;


-- Stored Procedure to insert sample data with random dates and categories (categories can be
modified)

DELIMITER //


CREATE PROCEDURE InsertSampleData()

BEGIN

  DECLARE counter INT DEFAULT 1;


  WHILE counter <= 20 DO

    INSERT INTO Expenses (amount, date, category)

    VALUES (FLOOR(10 + RAND() * 100),

            GetRandomDate(DATE_SUB(CURDATE(), INTERVAL 4 YEAR), CURDATE()),  -- Random date within
the last 4 years

            CASE WHEN counter % 4 = 0 THEN 'Groceries'

                 WHEN counter % 4 = 1 THEN 'Entertainment'

                 WHEN counter % 4 = 2 THEN 'Transportation'

                 ELSE 'Other'

            END);

    SET counter = counter + 1;

  END WHILE;
```

```
END; //

DELIMITER ;

-- Call the procedure to insert sample data
CALL InsertSampleData();

-- Drop the functions and procedures if they are no longer needed
DROP PROCEDURE IF EXISTS InsertSampleData;
DROP FUNCTION IF EXISTS GetRandomDate;

-- Part 1: Retrieving Data with SELECT
-- 1.1 Retrieving all expenses
 SELECT * FROM expenses;

-- 1.2 Retrieving specififc columns
 SELECT date, category, amount FROM Expenses;

-- 1.3 Filtering by date range
 SELECT date, category, amount
 FROM Expenses
 WHERE date BETWEEN '2021-01-01' AND '2024-12-15';

-- Part 2: Filtering with WHERE Clause
-- 2.1 Filtering by category
```

```sql
 SELECT * FROM expenses
 WHERE category='groceries'

-- 2.2  Filtering with Comparison Operators
 SELECT date, category, amount
 FROM Expenses
 WHERE amount > 50;


-- 2.3 Combining Filters (AND)
SELECT * FROM Expenses
 WHERE amount > 75
AND category = 'Food';

-- 2.4 ombining Filters (OR)
-- SELECT *
-- FROM expenses
-- WHERE category = 'Transportation'
-- OR category = 'Groceries';

-- 2.5 Filtering with NOT
SELECT * FROM expenses
WHERE category <> 'Rent';

-- Part 3: Sorting Retrieved Data
-- 3.1 Sorting by Amount
SELECT * FROM expenses
 ORDER BY amount DESC;

-- 3.2 Sorting by Date and Category
 SELECT * FROM expenses
 ORDER BY date DESC, category ASC;

-- Part 4: Database Upgrade
--  Write SQL commands to achieve the following
 CREATE TABLE income(
 income_id INT,
 amount DECIMAL(10,2),
 date DATE,
 source VARCHAR(50));

-- 4.2 After creating the "Income" table, you realize you also want to track the income category "source" (e.g., "Salary," "Freelance Work").
 ALTER TABLE Income
 ADD category VARCHAR(50);

-- 4.3 Let's say you decide tracking the income source isn't necessary for now.
 ALTER TABLE Income
 DROP COLUMN source;

-- Dropping table income
```

```
DROP TABLE Income;
```