-- creating 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;

/\* data from expenses, amount,date, and category in ascending order\*/

select \* from expenses;

select amount, date, category

from expenses

order by date asc;

/\* data from 01/01/2021 to 15/12/2024 \*/

select date, category, amount

from expenses

where date(date) between '2021-01-01' and '2024-12-15'

order by date asc;

/\* expenses filtered by specific category \*/

select date, category, amount

from expenses

where category = 'Transportation'

order by date asc;

/\* sorting by amount greater than 50 \*/

select \*

from expenses

where amount > 50

order by amount asc, date asc;

/\* sorting by amount greater than 70 and belonging "groceries" category \*/

select \*

from expenses

where amount > 75

and category = 'Groceries'

order by date asc;

/\* modify for transportation or groceries \*/

select \*

from expenses

where category = 'Transportation'

or category = 'Groceries'

order by date asc;

/\* query to display expenses unrelated to groceries \*/

select date, category, amount

from expenses

where category != 'Groceries'

order by date asc;

/\* All expenses in Desc \*/

select \*

from expenses

where amount

order by amount desc;

/\* sorting by multi columns \*/

select date, category, amount

from expenses

order by date desc, category asc;

/\* Database Upgrade \*/

start transaction;

create table if not exists MontlyExpenses (

expense\_id int primary key auto\_increment,

description varchar(100) not null,

amount decimal(10,2) not null,

category varchar(50) not null,

date DATE not null

);

alter table Expenses

add column payment\_method varchar(50);

commit;

/\* table for income \*/

create table if not exists Income (

incime\_id int primary key auto\_increment,

amount decimal(10,2) not null,

date DATE not null,

source varchar(50) not null

);

alter table Income

add column category varchar(50);

commit;

alter table Income

drop column source;

Using Drop Table command will completely erase the entire table, not just the column and if not backup was not initialized then data is all lost.