EXPENSE MANAGEMENT SYSTEM

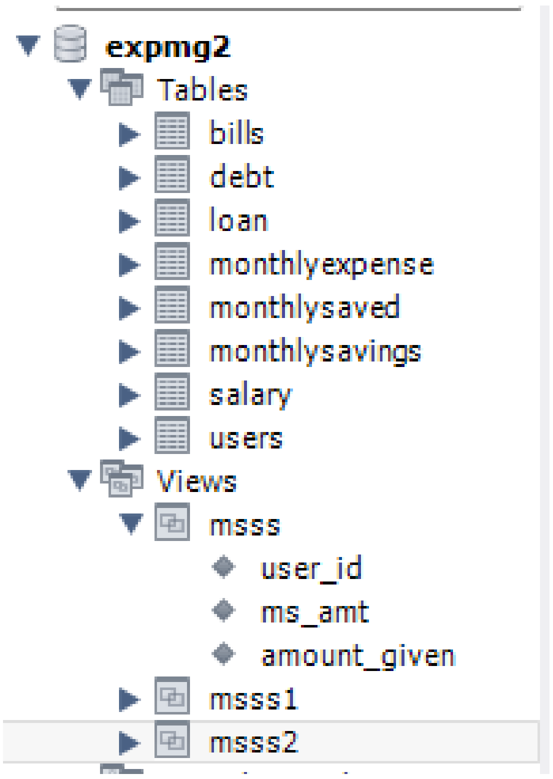
dbms - project

QUERY HANDLIING

Atraya Datta- RA2211026010256
Anvi Tandon-RA2211026010257

CEATION OF DATABASE:

```
CREATE DATABASE expmg2
use database expmg2


-- Create the Users table
CREATE TABLE Users (
   user_id INT PRIMARY KEY AUTO_INCREMENT,
   username VARCHAR(255) NOT NULL,
   password VARCHAR(255) NOT NULL,
   phone_number VARCHAR(20) NOT NULL
);

-- Create the Salary table with user_id column
CREATE TABLE Salary (
   salary_id INT PRIMARY KEY AUTO_INCREMENT,
   user_id INT,
   amount DECIMAL(10, 2) NOT NULL,
   FOREIGN KEY (user_id) REFERENCES Users(user_id)
);

-- Create the MonthlyExpense table with user_id column
CREATE TABLE MonthlyExpense (
   expense_id INT PRIMARY KEY AUTO_INCREMENT,
   user_id INT,
   month_year DATE NOT NULL,
   me_amt DECIMAL(10, 2) NOT NULL,
   FOREIGN KEY (user_id) REFERENCES Users(user_id)
);

-- Create the MonthlySavings table with user_id column
CREATE TABLE MonthlySavings (
   savings_id INT PRIMARY KEY AUTO_INCREMENT,
   user_id INT,
   month_year DATE NOT NULL,
   ms_amt DECIMAL(10, 2) NOT NULL,
   FOREIGN KEY (user_id) REFERENCES Users(user_id)
);

-- Create the Bills table with user_id column
CREATE TABLE Bills (
   bill_id INT PRIMARY KEY AUTO_INCREMENT,
   user_id INT,
   medical DECIMAL(10, 2) NOT NULL,
   electricity DECIMAL(10, 2) NOT NULL,
   food DECIMAL(10, 2) NOT NULL,
   rent DECIMAL(10, 2) NOT NULL,
   grocery DECIMAL(10, 2) NOT NULL,
   personal_expense DECIMAL(10, 2) NOT NULL,
   FOREIGN KEY (user_id) REFERENCES Users(user_id)
);
```



```
▼ expmg2
   ▼ Tables
      ▶ bills
      ▶ debt
      ▶ loan
      ▶ monthlyexpense
      ▶ monthlysaved
      ▶ monthlysavings
      ▶ salary
      ▶ users
   ▼ Views
      ▼ msss
            user_id
            ms_amt
            amount_given
      ▶ msss1
      ▶ msss2
```

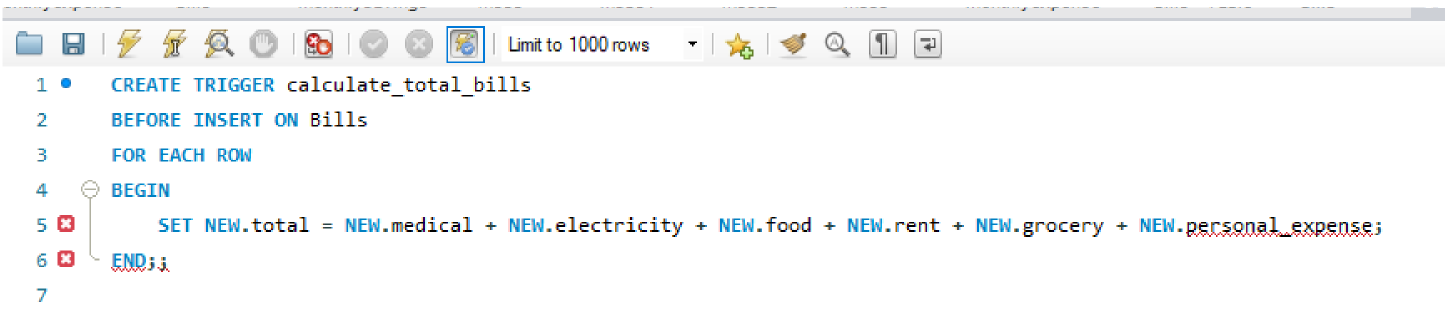
```
-- Create the MonthlySaved
CREATE TABLE MonthlySaved (
 saved_id INT PRIMARY KEY AUTO_INCREMENT,
 user_id INT,
 saving_data DATE NOT NULL,
 saving_amt DECIMAL(10, 2) NOT NULL,
 expense_id INT,
 FOREIGN KEY (user_id) REFERENCES Users(user_id),
 FOREIGN KEY (expense_id)
REFERENCES MonthlyExpense(expense_id)
);

-- Create the Loan table with user_id column
CREATE TABLE Loan (
 loan_id INT PRIMARY KEY AUTO_INCREMENT,
 user_id INT,
 amount_given DECIMAL(10, 2) NOT NULL,
 return_date DATE,
 FOREIGN KEY (user_id) REFERENCES Users(user_id)
);

-- Create the Debt table with user_id column
CREATE TABLE Debt (
 debt_id INT PRIMARY KEY AUTO_INCREMENT,
 user_id INT,
 amount_taken DECIMAL(10, 2) NOT NULL,
 return_date DATE,
 FOREIGN KEY (user_id) REFERENCES Users(user_id)
);
```

ADDING DATA INTO DATABASE:

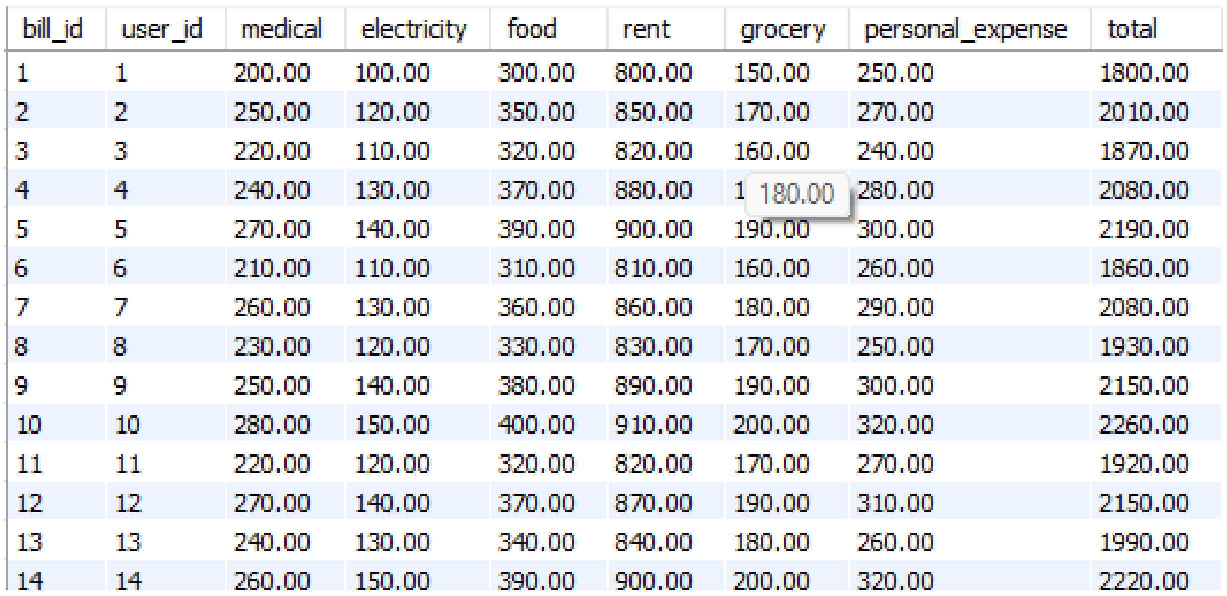
```
INSERT INTO Debt (user_id, amount_taken, return_date) VALUES
(1, 2000.00, '2024-09-01'),
(2, 2500.00, '2024-10-01'),
(3, 2200.00, '2024-11-01'),
(4, 2700.00, '2024-12-01'),
(5, 3000.00, '2025-01-01'),
(6, 2100.00, '2024-09-01'),
(7, 2600.00, '2024-10-01'),
(8, 2300.00, '2024-11-01'),
(9, 2800.00, '2024-12-01'),
(10, 3100.00, '2025-01-01'),
(11, 2200.00, '2024-09-01'),
(12, 2700.00, '2024-10-01'),
(13, 2400.00, '2024-11-01'),
(14, 2900.00, '2024-12-01'),
(15, 3200.00, '2025-01-01'),
(16, 2300.00, '2024-09-01'),
(17, 2800.00, '2024-10-01'),
(18, 2500.00, '2024-11-01'),
(19, 3000.00, '2024-12-01'),
(20, 3300.00, '2025-01-01');

INSERT INTO Loan (user_id, amount_given, return_date) VALUES
(1, 1000.00, '2024-06-01'),
(2, 1500.00, '2024-07-01'),
(3, 1200.00, '2024-08-01'),
(4, 1800.00, '2024-09-01'),
(5, 2000.00, '2024-10-01'),
(6, 1100.00, '2024-06-01'),
(7, 1600.00, '2024-07-01'),
(8, 1300.00, '2024-08-01'),
(9, 1900.00, '2024-09-01'),
(10, 2100.00, '2024-10-01'),
(11, 1200.00, '2024-06-01'),
(12, 1700.00, '2024-07-01'),
(13, 1400.00, '2024-08-01'),
(14, 2000.00, '2024-09-01'),
(15, 2200.00, '2024-10-01'),
(16, 1300.00, '2024-06-01'),
(17, 1800.00, '2024-07-01'),
(18, 1500.00, '2024-08-01'),
(19, 2100.00, '2024-09-01'),
(20, 2300.00, '2024-10-01');

INSERT INTO MonthlySaved (user_id, saving_data, saving_amt, expense_id) VALUES
(1, '2024-01-01', 500.00, 1),
(2, '2024-01-02', 600.00, 2),
(3, '2024-01-03', 450.00, 3),
(4, '2024-01-04', 550.00, 4),
(5, '2024-01-05', 700.00, 5),
(6, '2024-01-06', 550.00, 6),
(7, '2024-01-07', 650.00, 7),
(8, '2024-01-08', 500.00, 8),
(9, '2024-01-09', 600.00, 9),
(10, '2024-01-10', 750.00, 10),
(11, '2024-01-11', 600.00, 11),
IINSERT INTO MonthlySaved (user_id, saving_data, saving_amt, expense_id) VALUES
(1, '2024-01-01', 500.00, 1),
(2, '2024-01-02', 600.00, 2),
(3, '2024-01-03', 450.00, 3),
(4, '2024-01-04', 550.00, 4),
(5, '2024-01-05', 700.00, 5),
(6, '2024-01-06', 550.00, 6),
(7, '2024-01-07', 650.00, 7),
(8, '2024-01-08', 500.00, 8),
(9, '2024-01-09', 600.00, 9),
(10, '2024-01-10', 750.00, 10),
(11, '2024-01-11', 600.00, 11),
(12, '2024-01-12', 700.00, 12),
(13, '2024-01-13', 550.00, 13),
(14, '2024-01-14', 650.00, 14),
(15, '2024-01-15', 800.00, 15),
(16, '2024-01-16', 650.00, 16),
(17, '2024-01-17', 750.00, 17),
(18, '2024-01-18', 600.00, 18),
(19, '2024-01-19', 700.00, 19),
(20, '2024-01-20', 850.00, 20);
INSERT INTO Bills (user_id, medical, electricity, food, rent, grocery, personal_expense) VALUES
(1, 200.00, 100.00, 300.00, 800.00, 150.00, 250.00),
(2, 250.00, 120.00, 350.00, 850.00, 170.00, 270.00),
(3, 220.00, 110.00, 320.00, 820.00, 160.00, 240.00),
(4, 240.00, 130.00, 370.00, 880.00, 180.00, 280.00),
(5, 270.00, 140.00, 390.00, 900.00, 190.00, 300.00),
(6, 210.00, 110.00, 310.00, 810.00, 160.00, 260.00),
(7, 260.00, 130.00, 360.00, 860.00, 180.00, 290.00),
(8, 230.00, 120.00, 330.00, 830.00, 170.00, 250.00),
(9, 250.00, 140.00, 380.00, 890.00, 190.00, 300.00),
(10, 280.00, 150.00, 400.00, 910.00, 200.00, 320.00),
(11, 220.00, 120.00, 320.00, 820.00, 170.00, 270.00),
(12, 270.00, 140.00, 370.00, 870.00, 190.00, 310.00),
(13, 240.00, 130.00, 340.00, 840.00, 180.00, 260.00),
(14, 260.00, 150.00, 390.00, 900.00, 200.00, 320.00),
(15, 290.00, 160.00, 410.00, 920.00, 210.00, 330.00),
(16, 230.00, 130.00, 330.00, 830.00, 180.00, 280.00),
(17, 280.00, 150.00, 380.00, 880.00, 200.00, 330.00),
(18, 250.00, 140.00, 350.00, 850.00, 190.00, 270.00),
(19, 270.00, 160.00, 400.00, 910.00, 210.00, 340.00),
(20, 300.00, 170.00, 420.00, 930.00, 220.00, 350.00);
INSERT INTO MonthlySavings (user_id, month_year, ms_amt) VALUES
(1, '2024-02-01', 1000.00),
(2, '2024-02-02', 1200.00),
(3, '2024-02-03', 900.00),
(4, '2024-02-04', 1100.00),
(5, '2024-02-05', 1300.00),
(6, '2024-02-06', 1100.00),
(7, '2024-02-07', 1300.00),
(8, '2024-02-08', 1000.00),
(9, '2024-02-09', 1200.00),
(10, '2024-02-10', 1400.00),
(11, '2024-02-11', 1200.00),
(12, '2024-02-12', 1400.00),
(13, '2024-02-13', 1100.00),
(14, '2024-02-14', 1300.00),
(15, '2024-02-15', 1500.00),
(16, '2024-02-16', 1300.00),
(17, '2024-02-17', 1500.00),
INSERT INTO MonthlySavings (user_id, month_year, ms_amt) VALUES
(1, '2024-02-01', 1000.00),
(2, '2024-02-02', 1200.00),
(3, '2024-02-03', 900.00),
(4, '2024-02-04', 1100.00),
(5, '2024-02-05', 1300.00),
(6, '2024-02-06', 1100.00),
(7, '2024-02-07', 1300.00),
(8, '2024-02-08', 1000.00),
(9, '2024-02-09', 1200.00),
(10, '2024-02-10', 1400.00),
(11, '2024-02-11', 1200.00),
(12, '2024-02-12', 1400.00),
(13, '2024-02-13', 1100.00),
(14, '2024-02-14', 1300.00),
(15, '2024-02-15', 1500.00),
(16, '2024-02-16', 1300.00),
(17, '2024-02-17', 1500.00),
(18, '2024-02-18', 1200.00),
(19, '2024-02-19', 1400.00),
(20, '2024-02-20', 1600.00);
INSERT INTO MonthlyExpense (user_id, month_year, me_amt) VALUES
(1, '2024-01-01', 1500.00),
(2, '2024-01-02', 2000.00),
(3, '2024-01-03', 1800.00),
(4, '2024-01-04', 2200.00),
(5, '2024-01-05', 2500.00),
(6, '2024-01-06', 1600.00),
(7, '2024-01-07', 2100.00),
(8, '2024-01-08', 1900.00),
(9, '2024-01-09', 2300.00),
(10, '2024-01-10', 2600.00),
(11, '2024-01-11', 1700.00),
(12, '2024-01-12', 2200.00),
(13, '2024-01-13', 2000.00),
(14, '2024-01-14', 2400.00),
(15, '2024-01-15', 2700.00),
(16, '2024-01-16', 1800.00),
(17, '2024-01-17', 2300.00),
(18, '2024-01-18', 2100.00),
(19, '2024-01-19', 2500.00),
(20, '2024-01-20', 2800.00);
I
```

```
NSERT INTO Salary (user_id, amount) VALUES
(1, 5000.00),
(2, 6000.00),
(3, 4500.00),
(4, 5500.00),
(5, 7000.00),
(6, 5200.00),
(7, 6300.00),
(8, 4600.00),
(9, 5700.00),
(10, 7200.00),
(11, 5400.00),
(12, 6500.00),
(13, 4800.00),
(14, 5900.00),
(15, 7400.00),
(16, 5600.00),
(17, 6700.00),
(18, 5000.00),
(19, 6100.00),
(20, 7600.00);
INSERT INTO Users (username, password, phone_number) VALUES
('user1', 'password1', '123-456-7890'),
('user2', 'password2', '987-654-3210'),
('user3', 'password3', '555-555-5555'),
('user4', 'password4', '111-222-3333'),
('user5', 'password5', '444-444-4444'),
('user6', 'password6', '777-777-7777'),
('user7', 'password7', '888-888-8888'),
('user8', 'password8', '999-999-9999'),
('user9', 'password9', '000-000-0000'),
('user10', 'password10', '222-333-4444'),
('user11', 'password11', '555-666-7777'),
('user12', 'password12', '888-999-0000'),
('user13', 'password13', '111-222-3333'),
('user14', 'password14', '444-555-6666'),
('user15', 'password15', '777-888-9999'),
('user16', 'password16', '000-111-2222'),
('user17', 'password17', '333-444-5555'),
('user18', 'password18', '666-777-8888'),
('user19', 'password19', '999-000-1111'),
('user20', 'password20', '222-333-4444');
```

CREATE A TRIGGER TO CALCULATE THE TOTAL OF THE BILLS

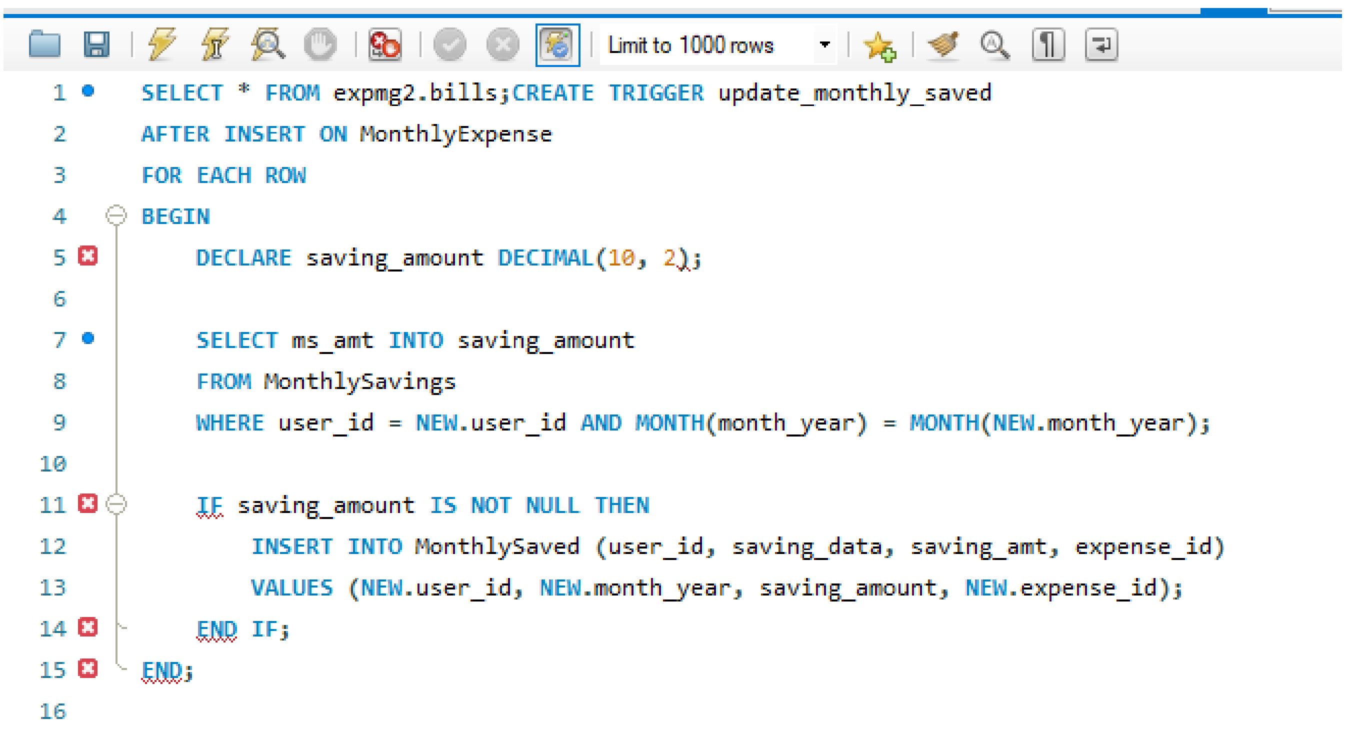


```sql
CREATE TRIGGER calculate_total_bills
BEFORE INSERT ON Bills
FOR EACH ROW
BEGIN
    SET NEW.total = NEW.medical + NEW.electricity + NEW.food + NEW.rent + NEW.grocery + NEW.personal_expense;
END;;
```



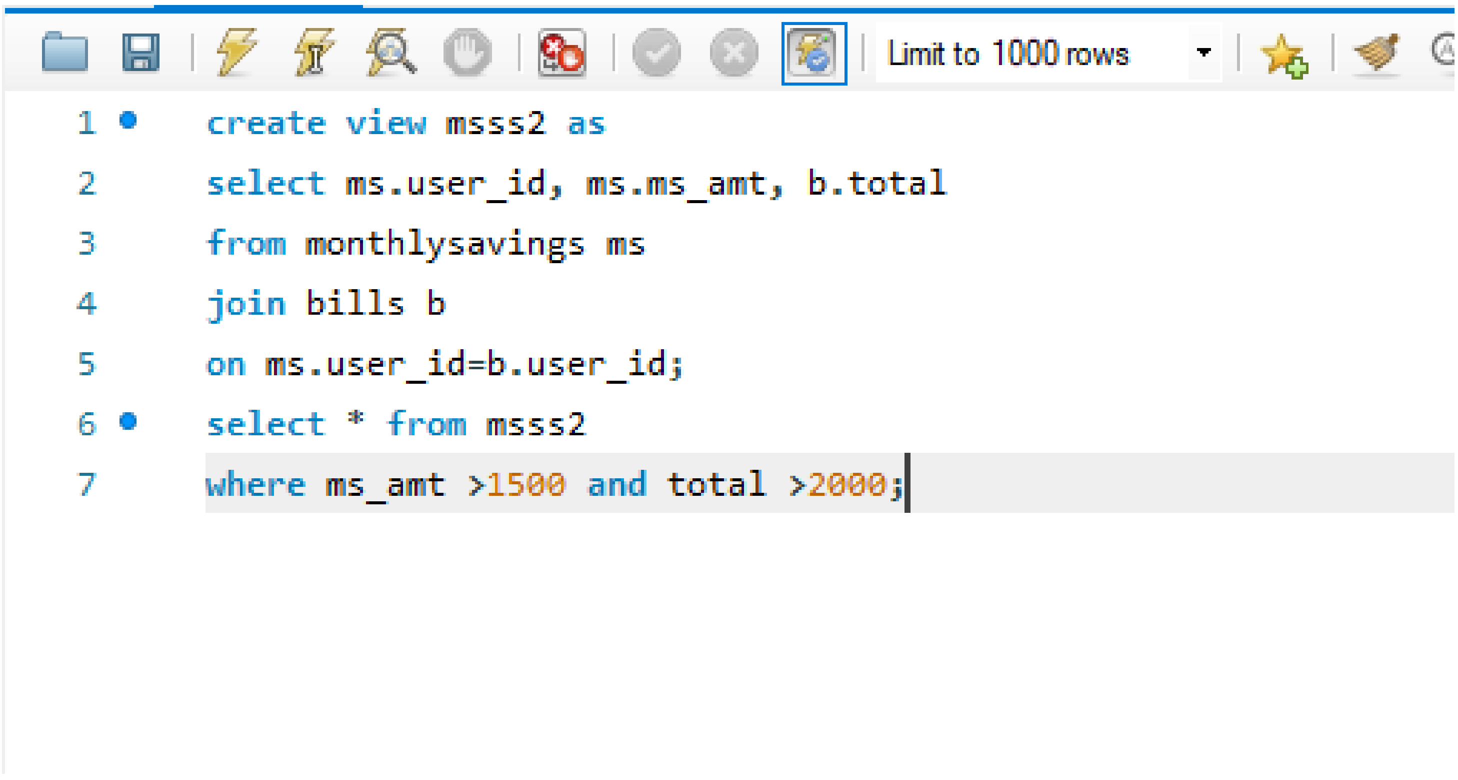
bill_id	user_id	medical	electricity	food	rent	grocery	personal_expense	total
1	1	200.00	100.00	300.00	800.00	150.00	250.00	1800.00
2	2	250.00	120.00	350.00	850.00	170.00	270.00	2010.00
3	3	220.00	110.00	320.00	820.00	160.00	240.00	1870.00
4	4	240.00	130.00	370.00	880.00	180.00	280.00	2080.00
5	5	270.00	140.00	390.00	900.00	190.00	300.00	2190.00
6	6	210.00	110.00	310.00	810.00	160.00	260.00	1860.00
7	7	260.00	130.00	360.00	860.00	180.00	290.00	2080.00
8	8	230.00	120.00	330.00	830.00	170.00	250.00	1930.00
9	9	250.00	140.00	380.00	890.00	190.00	300.00	2150.00
10	10	280.00	150.00	400.00	910.00	200.00	320.00	2260.00
11	11	220.00	120.00	320.00	820.00	170.00	270.00	1920.00
12	12	270.00	140.00	370.00	870.00	190.00	310.00	2150.00
13	13	240.00	130.00	340.00	840.00	180.00	260.00	1990.00
14	14	260.00	150.00	390.00	900.00	200.00	320.00	2220.00

CREATE A TRIGGER TO UPDATE THE MONTHLY SAVED TABLE AFTER THE MONEY IS SAVED FROM MONTHLY EXPENSE TABLE

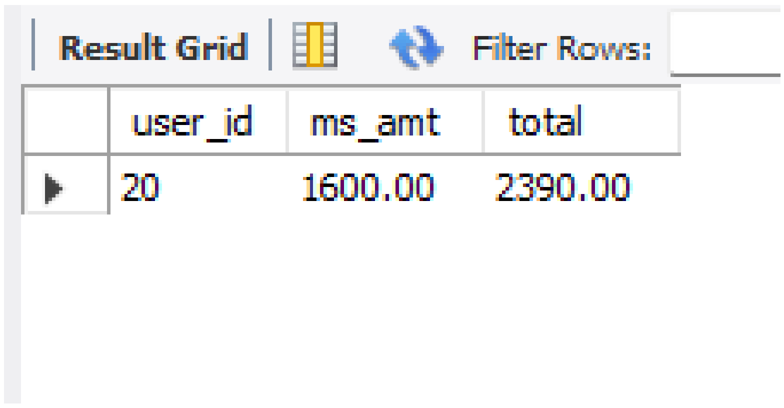


```sql
SELECT * FROM expmg2.bills;CREATE TRIGGER update_monthly_saved
AFTER INSERT ON MonthlyExpense
FOR EACH ROW
BEGIN
    DECLARE saving_amount DECIMAL(10, 2);

    SELECT ms_amt INTO saving_amount
    FROM MonthlySavings
    WHERE user_id = NEW.user_id AND MONTH(month_year) = MONTH(NEW.month_year);

    IF saving_amount IS NOT NULL THEN
        INSERT INTO MonthlySaved (user_id, saving_data, saving_amt, expense_id)
        VALUES (NEW.user_id, NEW.month_year, saving_amount, NEW.expense_id);
    END IF;
END;
```

PRINT WHOSE monthly savings>1500 and totalexpense >2000

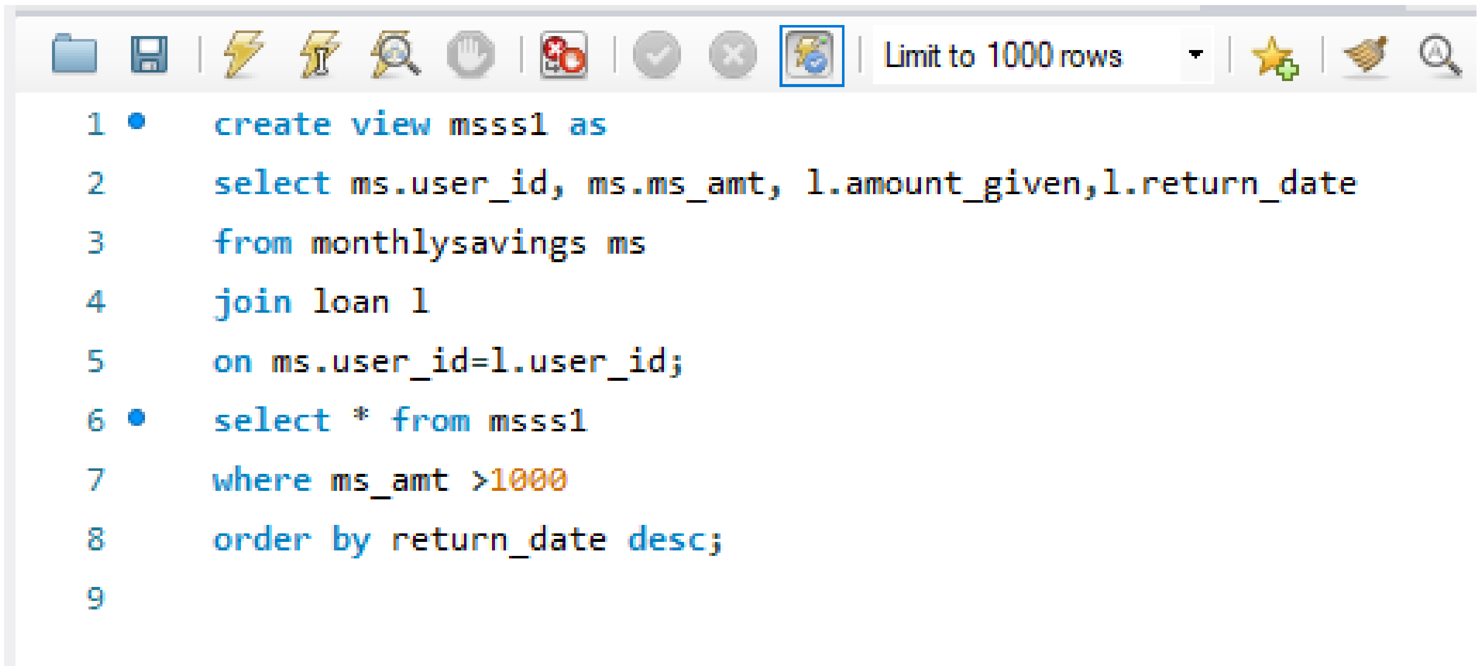


```sql
create view msss2 as
select ms.user_id, ms.ms_amt, b.total
from monthlysavings ms
join bills b
on ms.user_id=b.user_id;
select * from msss2
where ms_amt >1500 and total >2000;
```

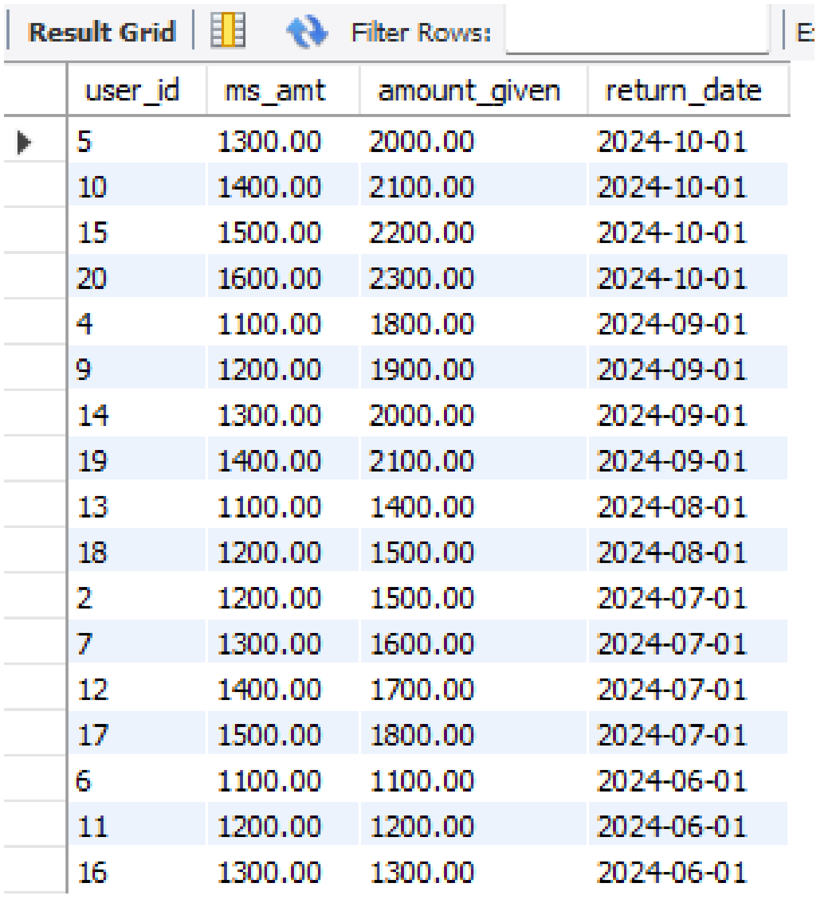


```
user_id   ms_amt    total
20        1600.00   2390.00
```

fetch the user id whose savings is more than 1000 and sort it based on their return date.



```sql
create view msss1 as
select ms.user_id, ms.ms_amt, l.amount_given,l.return_date
from monthlysavings ms
join loan l
on ms.user_id=l.user_id;
select * from msss1
where ms_amt >1000
order by return_date desc;
```



Result Grid
Filter Rows:

user_id	ms_amt	amount_given	return_date
5	1300.00	2000.00	2024-10-01
10	1400.00	2100.00	2024-10-01
15	1500.00	2200.00	2024-10-01
20	1600.00	2300.00	2024-10-01
4	1100.00	1800.00	2024-09-01
9	1200.00	1900.00	2024-09-01
14	1300.00	2000.00	2024-09-01
19	1400.00	2100.00	2024-09-01
13	1100.00	1400.00	2024-08-01
18	1200.00	1500.00	2024-08-01
2	1200.00	1500.00	2024-07-01
7	1300.00	1600.00	2024-07-01
12	1400.00	1700.00	2024-07-01
17	1500.00	1800.00	2024-07-01
6	1100.00	1100.00	2024-06-01
11	1200.00	1200.00	2024-06-01
16	1300.00	1300.00	2024-06-01