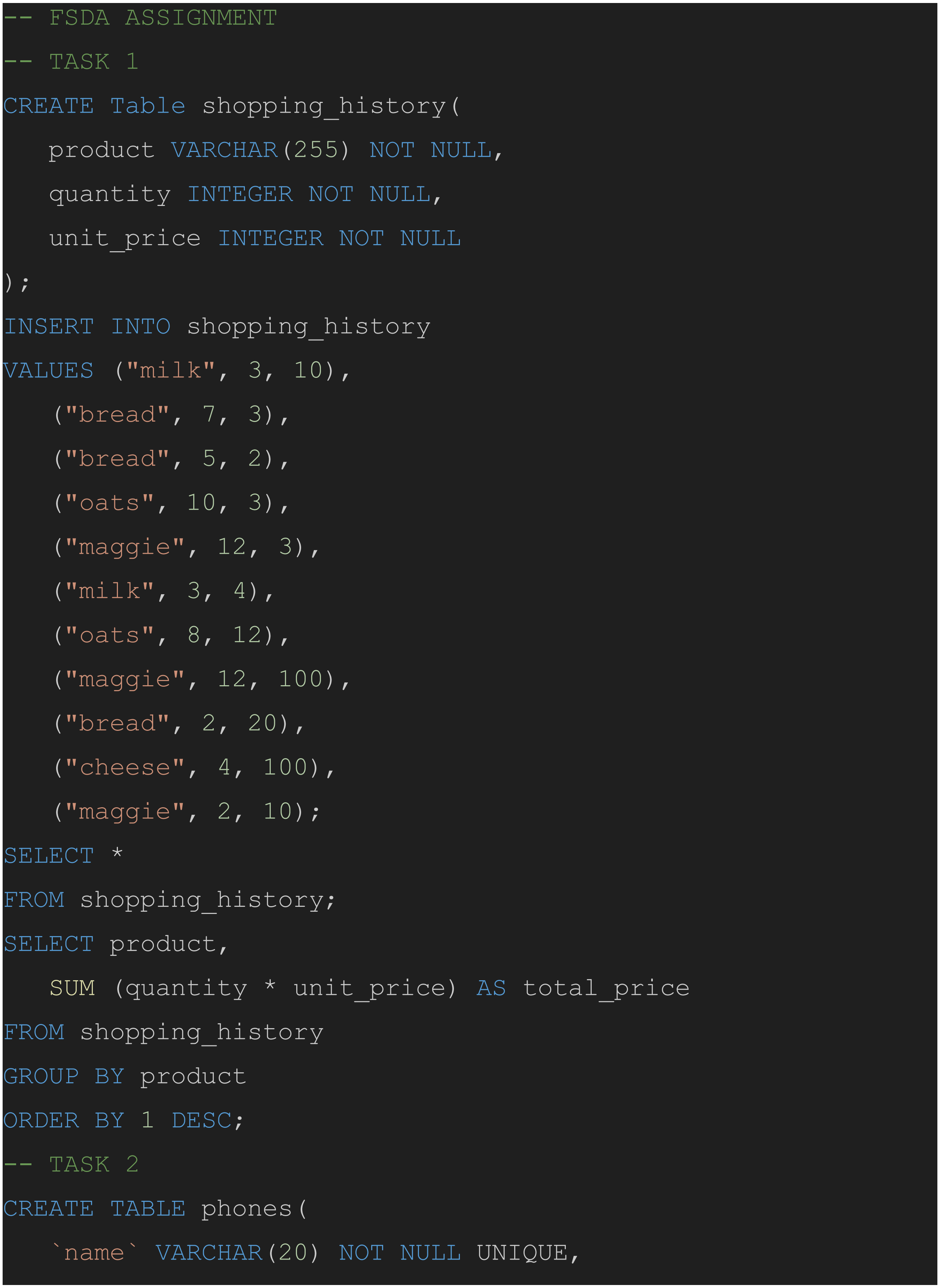
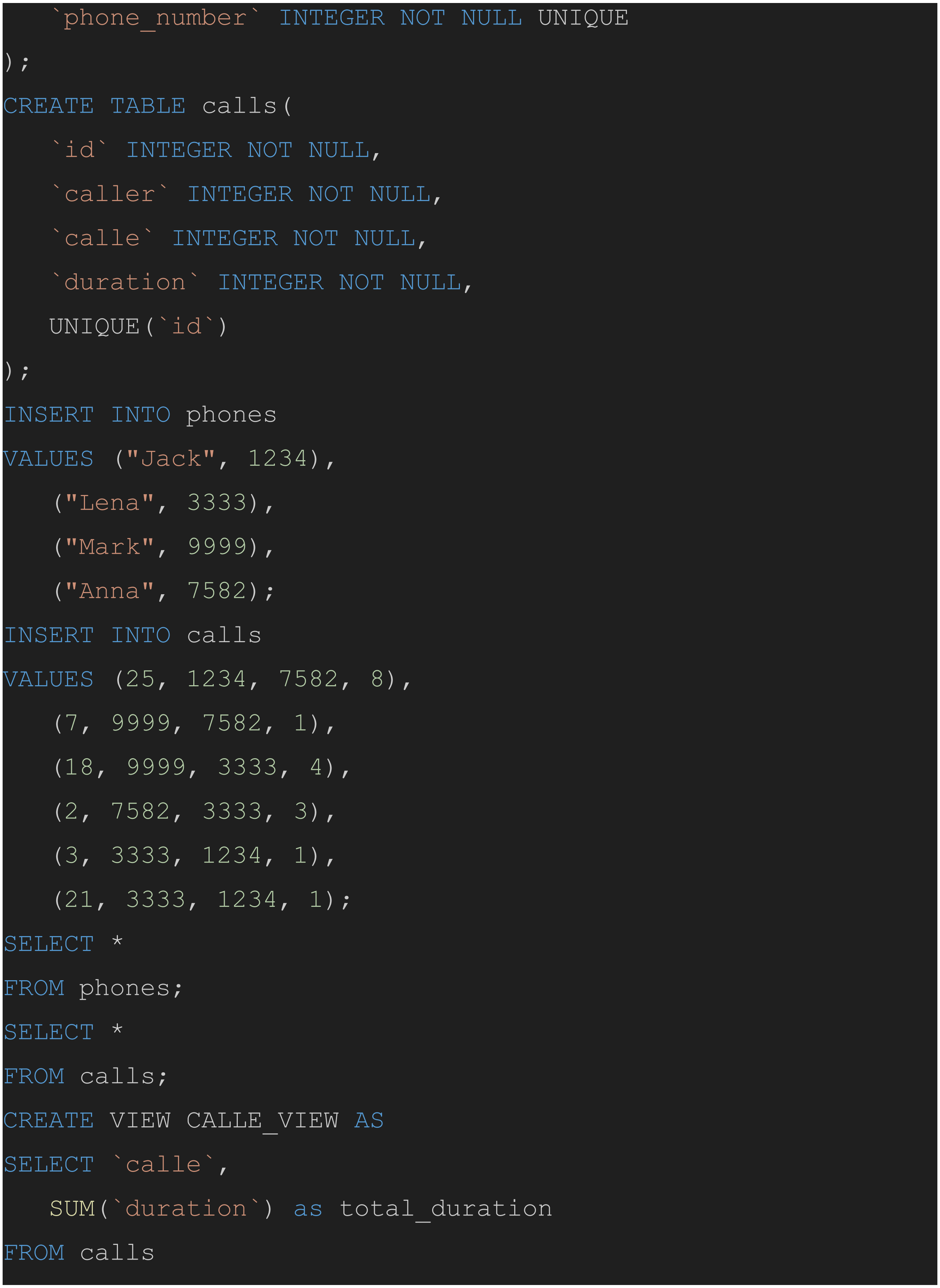
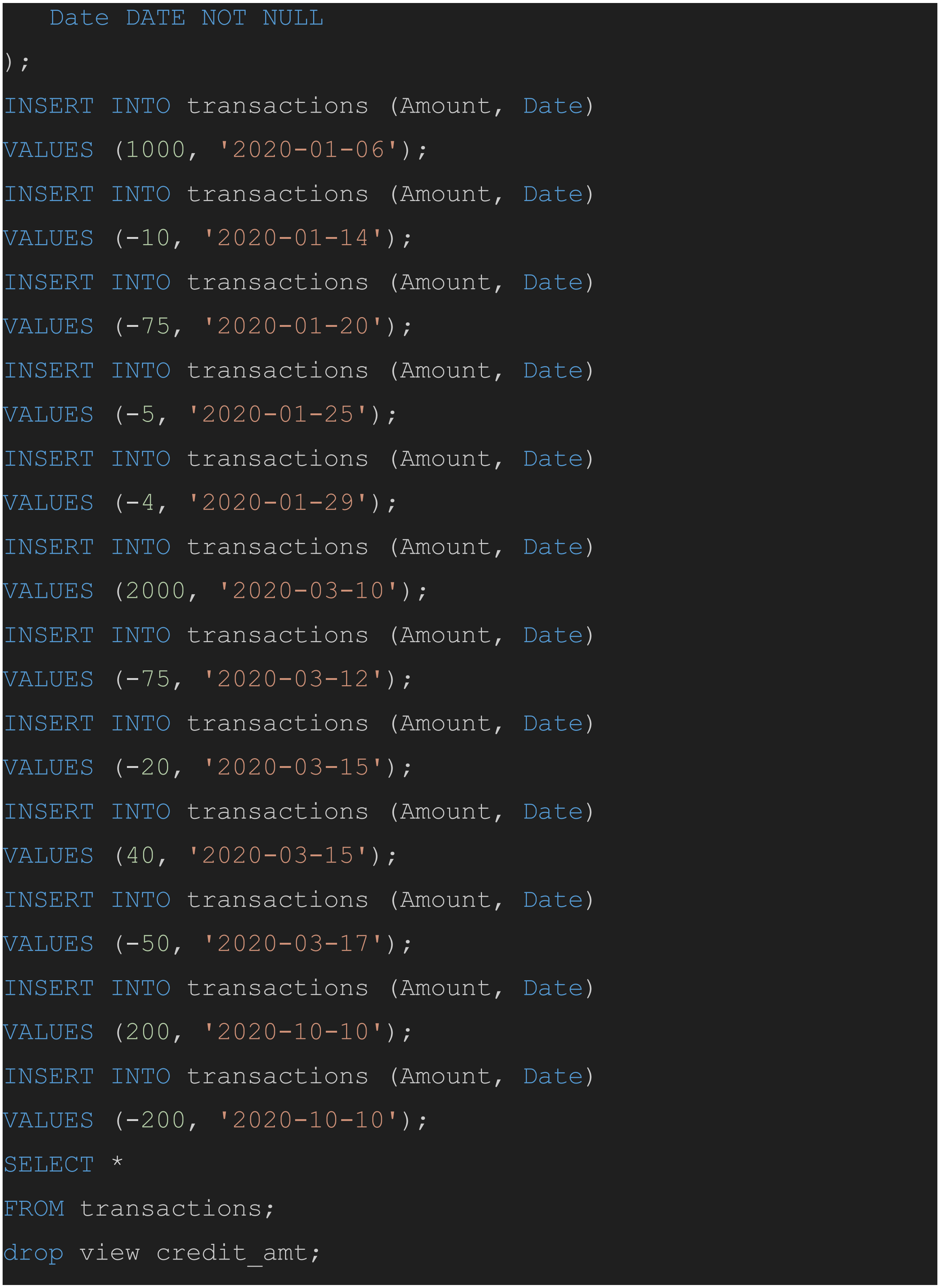


```sql
-- FSDA ASSIGNMENT
-- TASK 1
CREATE Table shopping_history(
   product VARCHAR(255) NOT NULL,
   quantity INTEGER NOT NULL,
   unit_price INTEGER NOT NULL
);
INSERT INTO shopping_history
VALUES ("milk", 3, 10),
   ("bread", 7, 3),
   ("bread", 5, 2),
   ("oats", 10, 3),
   ("maggie", 12, 3),
   ("milk", 3, 4),
   ("oats", 8, 12),
   ("maggie", 12, 100),
   ("bread", 2, 20),
   ("cheese", 4, 100),
   ("maggie", 2, 10);
SELECT *
FROM shopping_history;
SELECT product,
   SUM (quantity * unit_price) AS total_price
FROM shopping_history
GROUP BY product
ORDER BY 1 DESC;
-- TASK 2
CREATE TABLE phones(
   `name` VARCHAR(20) NOT NULL UNIQUE,
```

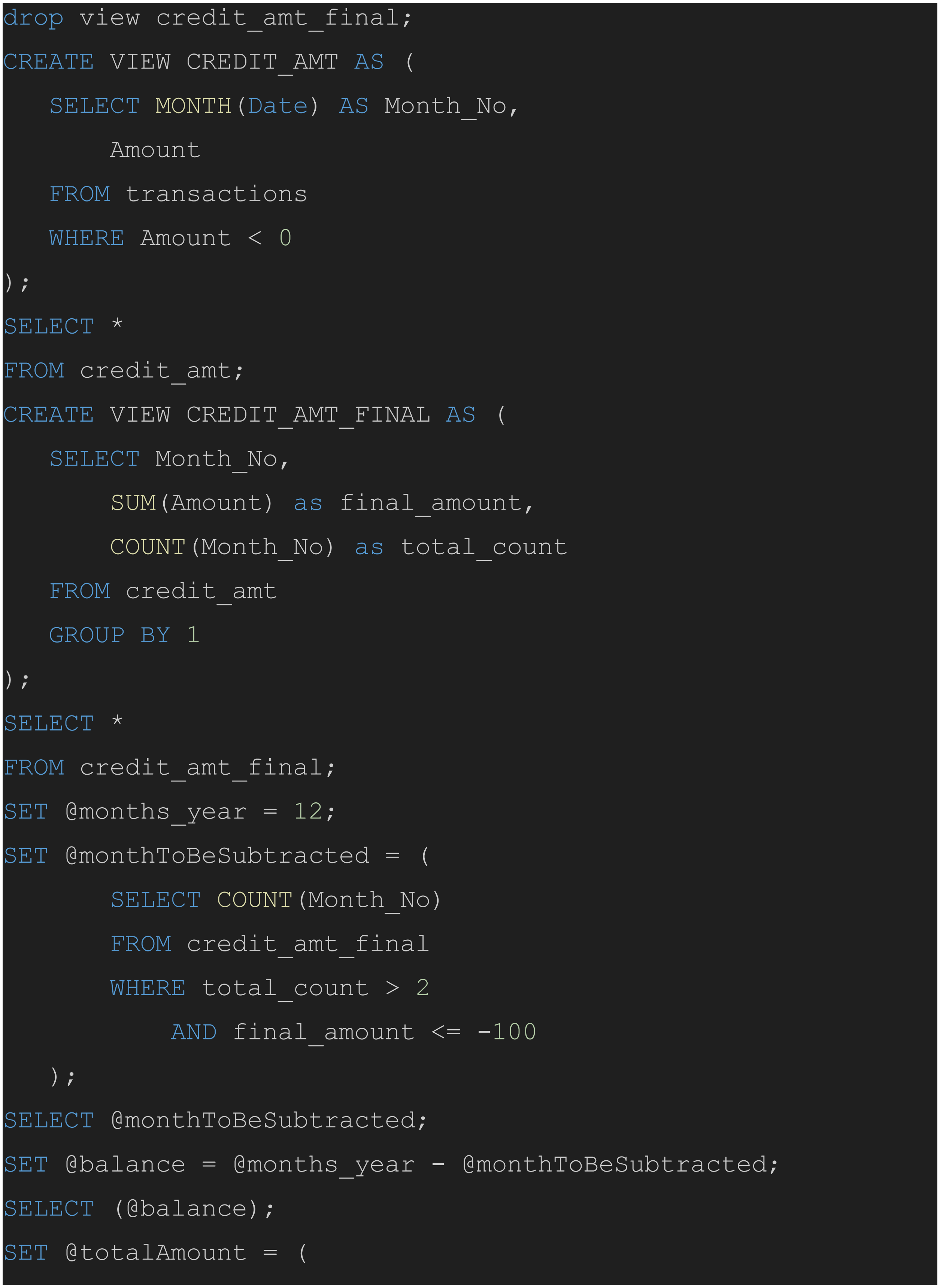


```sql
   `phone_number` INTEGER NOT NULL UNIQUE
);
CREATE TABLE calls(
   `id` INTEGER NOT NULL,
   `caller` INTEGER NOT NULL,
   `calle` INTEGER NOT NULL,
   `duration` INTEGER NOT NULL,
   UNIQUE(`id`)
);
INSERT INTO phones
VALUES ("Jack", 1234),
   ("Lena", 3333),
   ("Mark", 9999),
   ("Anna", 7582);
INSERT INTO calls
VALUES (25, 1234, 7582, 8),
   (7, 9999, 7582, 1),
   (18, 9999, 3333, 4),
   (2, 7582, 3333, 3),
   (3, 3333, 1234, 1),
   (21, 3333, 1234, 1);
SELECT *
FROM phones;
SELECT *
FROM calls;
CREATE VIEW CALLE_VIEW AS
SELECT `calle`,
   SUM(`duration`) as total_duration
FROM calls
```

```
GROUP BY 1
ORDER BY 2 DESC;
CREATE VIEW CALLER_VIEW AS
SELECT `caller`,
   SUM(`duration`) as total_duration
FROM calls
GROUP BY 1
ORDER BY 2 DESC;
CREATE VIEW CALLER_CALLE_VIEW AS
SELECT *
FROM caller_view
UNION
SELECT *
FROM CALLE_VIEW;
SELECT caller,
   SUM(total_duration) as total_time
FROM caller_calle_view
GROUP BY 1
HAVING total_time >= 10;
SELECT p.name
FROM caller_calle_view c
   LEFT JOIN phones p ON c.caller = p.phone_number
GROUP BY p.name
HAVING SUM(c.total_duration) >= 10
ORDER BY SUM(c.total_duration) DESC;
-- TASK 3
-- Active: 1690357858624@@127.0.0.1@3306@sample
CREATE TABLE transactions (
   Amount INTEGER NOT NULL,
```



```
   Date DATE NOT NULL
);
INSERT INTO transactions (Amount, Date)
VALUES (1000, '2020-01-06');
INSERT INTO transactions (Amount, Date)
VALUES (-10, '2020-01-14');
INSERT INTO transactions (Amount, Date)
VALUES (-75, '2020-01-20');
INSERT INTO transactions (Amount, Date)
VALUES (-5, '2020-01-25');
INSERT INTO transactions (Amount, Date)
VALUES (-4, '2020-01-29');
INSERT INTO transactions (Amount, Date)
VALUES (2000, '2020-03-10');
INSERT INTO transactions (Amount, Date)
VALUES (-75, '2020-03-12');
INSERT INTO transactions (Amount, Date)
VALUES (-20, '2020-03-15');
INSERT INTO transactions (Amount, Date)
VALUES (40, '2020-03-15');
INSERT INTO transactions (Amount, Date)
VALUES (-50, '2020-03-17');
INSERT INTO transactions (Amount, Date)
VALUES (200, '2020-10-10');
INSERT INTO transactions (Amount, Date)
VALUES (-200, '2020-10-10');
SELECT *
FROM transactions;
drop view credit_amt;
```



```
drop view credit_amt_final;
CREATE VIEW CREDIT_AMT AS (
   SELECT MONTH(Date) AS Month_No,
       Amount
   FROM transactions
   WHERE Amount < 0
);
SELECT *
FROM credit_amt;
CREATE VIEW CREDIT_AMT_FINAL AS (
   SELECT Month_No,
       SUM(Amount) as final_amount,
       COUNT(Month_No) as total_count
   FROM credit_amt
   GROUP BY 1
);
SELECT *
FROM credit_amt_final;
SET @months_year = 12;
SET @monthToBeSubtracted = (
       SELECT COUNT(Month_No)
       FROM credit_amt_final
       WHERE total_count > 2
           AND final_amount <= -100
   );
SELECT @monthToBeSubtracted;
SET @balance = @months_year - @monthToBeSubtracted;
SELECT (@balance);
SET @totalAmount = (
       SELECT SUM(Amount)
       FROM transactions
   );
SELECT @totalAmount;
SET @balances = @totalAmount - (@balance * 5);
SELECT @balances AS balance;
```