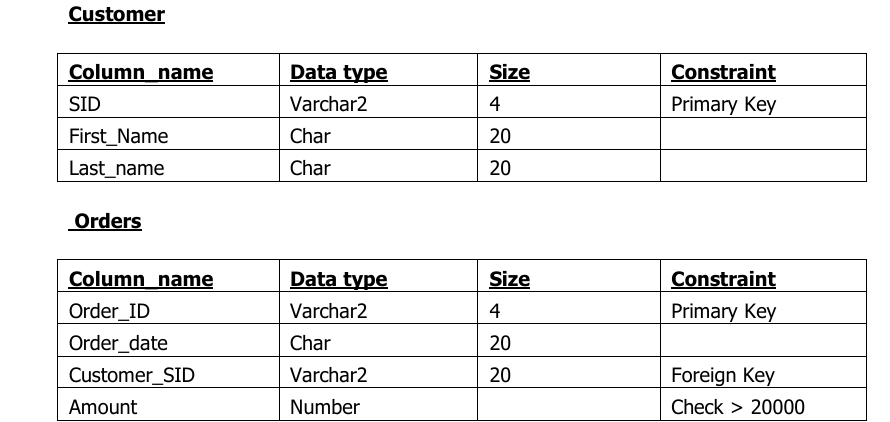
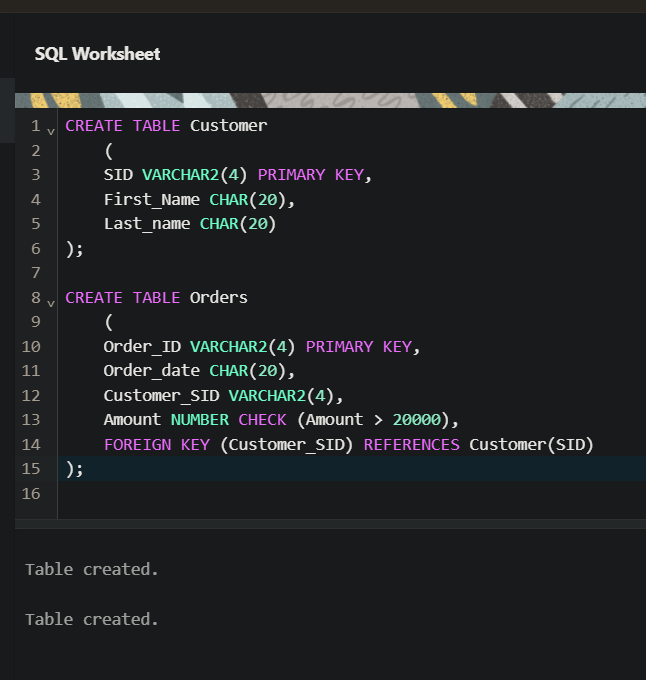
|  |  |
| --- | --- |
|  |  |

**EXERCISE 1**

**AIM**: Create the following table.



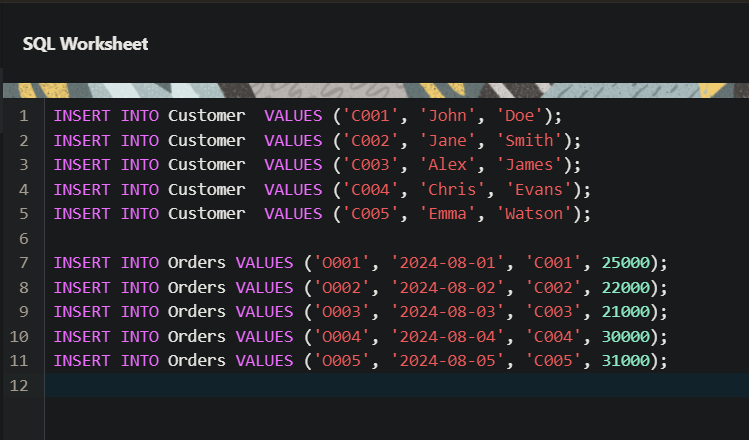
**Output:**

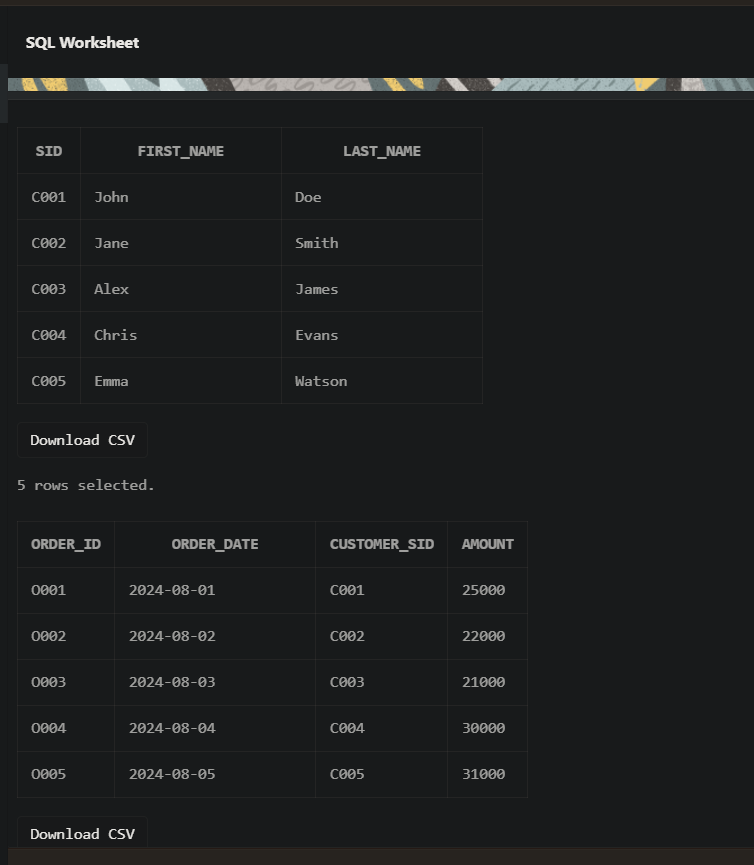


**EXERCISE 2**

**AIM:** Insert 5 records for each table.

**Output:**

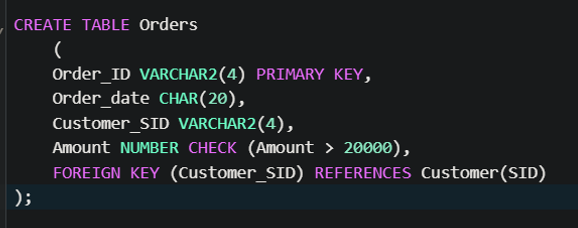




**EXERCISE 3**

**AIM:** Customer SID column in the ORDERS table is a foreign key pointing to the SID column in the CUSTOMER table.

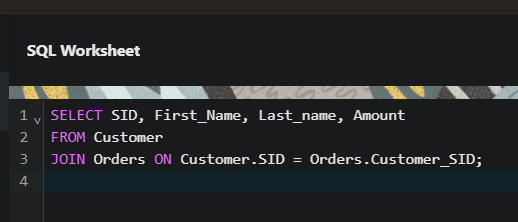
**Output:**

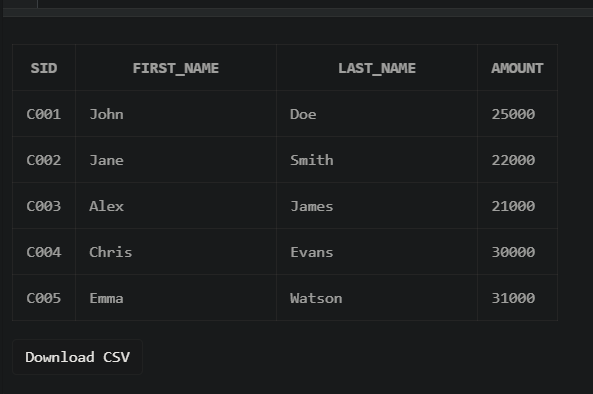


**EXERCISE 4**

**AIM:** List the details of the customers along with the amount.

**Output:**

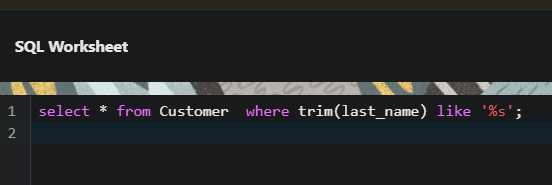
****

****

**EXERCISE 5**

**AIM:** List the customers whose names end with “s”.

**Output:**

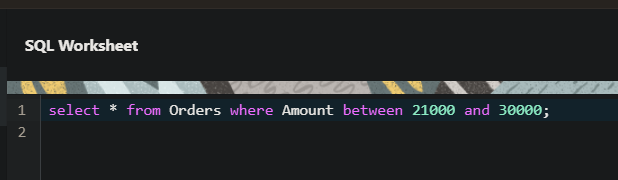


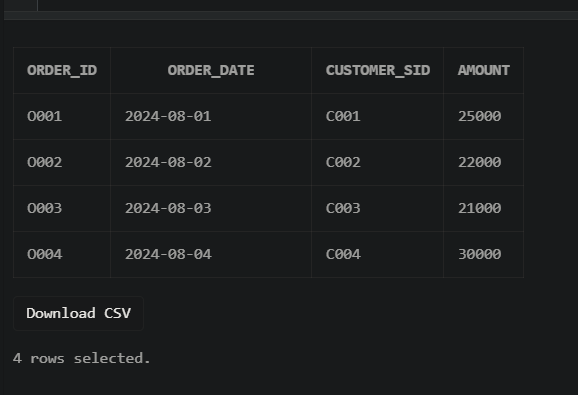


**EXERCISE 6**

**AIM:** List the orders where amount is between 21000 and 30000

**Output:**

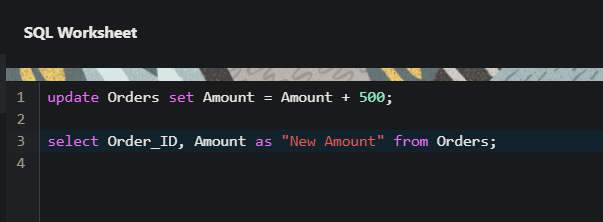


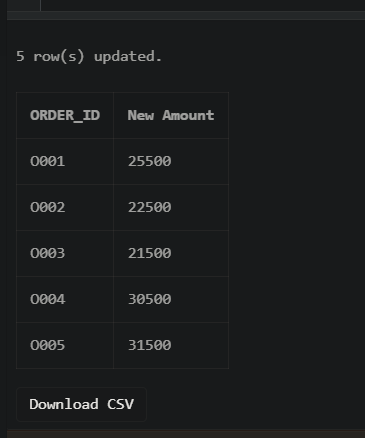


**EXERCISE 7**

**AIM:** List the orders where amount is increased by 500 and replace with name “new amount”**.**

**Output:**

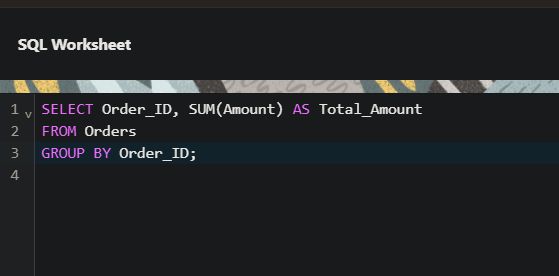


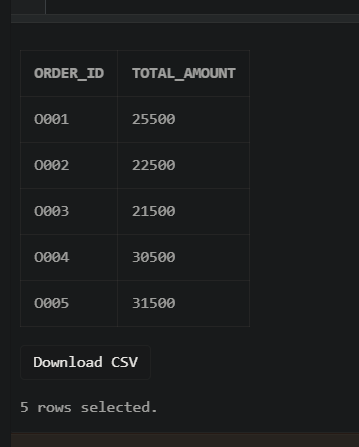


**EXERCISE 8**

**AIM:** Display the order\_id and total amount of orders.

**Output:**

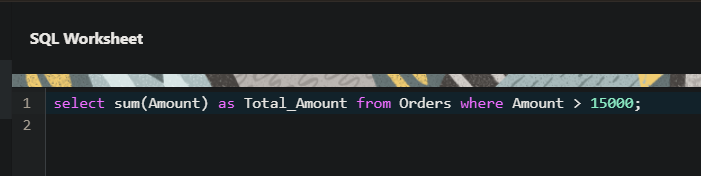


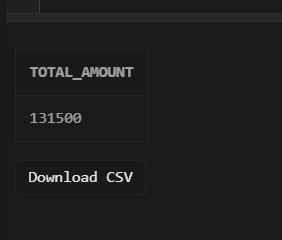


**EXERCISE 9**

**AIM:** Calculate the total amount of orders that has more than 15000.

**Output:**

****



**EXERCISE 10**

**AIM:** Display all the string functions used in SQL.

**Output:**

SELECT

LOWER('ORACLE') AS "Lowercase", -- Converts string to lowercase

UPPER('oracle') AS "Uppercase", -- Converts string to uppercase

SUBSTR('ORACLE', 2, 3) AS "Substring", -- Extracts substring

LENGTH('ORACLE') AS "Length”, -- Returns length of string

INSTR('ORACLE', 'A') AS "Position", -- Returns position of a character

LPAD('123', 5, '0') AS "Left Padding", -- Pads a string on the left

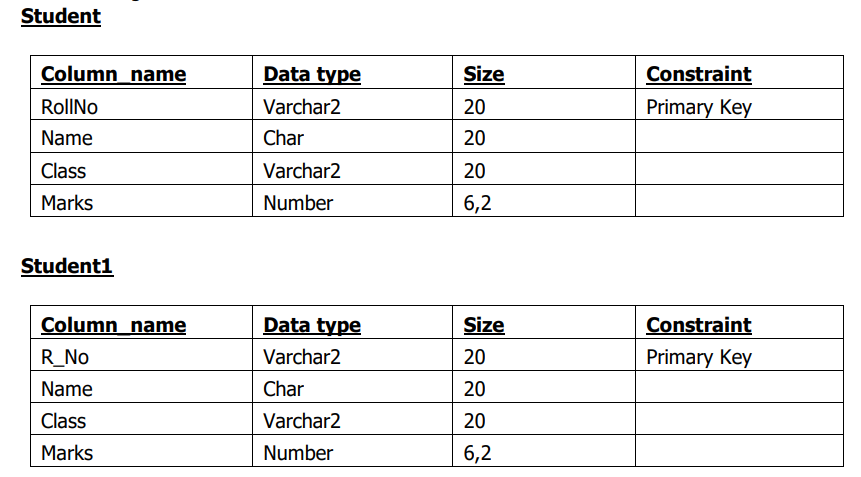
RPAD('123', 5, '0') AS "Right Padding",-- Pads a string on the right

TRIM('O' FROM 'ORACLE') AS "Trimmed" -- Trims a specified character

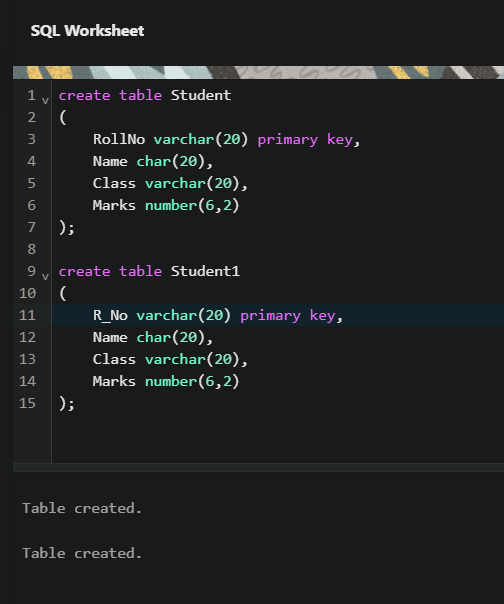
FROM DUAL;

**EXERCISE 11**

**AIM:** Create the following tables.

****

**Output:**

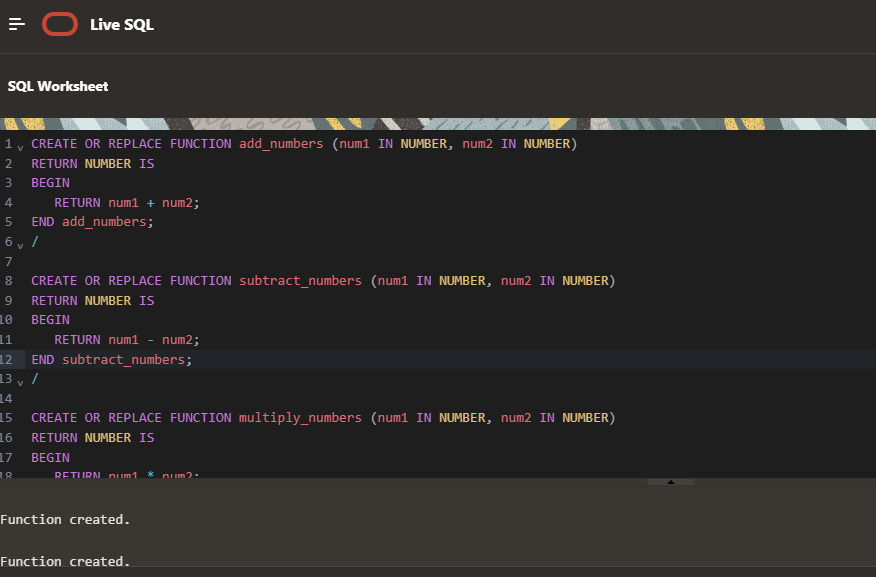
****

Consider a PL/SQL code that accepts 2 numbers & return addition, subtraction, multiplication & division of two numbers using stored functions and local function.

Solution

1. Create stored functions to perform the arithmetic operations.

2. Use local functions within the main procedure or function to invoke the stored functions and perform calculations.



CREATE OR REPLACE FUNCTION add\_numbers (num1 IN NUMBER, num2 IN NUMBER)

RETURN NUMBER IS

BEGIN

RETURN num1 + num2;

END add\_numbers;

/

CREATE OR REPLACE FUNCTION subtract\_numbers (num1 IN NUMBER, num2 IN NUMBER)

RETURN NUMBER IS

BEGIN

RETURN num1 - num2;

END subtract\_numbers;

/

CREATE OR REPLACE FUNCTION multiply\_numbers (num1 IN NUMBER, num2 IN NUMBER)

RETURN NUMBER IS

BEGIN

RETURN num1 \* num2;

END multiply\_numbers;

/

CREATE OR REPLACE FUNCTION divide\_numbers (num1 IN NUMBER, num2 IN NUMBER)

RETURN NUMBER IS

BEGIN

IF num2 = 0 THEN

RETURN NULL; -- To avoid division by zero

ELSE

RETURN num1 / num2;

END IF;

END divide\_numbers;

/

