Assignment for Week 1

Linux

1) Find all files in a directory larger than 10 kb.

2) Find all commands from history which contains ‘sudo’.

3) Set global alias for “ls –al” as lsal.

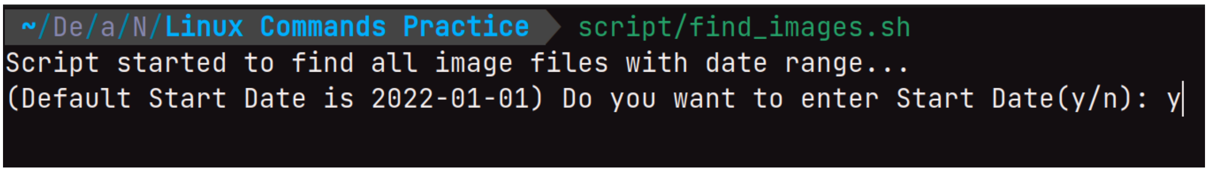
4) Count the number of lines in a text file.

5) Write a shell script to find all image files (.jpg, .png, .gif) in a directory by start date and end date.

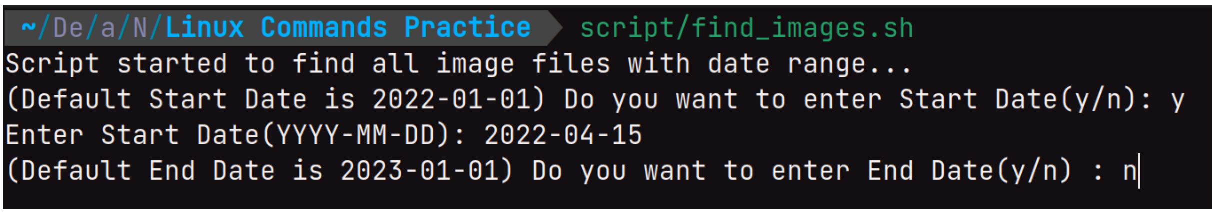
Create a sample directory which contains various image files.

Set the default value of start date = “2023-01-01" and end date = “2023-01-01" in the script

Give the user the choice(y/n) of entering the start and end dates. And then output the file paths and also the number of files found.

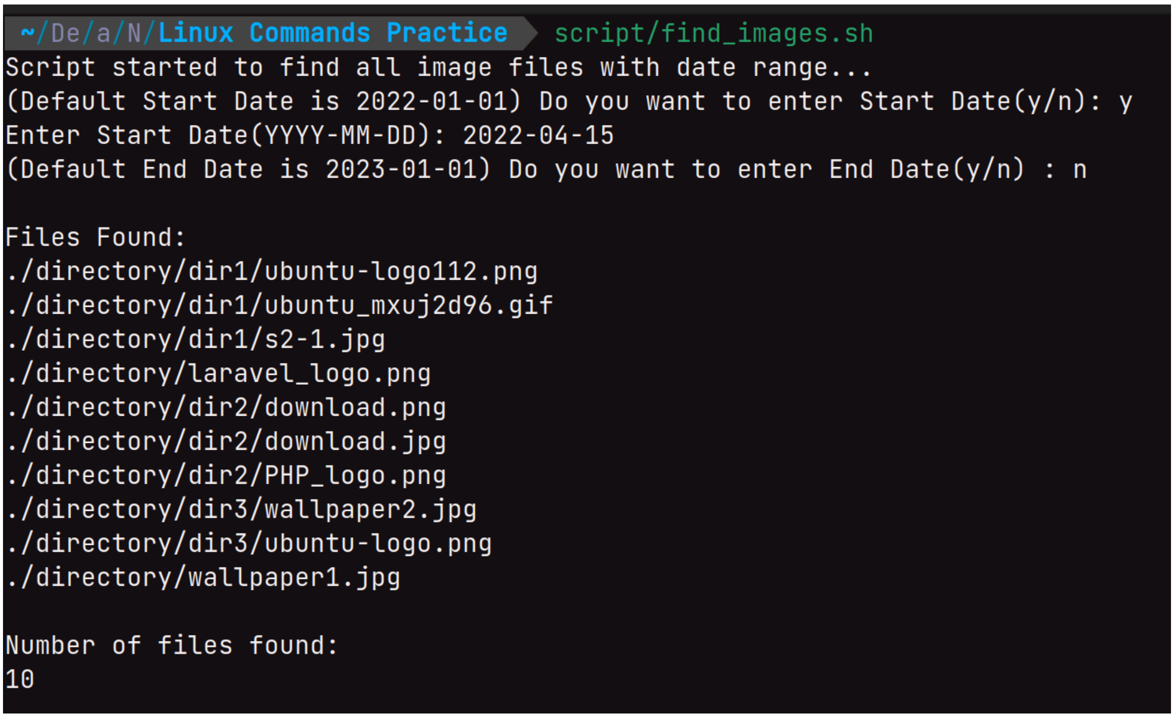


```
 ~/De/a/N/Linux Commands Practice  script/find_images.sh
Script started to find all image files with date range...
(Default Start Date is 2022-01-01) Do you want to enter Start Date(y/n): y
```



```
 ~/De/a/N/Linux Commands Practice  script/find_images.sh
Script started to find all image files with date range...
(Default Start Date is 2022-01-01) Do you want to enter Start Date(y/n): y
Enter Start Date(YYYY-MM-DD): 2022-04-15
(Default End Date is 2023-01-01) Do you want to enter End Date(y/n) : n
```

Output:



```
~/De/a/N/Linux Commands Practice  script/find_images.sh
Script started to find all image files with date range...
(Default Start Date is 2022-01-01) Do you want to enter Start Date(y/n): y
Enter Start Date(YYYY-MM-DD): 2022-04-15
(Default End Date is 2023-01-01) Do you want to enter End Date(y/n) : n

Files Found:
./directory/dir1/ubuntu-logo112.png
./directory/dir1/ubuntu_mxuj2d96.gif
./directory/dir1/s2-1.jpg
./directory/laravel_logo.png
./directory/dir2/download.png
./directory/dir2/download.jpg
./directory/dir2/PHP_logo.png
./directory/dir3/wallpaper2.jpg
./directory/dir3/ubuntu-logo.png
./directory/wallpaper1.jpg

Number of files found:
10
```

SQL

1) Write an SQL query to find the second highest salary from the Employee table. If there is no second highest salary, the query should output null.
   a. Create the sample table which the attributes as “id”, “salary”.
   b. Employee table:

```
+----+--------+
| id | salary |
+----+--------+
| 1  | 400    |
| 2  | 200    |
| 3  | 300    |
+----+--------+
Output:
+---------------------+
| result              |
+---------------------+
| 300                 |
+---------------------+
```

Case 1:

create table Employee (id INT, salary INT); insert into Employee (id, salary) values (1, 2500); insert into Employee (id, salary) values (2, 2936); insert into Employee (id, salary) values (3, 5918); insert into Employee (id, salary) values (4, 2500); insert into Employee (id, salary) values (5, 5500); insert into Employee (id, salary) values (6, 3211); insert into Employee (id, salary) values (7, 6283); insert into Employee (id, salary) values (8, 2500); insert into Employee (id, salary) values (9, 5500); insert into Employee (id, salary) values (10, 5532);

Case 2:

create table Employee (id INT, salary INT);

insert into Employee (id, salary) values (1, 5500);

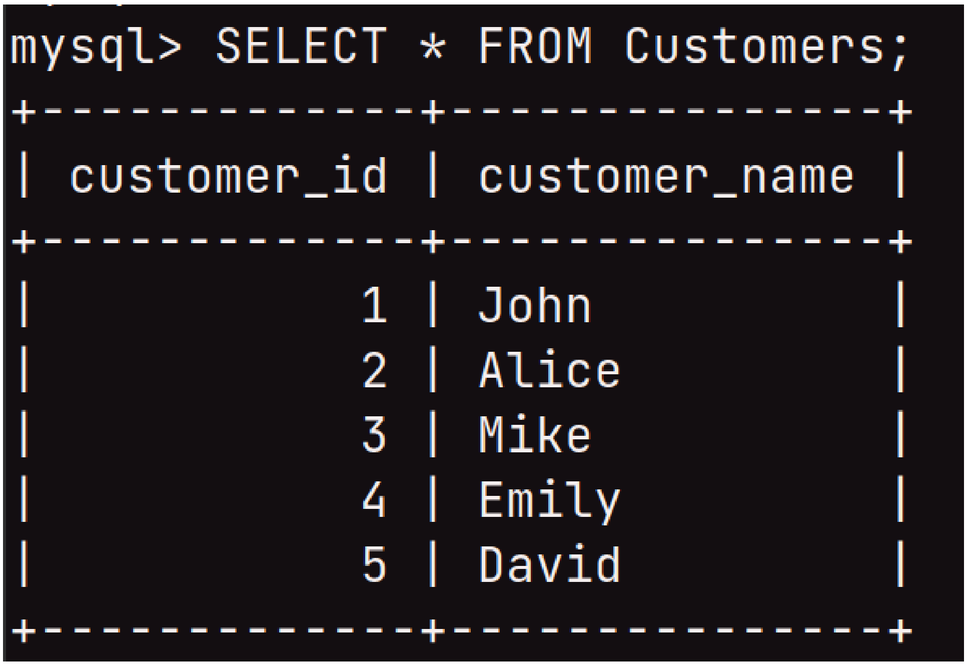
insert into Employee (id, salary) values (2, 5500);

insert into Employee (id, salary) values (3, 5500);

2) Find out the total number of orders for each customer, including customers who have not placed any orders.

CREATE TABLE Customers (customer_id INT, customer_name VARCHAR(50));

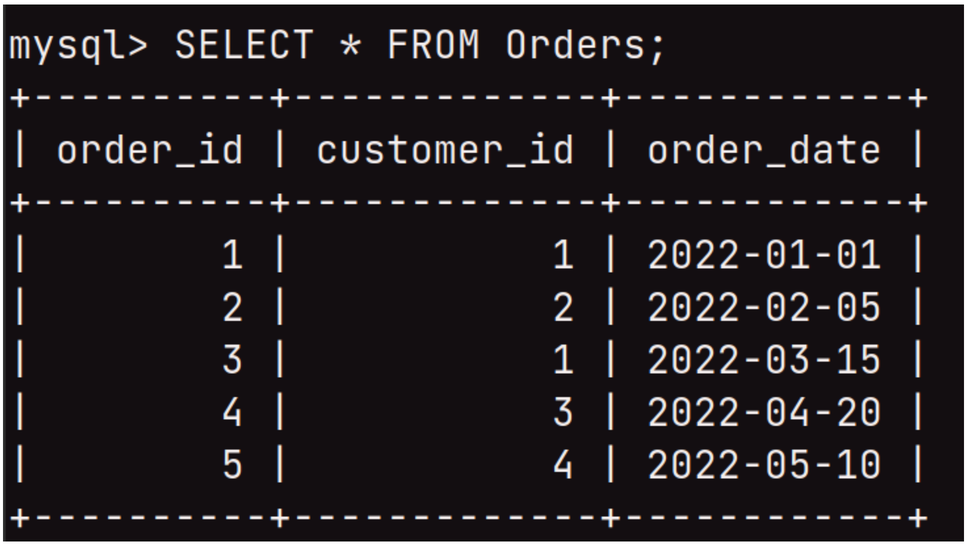
INSERT INTO Customers (customer_id, customer_name) VALUES (1, 'John'), (2, 'Alice'), (3, 'Mike'), (4, 'Emily'), (5, 'David');



```
mysql> SELECT * FROM Customers;
+-------------+---------------+
| customer_id | customer_name |
+-------------+---------------+
|           1 | John          |
|           2 | Alice         |
|           3 | Mike          |
|           4 | Emily         |
|           5 | David         |
+-------------+---------------+
```

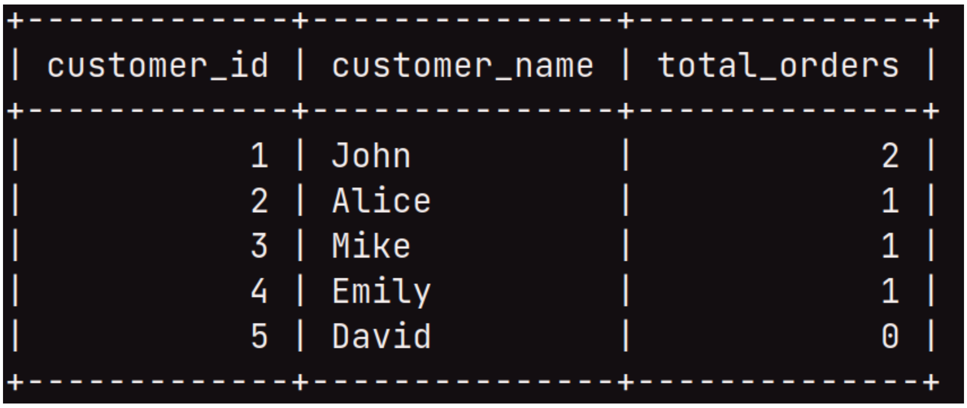
CREATE TABLE Orders (order_id INT, customer_id INT, order_date DATE);

INSERT INTO Orders (order_id, customer_id, order_date) VALUES (1, 1, '2022-01-01'), (2, 2, '2022-02-05'), (3, 1, '2022-03-15'), (4, 3, '2022-04-20'), (5, 4, '2022-05-10');



```
mysql> SELECT * FROM Orders;
+----------+-------------+------------+
| order_id | customer_id | order_date |
+----------+-------------+------------+
|        1 |           1 | 2022-01-01 |
|        2 |           2 | 2022-02-05 |
|        3 |           1 | 2022-03-15 |
|        4 |           3 | 2022-04-20 |
|        5 |           4 | 2022-05-10 |
+----------+-------------+------------+
```

Output:



```
+-------------+---------------+--------------+
| customer_id | customer_name | total_orders |
+-------------+---------------+--------------+
|           1 | John          |            2 |
|           2 | Alice         |            1 |
|           3 | Mike          |            1 |
|           4 | Emily         |            1 |
|           5 | David         |            0 |
+-------------+---------------+--------------+
```