***LAB CYCLE II***

***Q.SET 1 date :8-6-2021***

**Create the following tables and execute the queries given below**

**SAILORS**

|  |  |  |  |
| --- | --- | --- | --- |
| **sid** | **sname** | **rating** | **age** |
| 22 | Dustin | 7 | 45 |
| 29 | Brutas | 1 | 33 |
| 31 | Lubber | 8 | 55 |
| 32 | Andy | 8 | 25 |
| 58 | Rusty | 10 | 35 |
| 64 | Horatio | 7 | 35 |
| 71 | Zorba | 10 | 16 |
| 74 | Horatio | 9 | 35 |
| 85 | Art | 3 | 26 |
| 95 | Bob | 3 | 64 |

**BOATS**

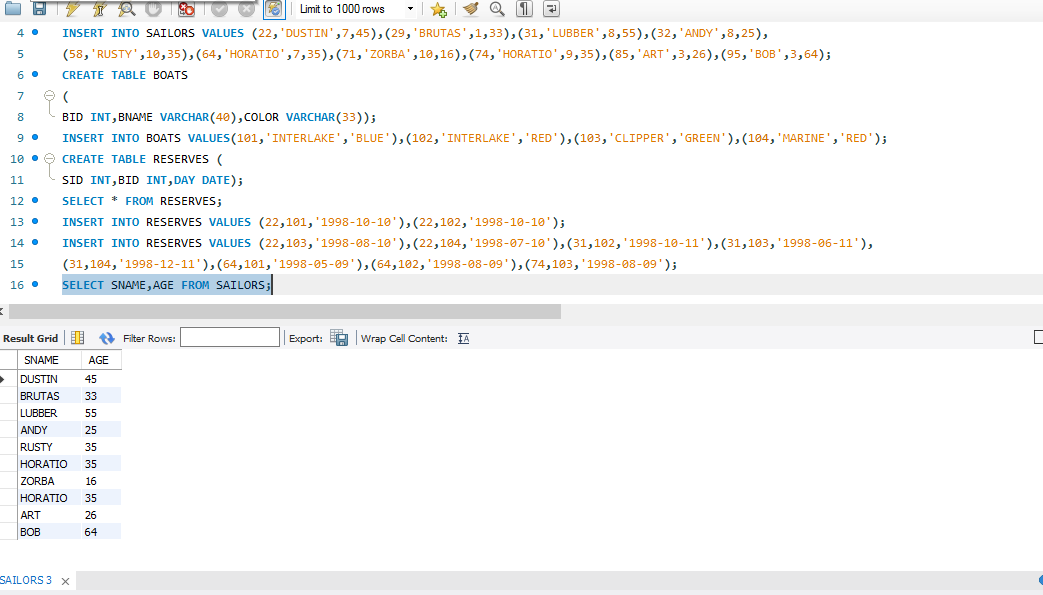
|  |  |  |
| --- | --- | --- |
| **Bid** | **bname** | **color** |
| 101 | Interlake | Blue |
| 102 | Interlake | Red |
| 103 | Clipper | Green |
| 104 | Marine | Red |

**RESERVES**

|  |  |  |
| --- | --- | --- |
| **sid** | **bid** | **day** |
| 22 | 101 | 10/10/98 |
| 22 | 102 | 10/10/98 |
| 22 | 103 | 10/8/98 |
| 22 | 104 | 10/7/98 |
| 31 | 102 | 11/10/98 |
| 31 | 103 | 11/6/98 |
| 31 | 104 | 11/12/98 |
| 64 | 101 | 9/5/98 |
| 64 | 102 | 9/8/98 |
| 74 | 103 | 9/8/98 |

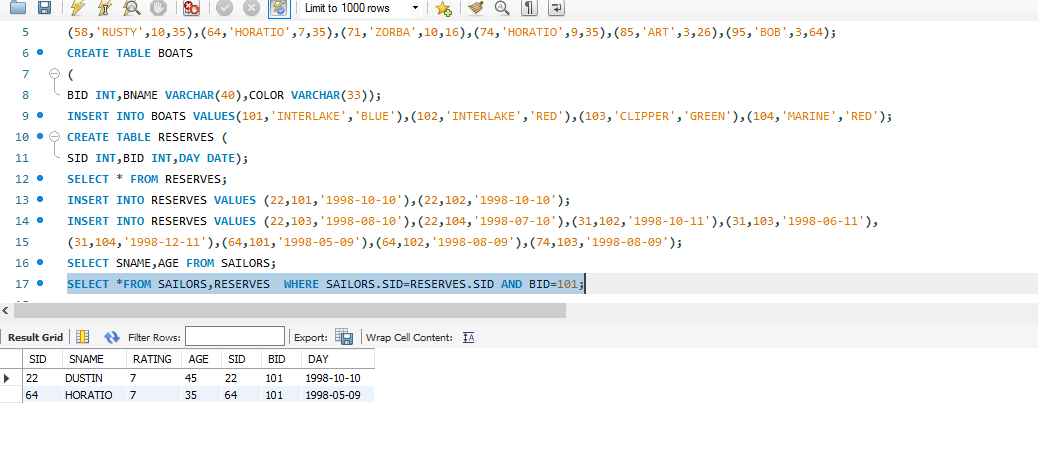
1. Find the names and ages of all sailors

**OUTPUT**



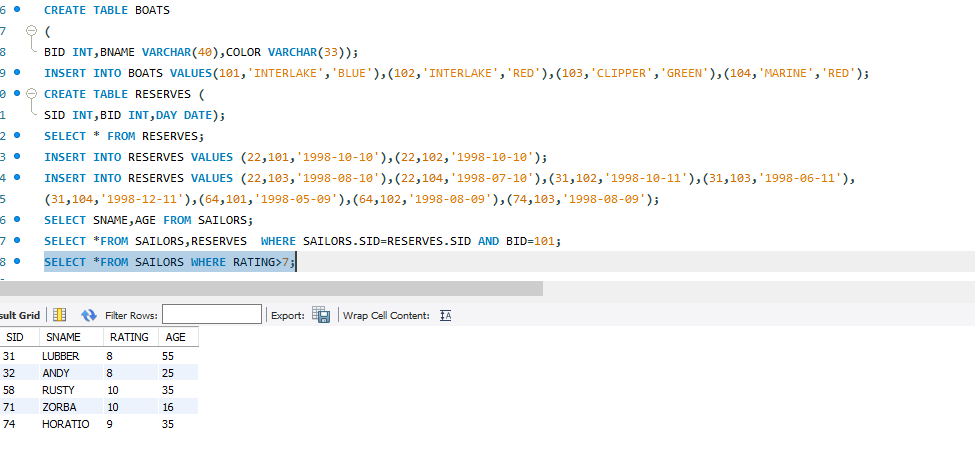
1. Find all information of sailors who have reserved boat number 101.

**OUTPUT**



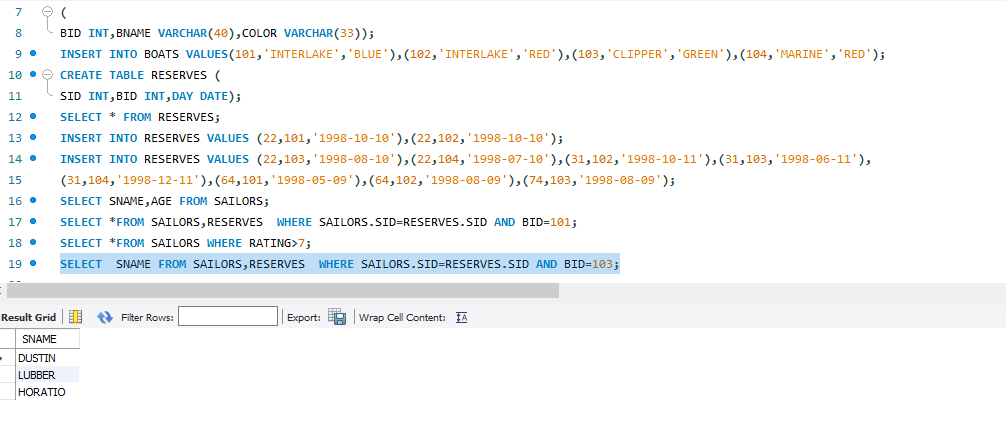
1. Find all sailors with rating above 7

**OUTPUT**



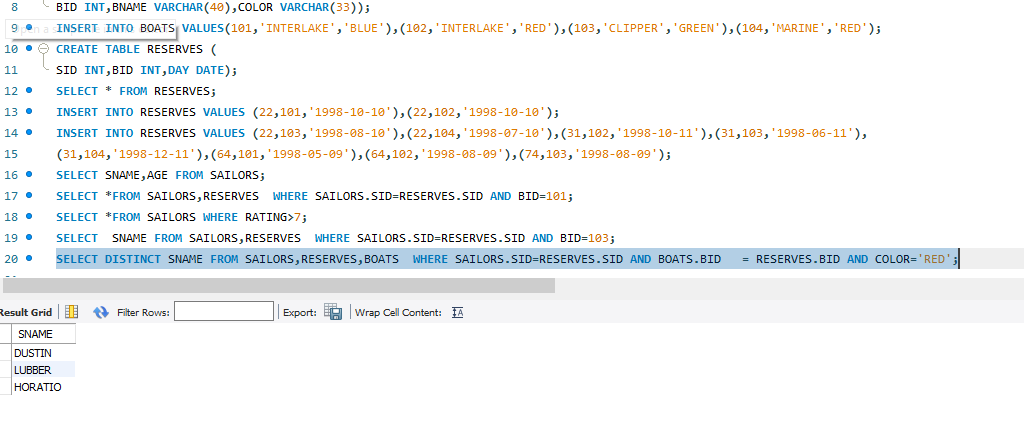
1. Find the names of sailors who have reserved boat no 103

**OUTPUT**



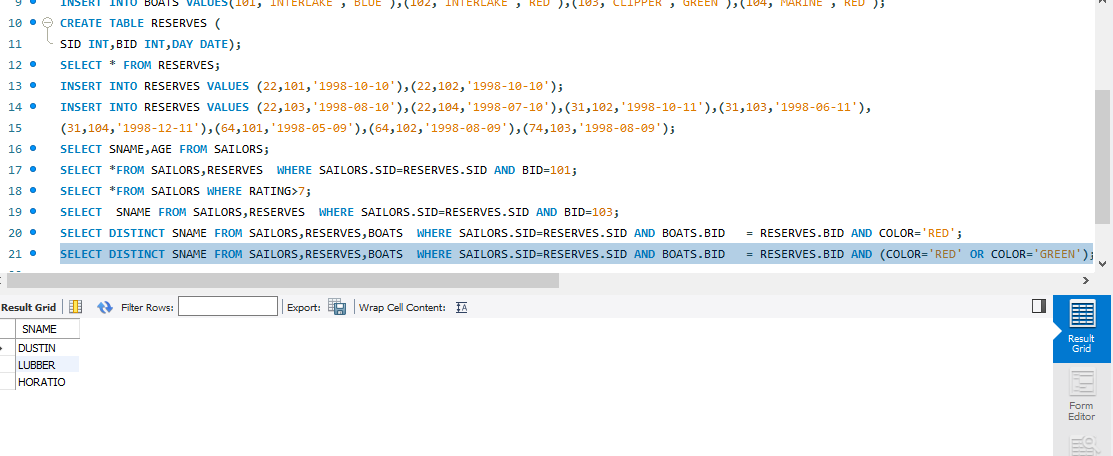
1. Find the names of sailors who have reserved a red boat, and list in the order of age.

**OUTPUT**



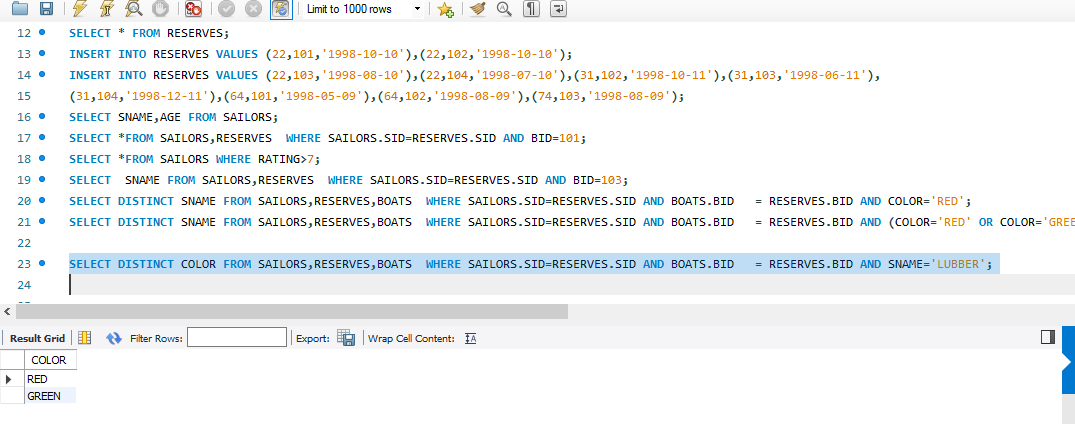
1. Find the names of sailors who have reserved either a red or green boat

**OUTPUT**



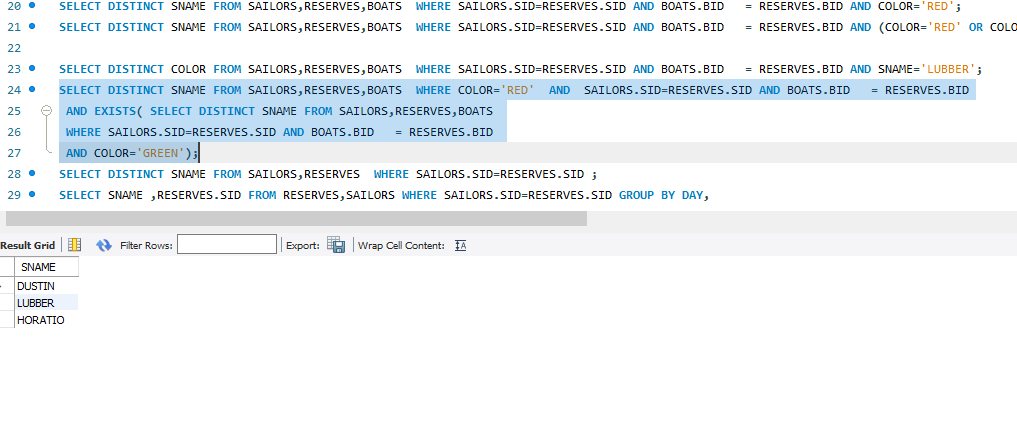
1. Find the colors of boats reserved by “Lubber”.

**OUTPUT**



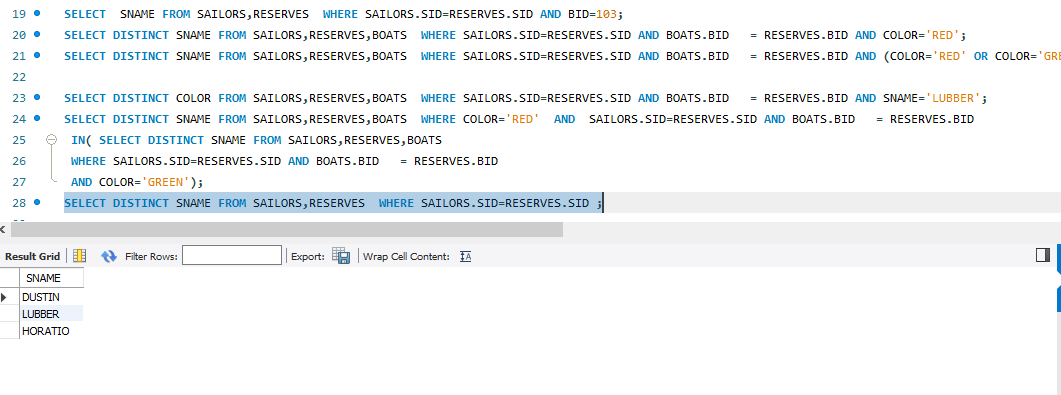
1. Find the names of sailors who have reserved both red and green boats

**OUTPUT**



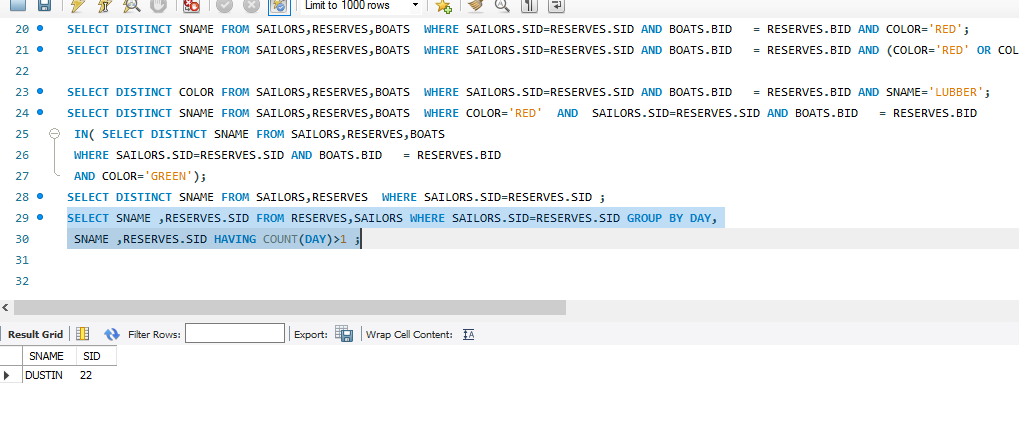
1. Find the names of sailors who have reserved at least one boat

**OUTPUT**



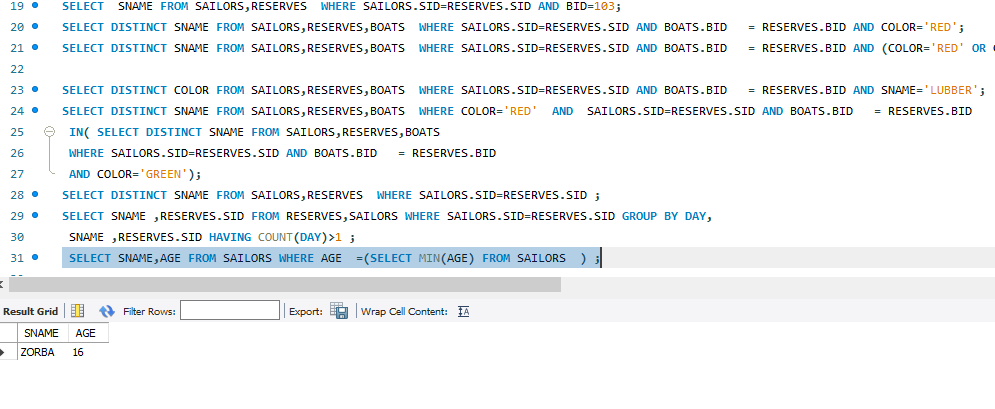
1. Find the ids and names of sailors who have reserved two different boats on the same day.

**OUTPUT**



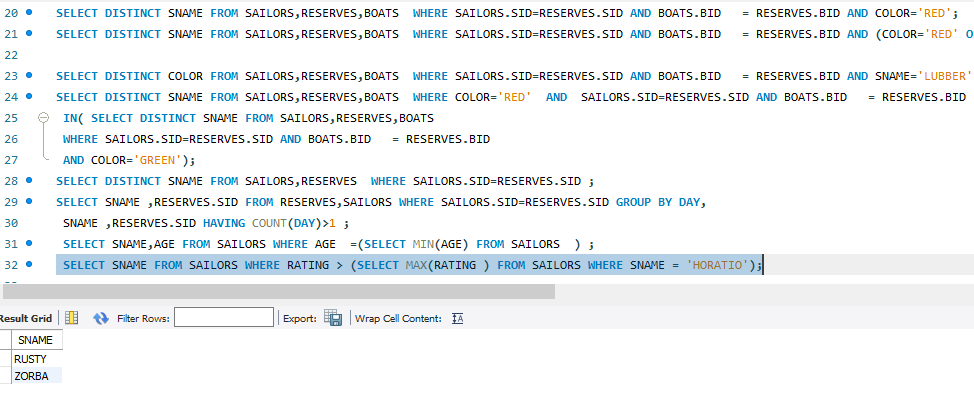
1. Find the name and the age of the youngest sailor.

**OUTPUT**



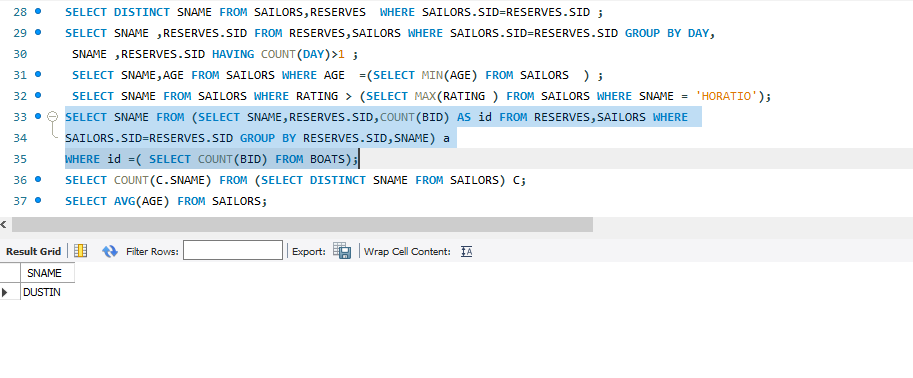
1. Find the names and ratings of sailor whose rating is better than some sailor called Horatio.

**OUTPUT**



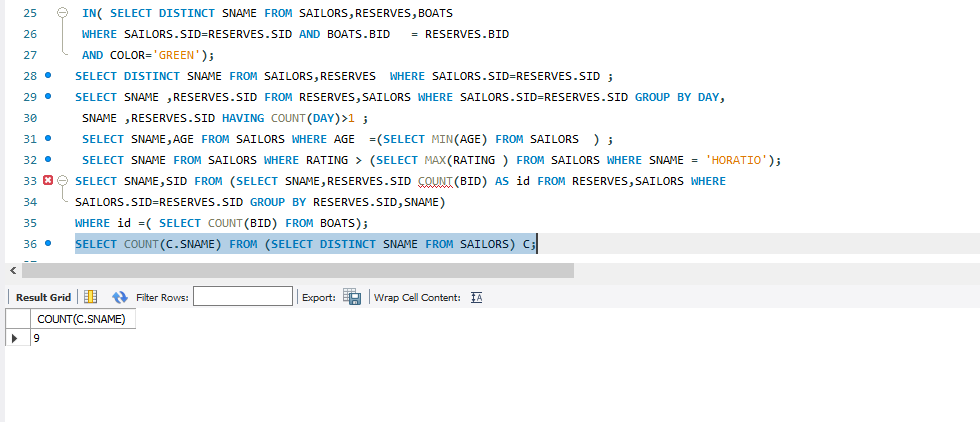
1. Find the names of sailors who have reserved all boats.

**OUTPUT**



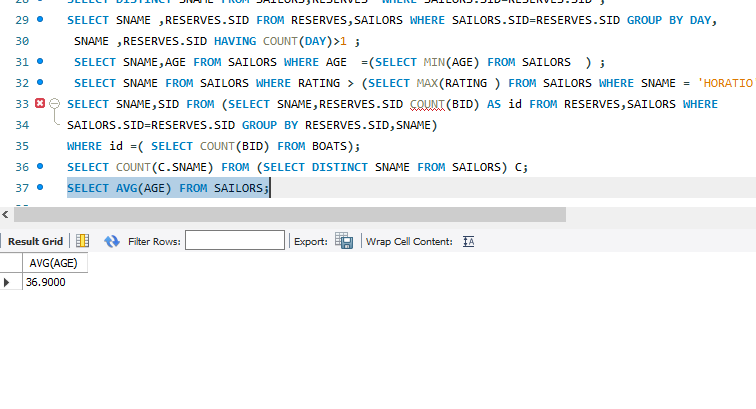
1. Count the number of different sailor names.

**OUTPUT**



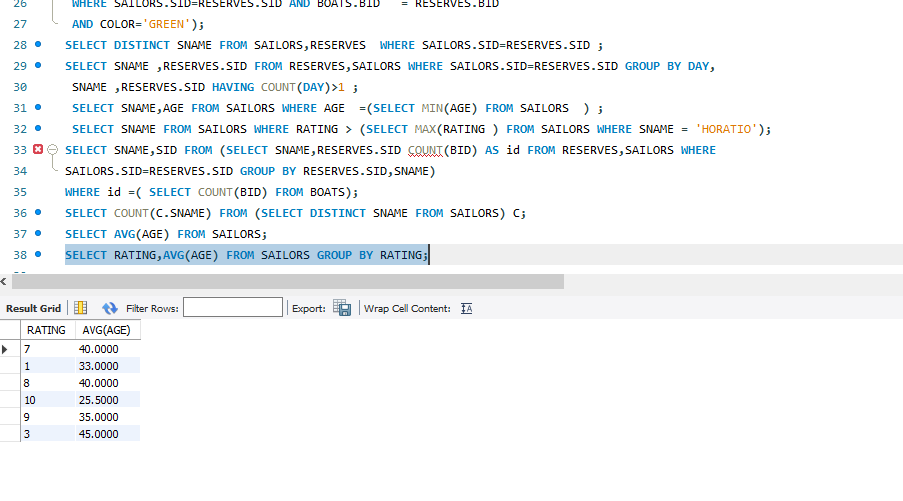
1. Calculate the average age of all sailors.

**OUTPUT**



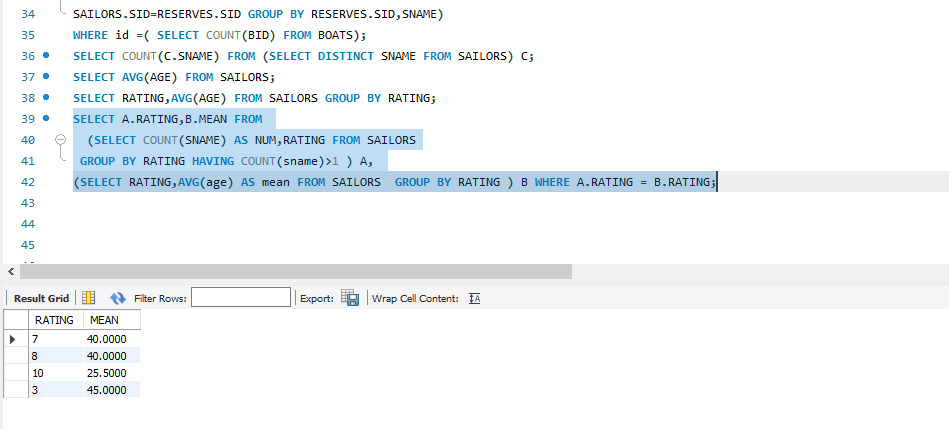
1. Find the average age of sailors for each rating level.

**OUTPUT**



1. Find the average age of sailors for each rating level that has at least two sailors.

**OUTPUT**

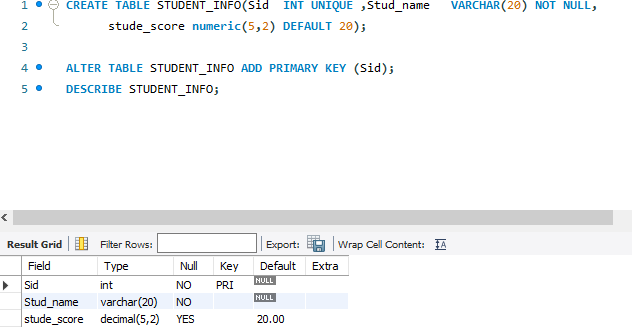


***Q.SET 2 DATE :8-6-2021***

1. Create the table STUDENT\_INFO with Columns: Sid, Stud\_name & stude\_score.

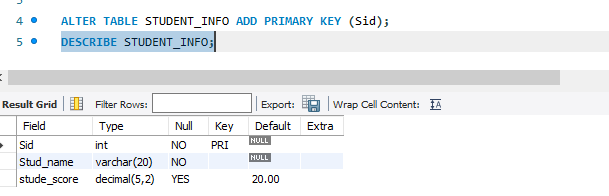
* Insert values into STUDENT\_INFO with the following constraints:Sid should be unique, Stud name NOT NULL and stude\_score DEFAULT value of 20.

**OUTPUT**



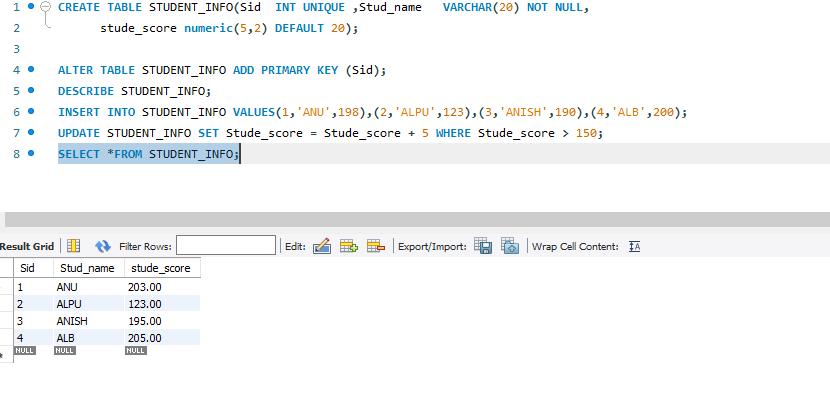
* Set Sid as primary key.

**OUTPUT**



* Update stude\_score by adding a value of 5 to stude\_score in the table STUDENT\_INFO for the rows satisfying the condition of stude\_score >150 (Using CASE)

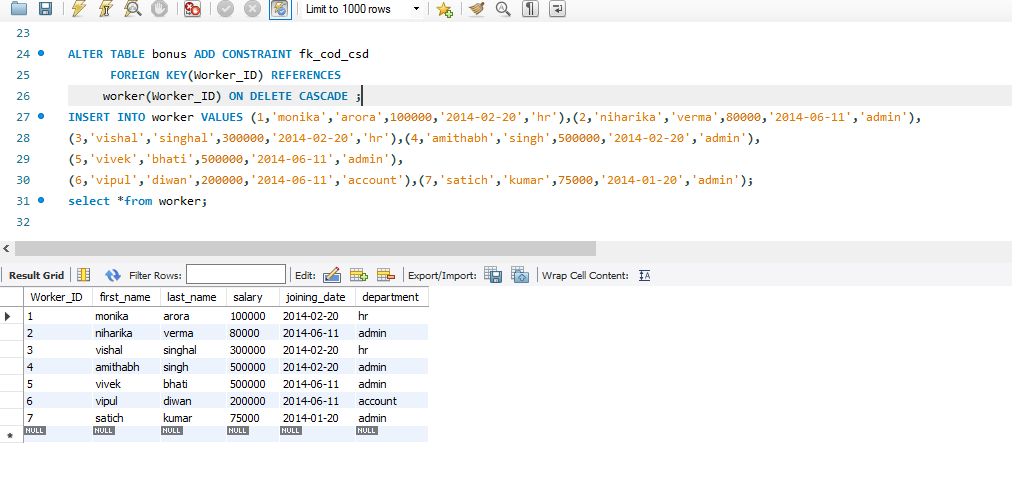
**OUTPUT**



1. Create the tables **worker** and **bonus** with the following fields. The primary key of Worker table is Worker\_ID. Set Worker\_id as foreign key of bonus on update and delete cascade constraints. Each constraint should be given a name

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **WORKER\_ID** | **FIRST\_NAME** | **LAST\_NAME** | **SALARY** | **JOINING\_DATE** | **DEPARTMENT** |
| 1 | Monika | Arora | 100000 | 2014-02-20 | HR |
| 2 | Niharika | Verma | 80000 | 2014-06-11 | Admin |
| 3 | Vishal | Singhal | 300000 | 2014-02-20 | HR |
| 4 | Amitabh | Singh | 500000 | 2014-02-20 | Admin |
| 5 | Vivek | Bhati | 500000 | 2014-06-11 | Admin |
| 6 | Vipul | Diwan | 200000 | 2014-06-11 | Account |
| 7 | Satish | Kumar | 75000 | 2014-01-20 | Account |
| 8 | Geetika | Chauhan | 90000 | 2014-04-11 | Admin |

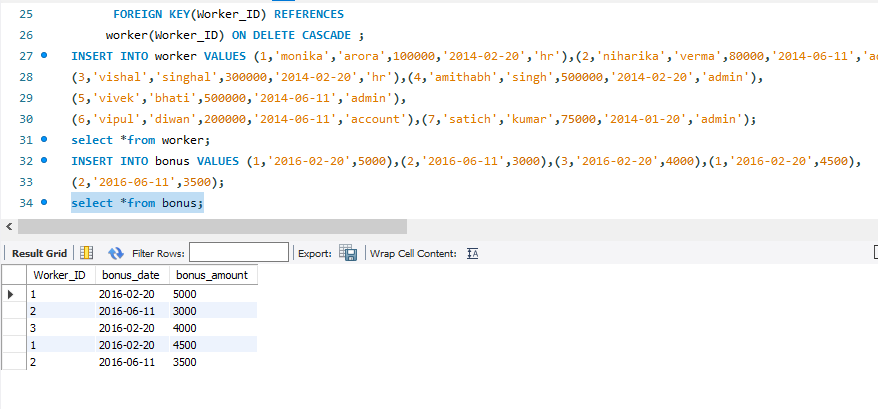
**Output**



1. **Sample Table – Bonus**

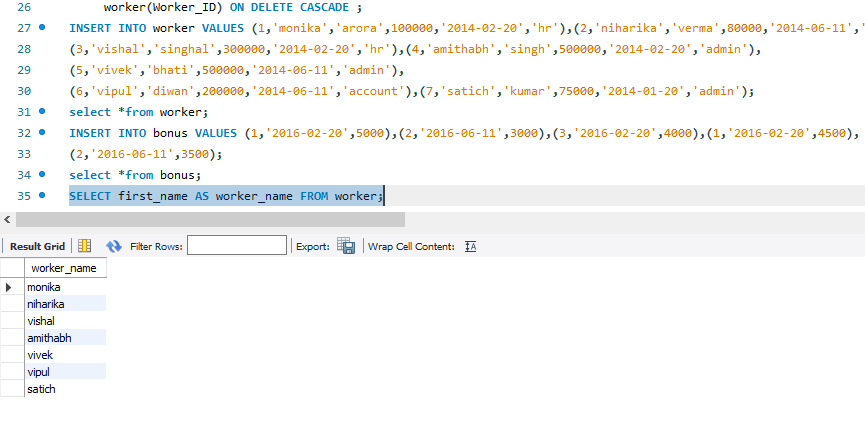
|  |  |  |
| --- | --- | --- |
| **WORKER\_ID** | **BONUS\_DATE** | **BONUS\_AMOUNT** |
| 1 | 2016-02-20 | 5000 |
| 2 | 2016-06-11 | 3000 |
| 3 | 2016-02-20 | 4000 |
| 1 | 2016-02-20 | 4500 |
| 2 | 2016-06-11 | 3500 |

**Output**



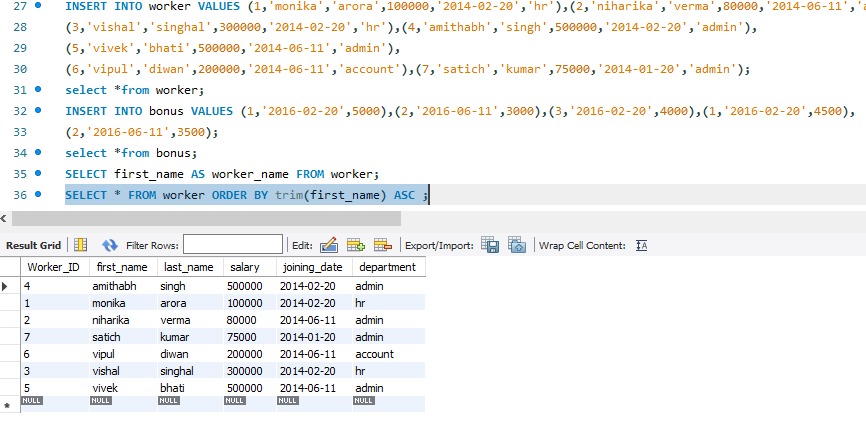
1. Write An SQL Query To Fetch “FIRST\_NAME” From Worker Table Using The Alias Name As <WORKER\_NAME>.

**Output**



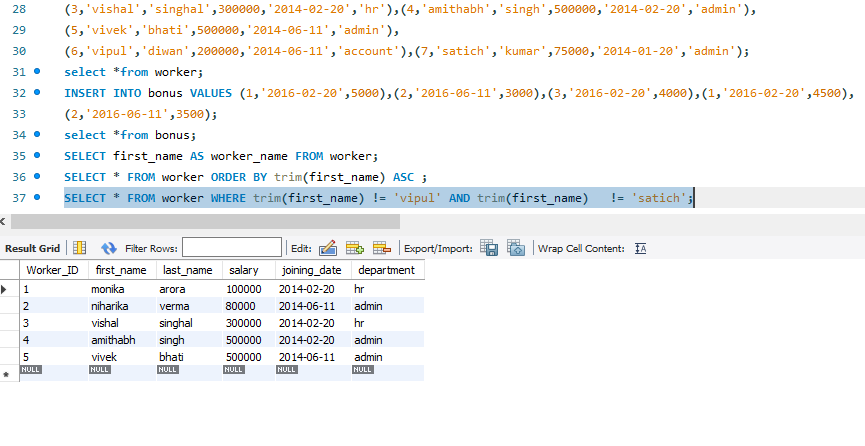
1. Write An SQL Query To Print All Worker Details From The Worker Table Order By FIRST\_NAME Ascending

**Output**



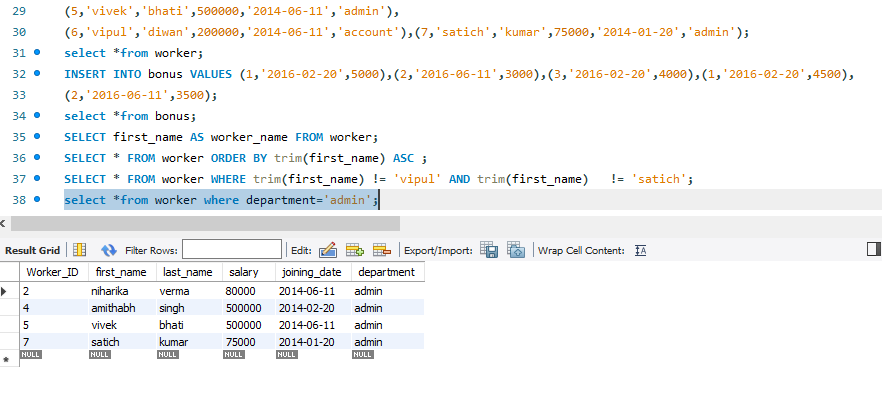
1. Write An SQL Query To Print Details Of Workers Excluding First Names, “Vipul” And “Satish” From Worker Table.

**Output**



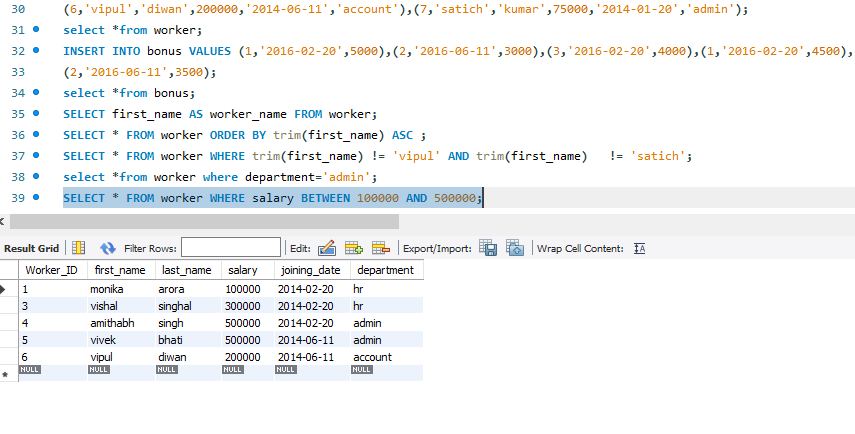
1. Write An SQL Query To Print Details Of Workers With DEPARTMENT Name As “Admin”.

**Output**



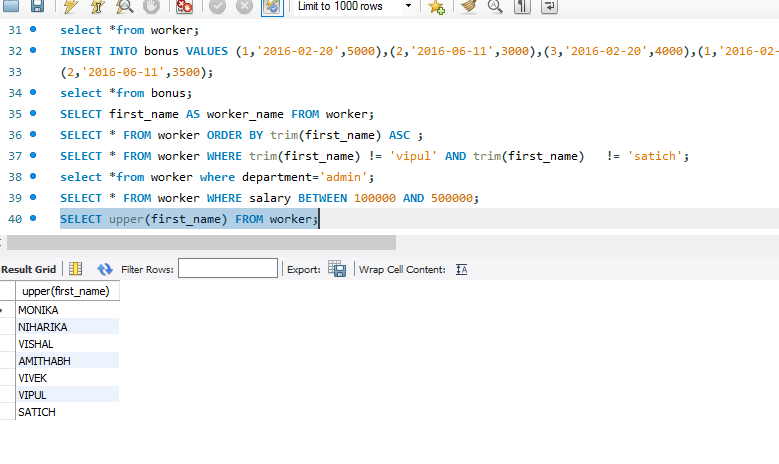
1. Write An SQL Query To Print Details Of The Workers Whose SALARY Lies Between 100000 And 500000

**Output**



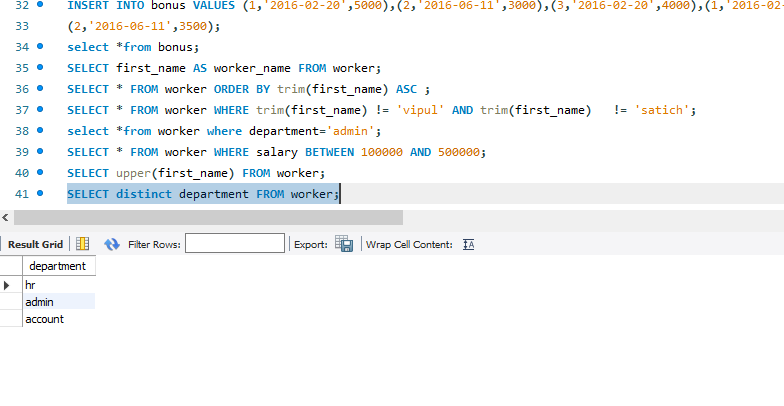
1. Write An SQL Query To Fetch “FIRST\_NAME” From Worker Table In Upper Case. (upper())

**Output**



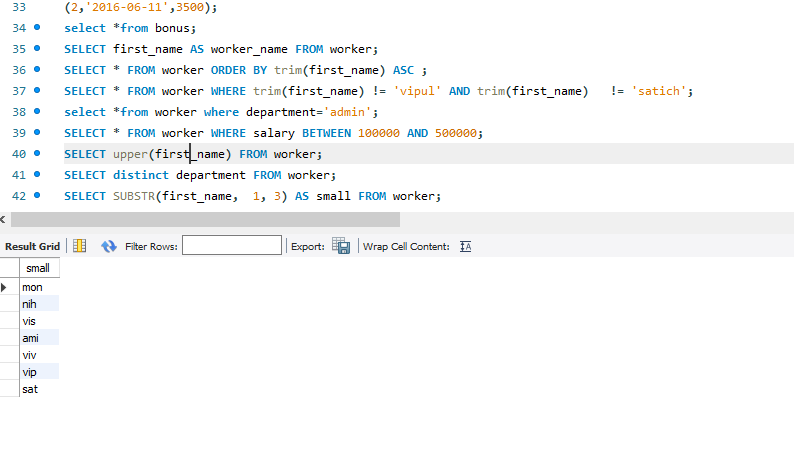
1. Write An SQL Query To Fetch Unique Values Of DEPARTMENT From Worker Table.

**Output**



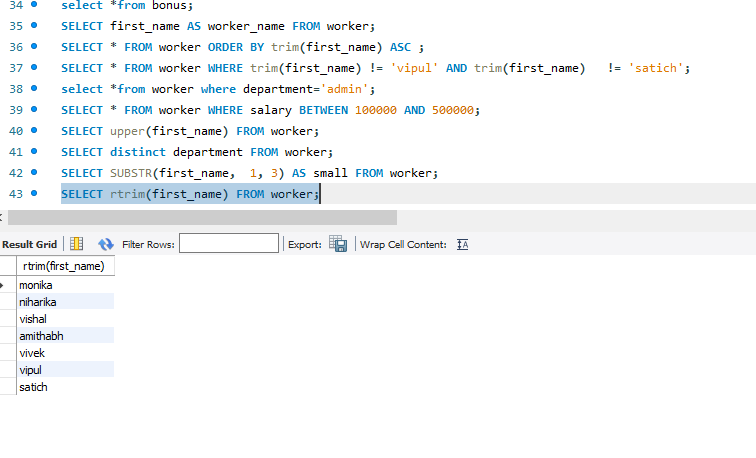
1. Write An SQL Query To Print First Three Characters Of  FIRST\_NAME From Worker Table.( substring())

**Output**



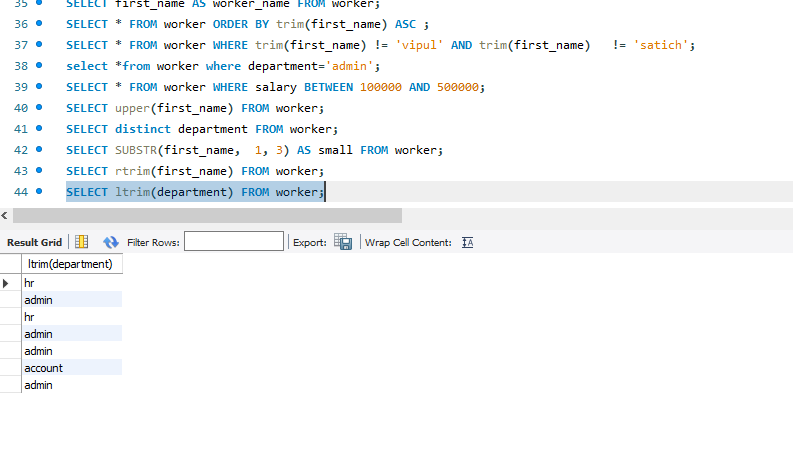
1. Write An SQL Query To Print The FIRST\_NAME From Worker Table After Removing White Spaces From The Right Side( RTRIM ( ))

**Output**



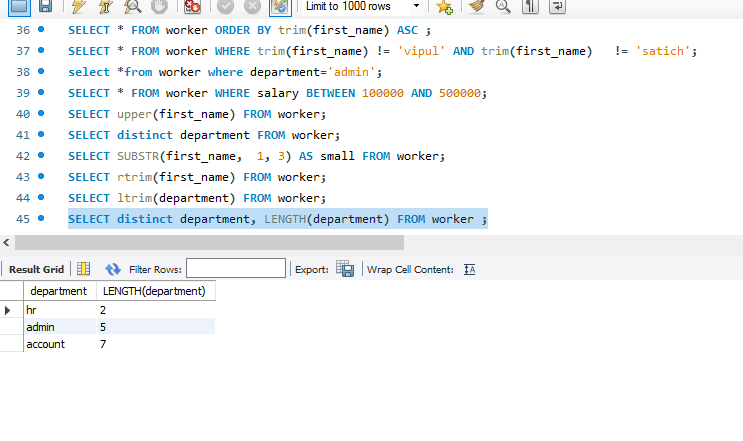
1. Write An SQL Query To Print The DEPARTMENT From Worker Table After Removing White Spaces From The Left Side. ( LTRIM ( ))

**Output**



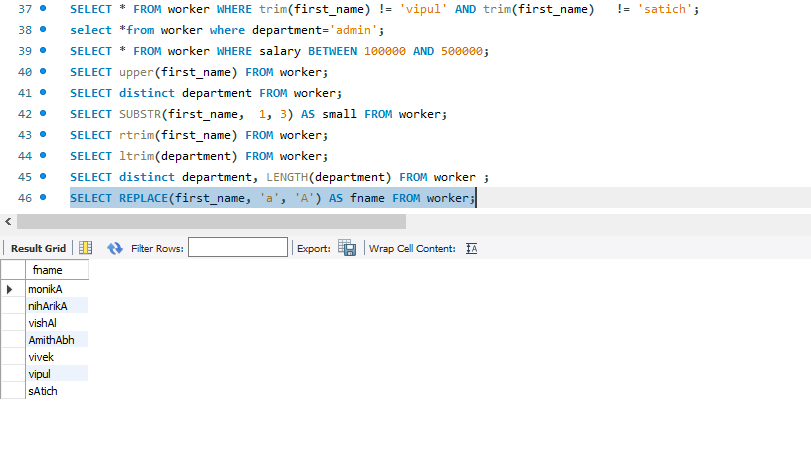
1. Write An SQL Query That Fetches The Unique Values Of DEPARTMENT From Worker Table And Prints Its Length.( length())

**Output**



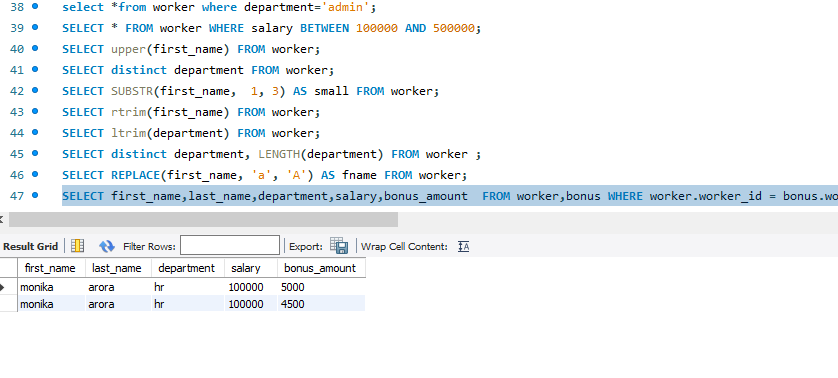
1. Write An SQL Query To Print The FIRST\_NAME From Worker Table After Replacing ‘a’ With ‘A’.( REPLACE( ))

**Output**



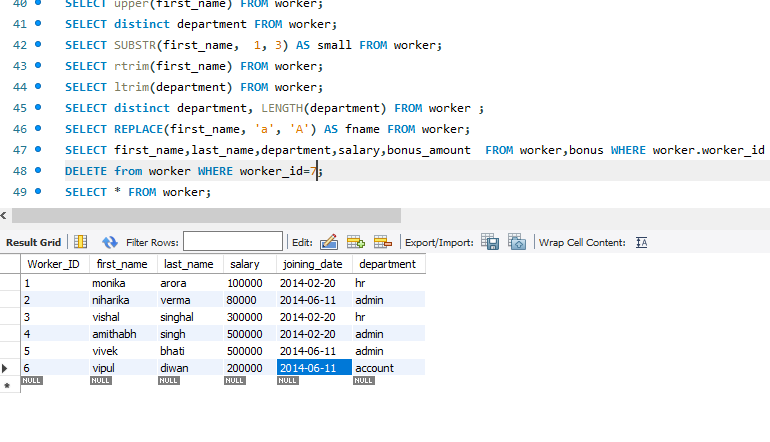
1. Find the First name , last name ,Department, Salary and Bonus of employees whose bonus amount is greater than 4000

**Output**



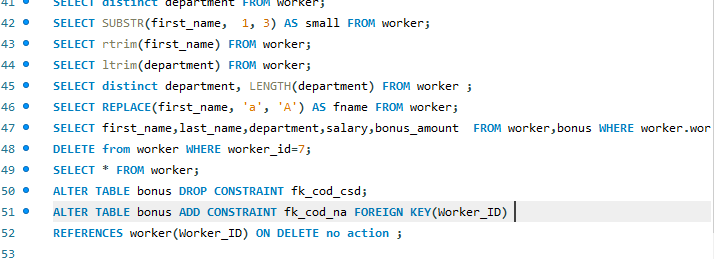
1. Delete the employee with worker\_id=7 from worker and display the details of both tables.

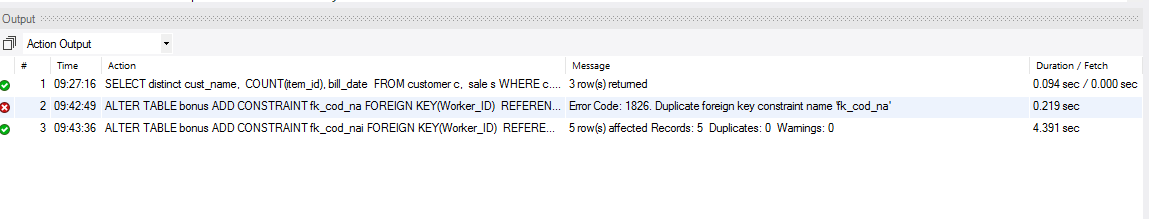
**Output**



1. Drop the foreign key constraint and add a new referential integrity constraint with ‘on update or delete with no action’

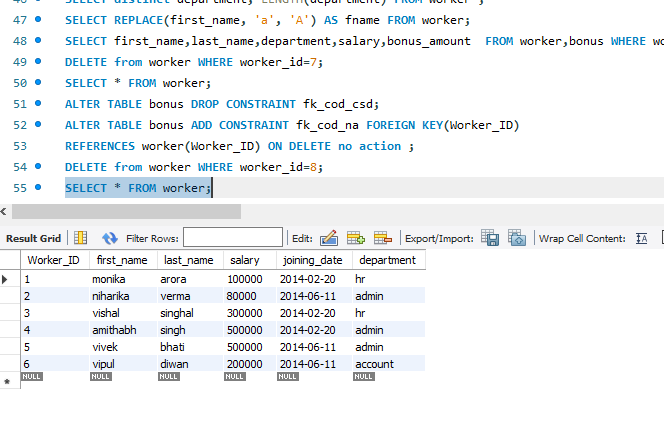
**Output**





1. Delete the employee with worker\_id = 8 from worker.

**Output**



***Q.SET 3 date :11-06-2021***

Create the tables given below and execute the queries:

**Customer(Cust id : integer, cust\_name: string)**

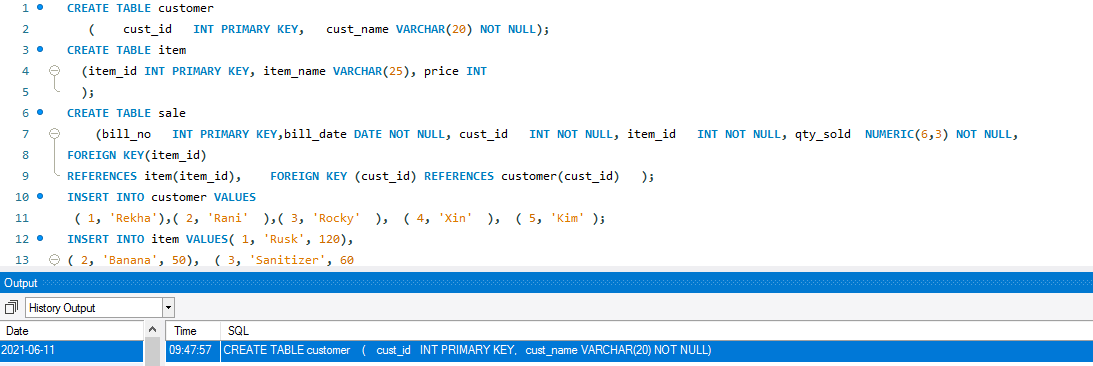
**Item(item\_id: integer, item\_name: string, price: integer)**

**Sale(bill\_no: integer, bill\_date: date, cust\_id: integer, item\_id: integer, qty\_sold: integer)**

For the above schema, perform the following—

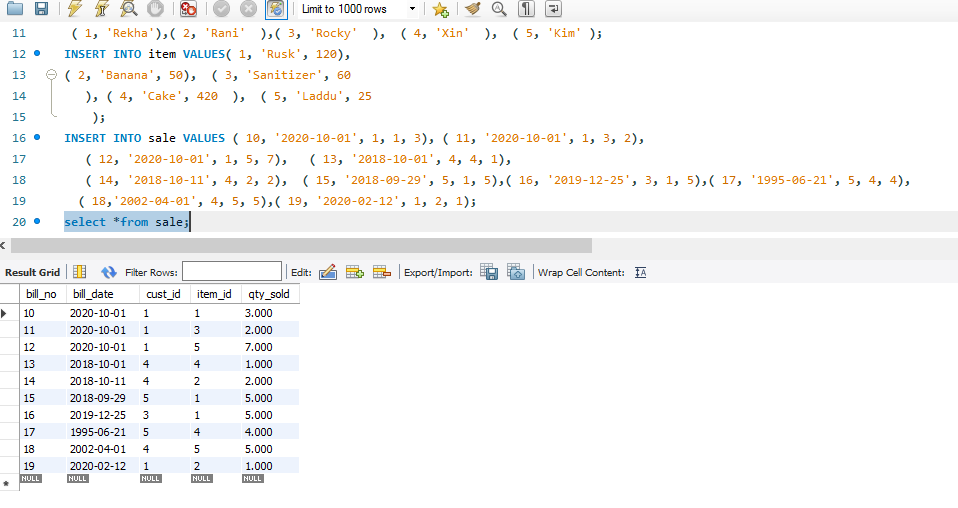
1. Create the tables with the appropriate integrity constraints

**Output**



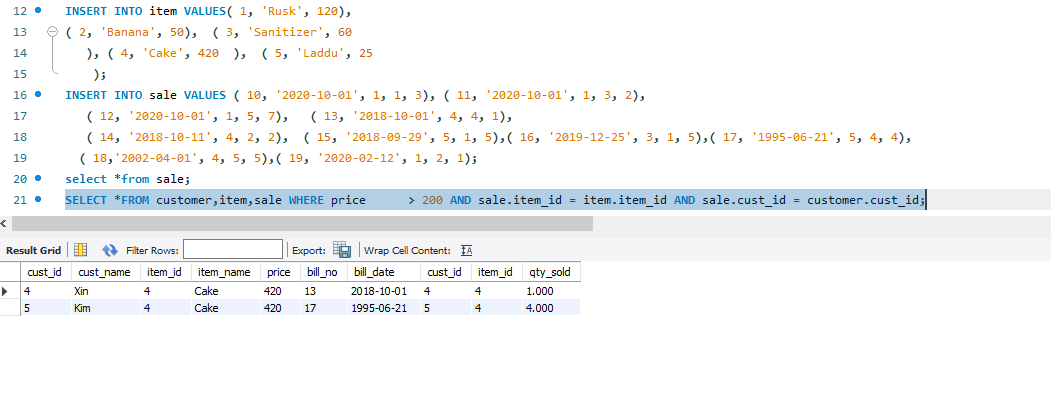
1. Insert details of 5 customers, 5 items and 10 sales details. There should be one customer ‘rekha’ who had purchased 3 different products on the same date. And there should be atleast one customer who had purchased 2 different products on the same date in the year ‘2018’.

**Output**



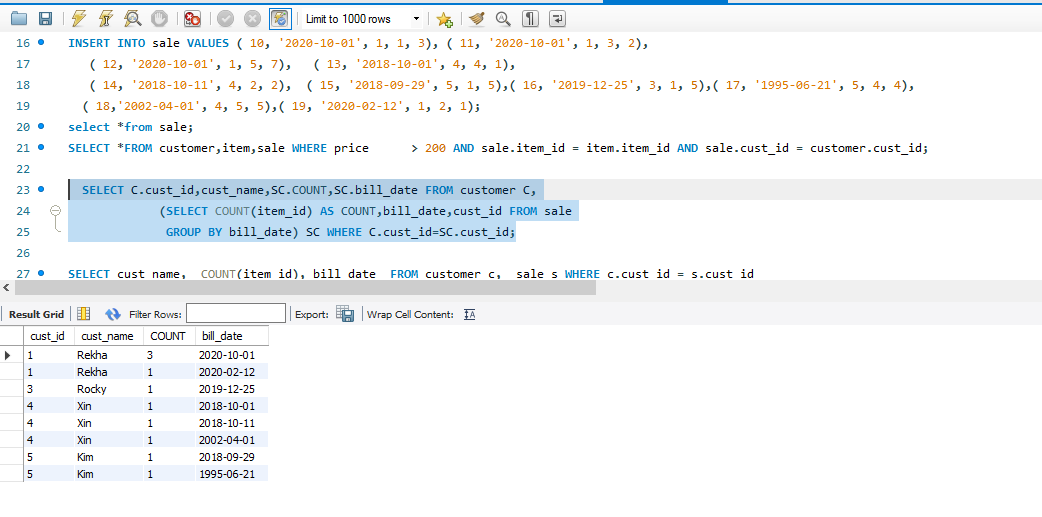
1. List the details of the customer who have bought a product which has a price>200 .

**Output**



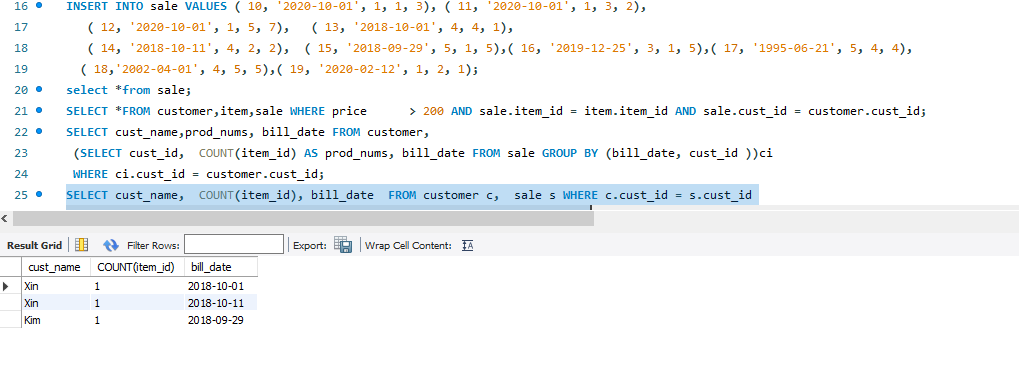
1. Give a count of how many products have been bought by each customer group by bill date.

**Output**



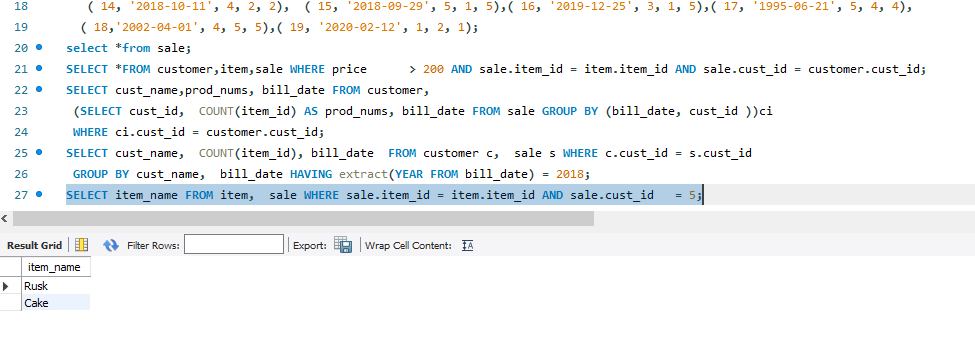
1. Give a count of how many products have been bought by each customer group by bill date only for the year 2018.

**Output**



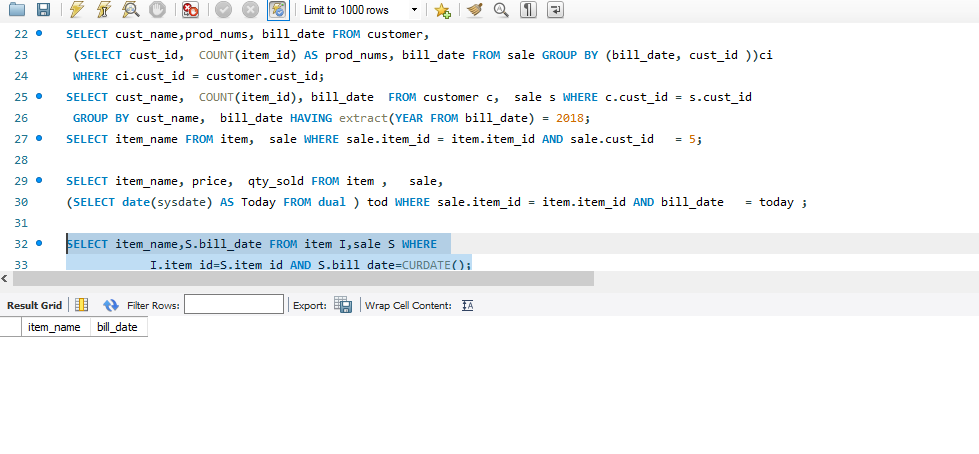
1. Give a list of products bought by a customer having cust\_id as 5

**Output**



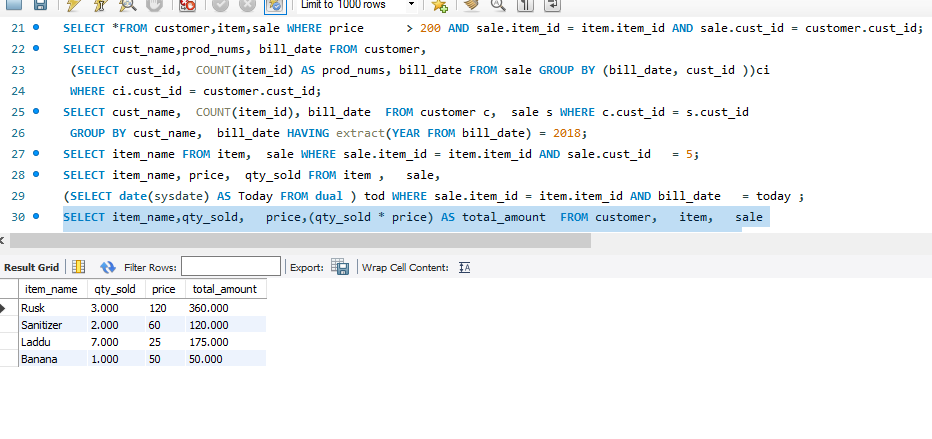
1. List the item details which are sold as of today

**Output**



1. Print the bill in a neat format with the quantity sold, price of the item and the final amount of customer ‘rekha’

**Output**



**--------------------------------------------**