Week 02: SQL Practice Tasks

Online IDE for practice: <http://www.sqlfiddle.com/>

Practice document: <https://github.com/NYU-DataScienceBootCamp/Week-2-SQL/blob/main/SQL_Practice.pdf>

|  |
| --- |
| **NOTE:** Make sure you answer the queries in the boxes given and paste screenshots in the output box.  **The solution queries will be posted on June 24th before the session** |

# Input Data

Use the database which was discussed during the session and feel free to change the attributes of the tables. Make sure that the following conditions are satisfied:

* There are three “tables”. One for storing Employee Details, One for Bonus, and One for Employee Title.
* There are at least 12 employees in the table which stores Employee Details.

NOTE: Make sure that you paste your input data in the box given below

|  |
| --- |
| /\* Comment with more than one line  \*/  CREATE TABLE Employee (  EMPLOYEE\_ID INT NOT NULL PRIMARY KEY AUTO\_INCREMENT,  FIRST\_NAME CHAR(25),  LAST\_NAME CHAR(25),  SALARY INT(15),  JOINING\_DATE DATETIME,  DEPARTMENT CHAR(25)  );  INSERT INTO Employee  (EMPLOYEE\_ID, FIRST\_NAME, LAST\_NAME, SALARY, JOINING\_DATE,  DEPARTMENT) VALUES  (001, 'Neville', 'Longbottom', 100000, '14-02-20 09.00.00', 'HR'),  (002, 'Ronald’, 'Weasley', 80000, '14-06-11 09.00.00', 'Admin'),  (003, 'Hermoine', 'Granger', 300000, '14-02-20 09.00.00', 'HR'),  (004, 'Harry', 'Potter', 500000, '14-02-20 09.00.00', 'Admin'),  (005, 'Severus', 'Snape', 500000, '14-06-11 09.00.00', 'Admin'),  (006, ‘Luna', 'Lovegood', 200000, '14-06-11 09.00.00', 'Account'),  (007, 'Draco', 'Malfoy', 75000, '14-01-20 09.00.00', 'Account'),  (008, ‘Minerva’, 'Mcgonagall', 90000, '14-04-11 09.00.00', 'Admin'),  (009, 'Jack', 'Dorsey', 1000000, '01-01-02 09.00.00', 'Admin'),  (010, 'Jill', 'Boxer', 4200, '18-01-20 09.00.00', 'HR'),  (011, 'Matthew', 'Parry', 1111111, '10-01-12 09.00.00', 'HR'),  (012, 'Carl', 'Devonport', 420, '01-01-02 09.00.00', 'Account'),  (013, 'Gul', 'Panag', 90000, '22-04-15 09.00.00', 'Account');  CREATE TABLE Bonus (  EMPLOYEE\_REF\_ID INT,  BONUS\_AMOUNT INT(10),  BONUS\_DATE DATETIME,  FOREIGN KEY (EMPLOYEE\_REF\_ID)  REFERENCES Employee(EMPLOYEE\_ID)  ON DELETE CASCADE  );  INSERT INTO Bonus  (EMPLOYEE\_REF\_ID, BONUS\_AMOUNT, BONUS\_DATE) VALUES  (001, 5000, '16-02-20'),  (002, 3000, '17-06-11'),  (003, 4000, '17-02-20'),  (001, 4500, '18-02-20'),  (002, 3500, '19-06-02'),  (008, 3500, '20-06-16'),  (007, 3500, '21-06-11'),  (012, 3500, '22-06-14'),  (013, 3500, '23-06-13'),  (013, 3500, '24-06-12'),  (001, 3500, '25-06-01'),  (013, 3500, '01-08-21');  CREATE TABLE Title (  EMPLOYEE\_REF\_ID INT,  EMPLOYEE\_TITLE CHAR(25),  AFFECTED\_FROM DATETIME,  FOREIGN KEY (EMPLOYEE\_REF\_ID)  REFERENCES Employee(EMPLOYEE\_ID)  ON DELETE CASCADE  );  INSERT INTO Title  (EMPLOYEE\_REF\_ID, EMPLOYEE\_TITLE, AFFECTED\_FROM) VALUES  (001, 'Manager', '2016-02-20 00:00:00'),  (002, 'Executive', '2016-06-11 00:00:00'),  (008, 'Executive', '2016-06-11 00:00:00'),  (005, 'Manager', '2016-06-11 00:00:00'),  (004, 'Assistant Manager', '2016-06-11 00:00:00'),  (007, 'Executive', '2016-06-11 00:00:00'),  (006, 'Lead', '2016-06-11 00:00:00'),  (003, 'Lead', '2016-06-11 00:00:00'); |

# Tasks

## SELECTing data

* Display the entire table containing the details of all the Employees  
    
  **QUERY:**

|  |
| --- |
| SELECT \* FROM Employee |

**OUTPUT:**

|  |
| --- |
|  |

* Write a query to fetch “FIRST\_NAME” from the Employees table in the UPPER CASE  
    
  **QUERY:**

|  |
| --- |
| SELECT UPPER(FIRST\_NAME) AS UPPER\_FIRST\_NAME FROM Employee |

**OUTPUT:**

|  |
| --- |
|  |

## GROUPing them together

* Write a query to fetch the number of Employees for each department in the descending order  
    
  **QUERY:**

|  |
| --- |
| SELECT DEPARTMENT, COUNT(EMPLOYEE\_ID) AS EMPLOYEE\_ID\_COUNT FROM Employee  GROUP BY DEPARTMENT ORDER BY EMPLOYEE\_ID\_COUNT DESC |

**OUTPUT:**

|  |
| --- |
|  |

## Using WHERE somewhere

* Write a query to fetch the names of the Employees with salaries >= 90000 and <= 200000  
    
  **QUERY:**

|  |
| --- |
| SELECT FIRST\_NAME, LAST\_NAME, SALARY FROM Employee  WHERE (SALARY >= 90000) AND (SALARY <= 200000) |

**OUTPUT:**

|  |
| --- |
|  |

## JOINing the tables

* Write a query to print details of Employees who are also “Managers”  
    
  **QUERY:**

|  |
| --- |
| SELECT \* from Employee  INNER JOIN Title on Employee.EMPLOYEE\_ID = Title.EMPLOYEE\_REF\_ID  WHERE Title.EMPLOYEE\_TITLE = 'Manager' |

**OUTPUT:**

|  |
| --- |
|  |

## COPYing

* Write an SQL query to clone a new table from another table  
    
  **QUERY:**

|  |
| --- |
| create table clone1 select \* from employee;  select \* from clone1; |

**OUTPUT:**

|  |
| --- |
|  |

## Aliasing

* Find the average salary of employees in each department and name the AVG(SALARY) column as “AverageSalary”  
    
  **QUERY:**

|  |
| --- |
| select department, avg(salary) as AverageSalary from employee  group by department order by AverageSalary desc |

**OUTPUT:**

|  |
| --- |
|  |

## Some other stuff

* Write an SQL query to show the second-highest salary from a table  
    
  **QUERY:**

|  |
| --- |
| select max(salary) as second\_highest\_salary  from employee where salary < (select max(salary) from employee); |

**OUTPUT:**

|  |
| --- |
|  |

* Write an SQL query to show one row twice in results from a table

**QUERY:**

|  |
| --- |
| select \* from employee where employee\_id=001 or employee\_id=002  union all  select \* from employee where employee\_id=001 or employee\_id=002 order by employee\_id |

**OUTPUT:**

|  |
| --- |
|  |

* Write an SQL query to fetch the departments that have less than five people in it  
    
  **QUERY:**

|  |
| --- |
| select department from employee  group by department  having count(employee\_id) < 5; |

**OUTPUT:**

|  |
| --- |
|  |

* Write an SQL query to fetch the last five records from a table  
    
  **QUERY:**

|  |
| --- |
| select (count(\*)-5) c from employee;  select \* from employee  limit 5 offset 8 |

**OUTPUT:**

|  |
| --- |
|  |