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

|  |
| --- |
| 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, 'Susan', 'Bones', 100000, '14-02-20 09.00.00', 'HR'),  (002, 'Gregory', 'Goyle', 80000, '14-06-11 09.00.00', 'Admin'),  (003, 'Hermione', '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, 'Dean', 'Thomas', 250000, '14-08-05 09.00.00', 'Account'),  (009, 'Sirius', 'Black', 95000, '14-11-16 09.00.00', 'HR'),  (010, 'Romilda', 'Vane', 130000, '14-07-30 09.00.00', 'Admin'),  (011, 'Albus', 'Dumbledore', 700000, '14-03-13 09.00.00', 'HR'),  (012, 'Septima', 'Vector', 90000, '14-04-11 09.00.00', 'Admin');  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, 5500, '20-02-20'),  (002, 3000, '19-06-12'),  (003, 4200, '21-02-20'),  (001, 4500, '17-02-20'),  (002, 3500, '16-06-11'),  (006, 5000, '16-02-20'),  (007, 3000, '16-06-11'),  (008, 4000, '16-02-20'),  (011, 4500, '16-02-20'),  (012, 3500, '16-06-11');  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, 'Lead', '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'),  (009, 'Manager', '2016-06-11 00:00:00'),  (010, 'Assistant Manager', '2016-06-11 00:00:00'),  (011, 'Executive', '2016-06-11 00:00:00'),  (012, 'Executive', '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:**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| EMPLOYEE\_ID | FIRST\_NAME | LAST\_NAME | SALARY | JOINING\_DATE | DEPARTMENT |
| 1 | Susan | Bones | 100000 | 2014-02-20T09:00:00Z | HR |
| 2 | Gregory | Goyle | 80000 | 2014-06-11T09:00:00Z | Account |
| 3 | Hermione | Granger | 300000 | 2014-02-20T09:00:00Z | HR |
| 4 | Harry | Potter | 500000 | 2014-02-20T09:00:00Z | Admin |
| 5 | Severus | Snape | 500000 | 2014-06-11T09:00:00Z | Admin |
| 6 | Luna | Lovegood | 200000 | 2014-06-11T09:00:00Z | Account |
| 7 | Draco | Malfoy | 75000 | 2014-01-20T09:00:00Z | Account |
| 8 | Dean | Thomas | 250000 | 2014-08-05T09:00:00Z | Account |
| 9 | Sirius | Black | 95000 | 2014-11-16T09:00:00Z | HR |
| 10 | Romilda | Vane | 130000 | 2014-07-30T09:00:00Z | Admin |
| 11 | Albus | Dumbledore | 700000 | 2014-03-13T09:00:00Z | HR |
| 12 | Septima | Vector | 90000 | 2014-04-11T09:00:00Z | Admin |

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

|  |
| --- |
| Select upper(FIRST\_NAME) as UPPERCASE\_FIRST\_NAME from employee; |

**OUTPUT:**

|  |
| --- |
| **UPPERCASE\_FIRST\_NAME** |
| SUSAN |
| GREGORY |
| HERMIONE |
| HARRY |
| SEVERUS |
| LUNA |
| DRACO |
| DEAN |
| SIRIUS |
| ROMILDA |
| ALBUS |
| SEPTIMA |

## 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 Count\_Of\_Employees from Employee group by Department order by Count\_Of\_Employees; |

**OUTPUT:**

|  |  |
| --- | --- |
| **Department** | **Count\_Of\_Employees** |
| Admin | 4 |
| HR | 4 |
| Account | 4 |

## Using WHERE somewhere

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

|  |
| --- |
| Select Concat(FIRST\_NAME, ' ', LAST\_NAME) as Employee\_names from Employee where salary >= 90000 and salary <= 200000 |

**OUTPUT:**

|  |
| --- |
| **Employee\_names** |
| Susan Bones |
| Luna Lovegood |
| Sirius Black |
| Romilda Vane |
| Septima Vector |

## JOINing the tables

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

|  |
| --- |
| select \* from employee as a  left join  title as t on a.employee\_id = t.employee\_ref\_id  where t.employee\_title = "Manager"; |

**OUTPUT:**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **EMPLOYEE\_ID** | **FIRST\_NAME** | **LAST\_NAME** | **SALARY** | **JOINING\_DATE** | **DEPARTMENT** | **EMPLOYEE\_REF\_ID** | **EMPLOYEE\_TITLE** | **AFFECTED\_FROM** |
| 1 | Susan | Bones | 100000 | 2014-02-20T09:00:00Z | HR | 1 | Manager | 2016-02-20T00:00:00Z |
| 5 | Severus | Snape | 500000 | 2014-06-11T09:00:00Z | Admin | 5 | Manager | 2016-06-11T00:00:00Z |
| 9 | Sirius | Black | 95000 | 2014-11-16T09:00:00Z | HR | 9 | Manager | 2016-06-11T00:00:00Z |

## COPYing

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

|  |
| --- |
| CREATE TABLE EMPLOYEES\_CLONE as  SELECT \*  from EMPLOYEE; |

**OUTPUT:**

|  |
| --- |
| DDL and DML statements are not allowed in the query panel for MySQL; only SELECT statements are allowed. Put DDL and DML in the schema panel. |

## 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; |

**OUTPUT:**

|  |  |
| --- | --- |
| **DEPARTMENT** | **AverageSalary** |
| Account | 150750 |
| Admin | 191250 |
| HR | 146000 |

## 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:**

|  |
| --- |
| **second\_highest\_salary** |
| 300000 |

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

**QUERY:**

|  |
| --- |
| SELECT \* FROM EMPLOYEE  UNION ALL  SELECT \* FROM EMPLOYEE  Order by 1;; |

**OUTPUT:**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **EMPLOYEE\_ID** | **FIRST\_NAME** | **LAST\_NAME** | **SALARY** | **JOINING\_DATE** | **DEPARTMENT** |
| **1** | **Susan** | **Bones** | **100000** | **2014-02-20T09:00:00Z** | **HR** |
| **1** | **Susan** | **Bones** | **100000** | **2014-02-20T09:00:00Z** | **HR** |
| **2** | **Gregory** | **Goyle** | **78000** | **2014-06-11T09:00:00Z** | **Account** |
| **2** | **Gregory** | **Goyle** | **78000** | **2014-06-11T09:00:00Z** | **Account** |
| **3** | **Hermione** | **Granger** | **300000** | **2014-02-20T09:00:00Z** | **HR** |
| **3** | **Hermione** | **Granger** | **300000** | **2014-02-20T09:00:00Z** | **HR** |
| **4** | **Harry** | **Potter** | **500000** | **2014-02-20T09:00:00Z** | **Admin** |
| **4** | **Harry** | **Potter** | **500000** | **2014-02-20T09:00:00Z** | **Admin** |
| **5** | **Severus** | **Snape** | **45000** | **2014-06-11T09:00:00Z** | **Admin** |
| **5** | **Severus** | **Snape** | **45000** | **2014-06-11T09:00:00Z** | **Admin** |
| **6** | **Luna** | **Lovegood** | **200000** | **2014-06-11T09:00:00Z** | **Account** |
| **6** | **Luna** | **Lovegood** | **200000** | **2014-06-11T09:00:00Z** | **Account** |
| **7** | **Draco** | **Malfoy** | **75000** | **2014-01-20T09:00:00Z** | **Account** |
| **7** | **Draco** | **Malfoy** | **75000** | **2014-01-20T09:00:00Z** | **Account** |
| **8** | **Dean** | **Thomas** | **250000** | **2014-08-05T09:00:00Z** | **Account** |
| **8** | **Dean** | **Thomas** | **250000** | **2014-08-05T09:00:00Z** | **Account** |
| **9** | **Sirius** | **Black** | **95000** | **2014-11-16T09:00:00Z** | **HR** |
| **9** | **Sirius** | **Black** | **95000** | **2014-11-16T09:00:00Z** | **HR** |
| **10** | **Romilda** | **Vane** | **130000** | **2014-07-30T09:00:00Z** | **Admin** |
| **10** | **Romilda** | **Vane** | **130000** | **2014-07-30T09:00:00Z** | **Admin** |
| **11** | **Albus** | **Dumbledore** | **89000** | **2014-03-13T09:00:00Z** | **HR** |
| **11** | **Albus** | **Dumbledore** | **89000** | **2014-03-13T09:00:00Z** | **HR** |
| **12** | **Septima** | **Vector** | **90000** | **2014-04-11T09:00:00Z** | **Admin** |
| **12** | **Septima** | **Vector** | **90000** | **2014-04-11T09:00:00Z** | **Admin** |

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

|  |
| --- |
| SELECT DEPARTMENT,COUNT(\*) AS Number\_Of\_Employees  FROM EMPLOYEE  GROUP BY DEPARTMENT  HAVING COUNT(\*) < 5 ; |

**OUTPUT:**

|  |  |
| --- | --- |
| **DEPARTMENT** | **Number\_Of\_Employees** |
| Account | 4 |
| Admin | 4 |

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

|  |
| --- |
| SELECT \* FROM EMPLOYEE ORDER BY Employee\_ID desc LIMIT 5; |

**OUTPUT:**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **EMPLOYEE\_ID** | **FIRST\_NAME** | **LAST\_NAME** | **SALARY** | **JOINING\_DATE** | **DEPARTMENT** |
| 12 | Septima | Vector | 90000 | 2014-04-11T09:00:00Z | Admin |
| 11 | Albus | Dumbledore | 89000 | 2014-03-13T09:00:00Z | HR |
| 10 | Romilda | Vane | 130000 | 2014-07-30T09:00:00Z | Admin |
| 9 | Sirius | Black | 95000 | 2014-11-16T09:00:00Z | HR |
| 8 | Dean | Thomas | 250000 | 2014-08-05T09:00:00Z | Account |