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, 'John', 'Doe', 120000, '14-04-11 09.00.00', 'Admin'),  (010, 'Steve', 'Jobs', 999999, '14-06-11 09.00.00', 'Account'),  (011, 'John', 'Wick', 50000, '14-02-20 09.00.00', 'Admin'),  (012, 'Christian', 'Wolff', 80000, '14-01-20 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, '16-06-11'),  (003, 4000, '16-02-20'),  (001, 4500, '16-02-20'),  (002, 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, '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'),  (009, 'Manager', '2016-02-20 00:00:00'),  (010, 'Executive', '2016-06-11 00:00:00'),  (011, 'Lead', '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:**

|  |
| --- |
|  |

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

|  |
| --- |
| SELECT UPPER(first\_name) FROM employee; |

**OUTPUT:**

|  |
| --- |
| **UPPER(first\_name)** |
| NEVILLE |
| RONALD |
| HERMOINE |
| HARRY |
| SEVERUS |
| LUNA |
| DRACO |
| MINERVA |
| JOHN |
| STEVE |
| JOHN |
| CHRISTIAN |

## 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 num FROM employee GROUP BY department ORDER BY num DESC; |

**OUTPUT:**

|  |  |
| --- | --- |
| **department** | **num** |
| Admin | 6 |
| Account | 4 |
| HR | 2 |

## 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:**

|  |  |  |
| --- | --- | --- |
| **first\_name** | **last\_name** | **salary** |
| Neville | Longbottom | 100000 |
| Luna | Lovegood | 200000 |
| Minerva | Mcgonagall | 90000 |
| John | Doe | 120000 |

## JOINing the tables

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

|  |
| --- |
| SELECT \*, title.employee\_title FROM employee INNER JOIN title ON employee.employee\_id = title.employee\_ref\_id AND title.employee\_title = 'Manager'; |

**OUTPUT:**

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **EMPLOYEE\_ID** | **FIRST\_NAME** | **LAST\_NAME** | **SALARY** | **JOINING\_DATE** | **DEPARTMENT** | **EMPLOYEE\_REF\_ID** | **EMPLOYEE\_TITLE** | **AFFECTED\_FROM** | **employee\_title** |
| 1 | Neville | Longbottom | 100000 | 2014-02-20T09:00:00Z | HR | 1 | Manager | 2016-02-20T00:00:00Z | Manager |
| 5 | Severus | Snape | 500000 | 2014-06-11T09:00:00Z | Admin | 5 | Manager | 2016-06-11T00:00:00Z | Manager |
| 9 | John | Doe | 120000 | 2014-04-11T09:00:00Z | Admin | 9 | Manager | 2016-02-20T00:00:00Z | Manager |

## COPYing

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

|  |
| --- |
| CREATE TABLE temp SELECT \* FROM bonus; SELECT \* FROM temp; |

**OUTPUT:**

|  |  |  |
| --- | --- | --- |
| **EMPLOYEE\_REF\_ID** | **BONUS\_AMOUNT** | **BONUS\_DATE** |
| 1 | 5000 | 2016-02-20T00:00:00Z |
| 2 | 3000 | 2016-06-11T00:00:00Z |
| 3 | 4000 | 2016-02-20T00:00:00Z |
| 1 | 4500 | 2016-02-20T00:00:00Z |
| 2 | 3500 | 2016-06-11T00:00:00Z |

## 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 | 338749.75 |
| Admin | 223333.3333 |
| HR | 200000 |

## Some other stuff

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

|  |
| --- |
| SELECT first\_name, last\_name, MAX(salary) FROM employee WHERE salary < (SELECT MAX(salary) FROM employee); |

**OUTPUT:**

|  |  |  |
| --- | --- | --- |
| **first\_name** | **last\_name** | **MAX(salary)** |
| Neville | Longbottom | 500000 |

* 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 | Neville | Longbottom | 100000 | 2014-02-20T09:00:00Z | HR |
| 1 | Neville | Longbottom | 100000 | 2014-02-20T09:00:00Z | HR |
| 2 | Ronald | Weasley | 80000 | 2014-06-11T09:00:00Z | Admin |
| 2 | Ronald | Weasley | 80000 | 2014-06-11T09:00:00Z | Admin |
| 3 | Hermoine | Granger | 300000 | 2014-02-20T09:00:00Z | HR |
| 3 | Hermoine | 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 | 500000 | 2014-06-11T09:00:00Z | Admin |
| 5 | Severus | Snape | 500000 | 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 | Minerva | Mcgonagall | 90000 | 2014-04-11T09:00:00Z | Admin |
| 8 | Minerva | Mcgonagall | 90000 | 2014-04-11T09:00:00Z | Admin |
| 9 | John | Doe | 120000 | 2014-04-11T09:00:00Z | Admin |
| 9 | John | Doe | 120000 | 2014-04-11T09:00:00Z | Admin |
| 10 | Steve | Jobs | 999999 | 2014-06-11T09:00:00Z | Account |
| 10 | Steve | Jobs | 999999 | 2014-06-11T09:00:00Z | Account |
| 11 | John | Wick | 50000 | 2014-02-20T09:00:00Z | Admin |
| 11 | John | Wick | 50000 | 2014-02-20T09:00:00Z | Admin |
| 12 | Christian | Wolff | 80000 | 2014-01-20T09:00:00Z | Account |
| 12 | Christian | Wolff | 80000 | 2014-01-20T09:00:00Z | Account |

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

|  |
| --- |
| SELECT department, COUNT(\*) AS num\_employee FROM employee GROUP BY department HAVING COUNT(\*) < 5; |

**OUTPUT:**

|  |  |
| --- | --- |
| **department** | **num\_employee** |
| Account | 4 |
| HR | 2 |

* 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 | Christian | Wolff | 80000 | 2014-01-20T09:00:00Z | Account |
| 11 | John | Wick | 50000 | 2014-02-20T09:00:00Z | Admin |
| 10 | Steve | Jobs | 999999 | 2014-06-11T09:00:00Z | Account |
| 9 | John | Doe | 120000 | 2014-04-11T09:00:00Z | Admin |
| 8 | Minerva | Mcgonagall | 90000 | 2014-04-11T09:00:00Z | Admin |