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

|  |
| --- |
|  |

# 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:**

|  |
| --- |
|  |

## GROUPing them together

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

|  |
| --- |
| select department, COUNT(department) from EMPLOYEE group by department |

**OUTPUT:**

|  |
| --- |
|  |

## 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) 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 CONCAT(first\_name, ' ', last\_name) 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 new\_table SELECT \* FROM EMPLOYEE; SELECT \* FROM new\_table |

**OUTPUT:**

|  |
| --- |
|  |

## Aliasing

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

|  |
| --- |
| SELECT DEPARTMENT, AVG(SALARY) FROM EMPLOYEE GROUP BY DEPARTMENT |

**OUTPUT:**

|  |
| --- |
|  |

## Some other stuff

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

|  |
| --- |
| Select Max(Salary) as 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 t1 cross join  (select 1 as n union all select 2) n; |

**OUTPUT:**

|  |
| --- |
|  |

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

|  |
| --- |
| SELECT DEPARTMENT, COUNT(DEPARTMENT) FROM EMPLOYEE GROUP BY (DEPARTMENT) HAVING COUNT(DEPARTMENT) < 5; |

**OUTPUT:**

|  |
| --- |
|  |

* 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:**

|  |
| --- |
|  |