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) as 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 count(employee\_id) as number\_of\_employee, department from employee group by department order by number\_of\_employee desc |

**OUTPUT:**

|  |
| --- |
|  |

## Using WHERE somewhere

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

|  |
| --- |
| select \* 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,title where employee\_id=employee\_ref\_id and employee\_title='Manager' |

**OUTPUT:**

|  |
| --- |
|  |

## COPYing

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

|  |
| --- |
| create table new\_employee\_table  select  employee\_id,first\_name,last\_name,salary,joining\_date,department from employee |

**OUTPUT:**

|  |
| --- |
|  |

## Aliasing

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

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

**OUTPUT:**

|  |
| --- |
|  |

## Some other stuff

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

|  |
| --- |
| select min(salary) from (select distinct salary from employee order by salary desc limit 2)as e |

**OUTPUT:**

|  |
| --- |
|  |

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

**QUERY:**

|  |
| --- |
| select \* from (select \* from employee union all select \* from employee) as e order by employee\_id asc |

**OUTPUT:**

|  |
| --- |
|  |

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

|  |
| --- |
| select count(first\_name) as c, department from employee group by department having c < 5 |

**OUTPUT:**

|  |
| --- |
|  |

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

|  |
| --- |
| select \* from (select \* from employee order by employee\_id desc limit 5) as e order by employee\_id asc |

**OUTPUT:**

|  |
| --- |
|  |