[](https://www.iiuc.ac.bd/)

**Assignment**

**Assignment No. – 03**

**Submission date- 16 February, 2022**

**Course Title- DBMS (Lab)**

**Course Code: CSE-2424**

Submitted to-

**Dr. Mohammad Aman Ullah**

**Assistant Professor**

01815641524

[ullah047@yahoo.com](mailto:ullah047@yahoo.com) , [aman\_cse@iiuc.ac.bd](mailto:aman_cse@iiuc.ac.bd)

Submitted by-

**MD. SOROWAR MAHABUB RABBY**

Matric ID: ***C201032*,** Section: ***4AM* ,** Semester**: *4th***

**Department of CSE (***Computer Science and Engineering***), IIUC**

**Cell:** *01834756433, 01521564157,* [*c201032@ugrad.iiuc.ac.bd*](mailto:c201032@ugrad.iiuc.ac.bd)

1. **Write a query to display the current date. Label the column Date.**

|  |
| --- |
| **Date** |
| 16-FEB-22 |
| 1 rows returned |

1. **The HR department needs a report to display the employee number, last\_name, salary, and salary increased by 15.5% (expressed as a whole number) for each employee. Label the column New Salary. Place your SQL statement in a text file named lab\_03\_02.sql.**

|  |  |  |  |
| --- | --- | --- | --- |
| **EMPLOYEE\_ID** | **LAST\_NAME** | **SALARY** | **New Salary** |
| 100 | King | 24000 | 27720 |
| 101 | Kochhar | 17000 | 19635 |
| 102 | De Haan | 17000 | 19635 |
| 103 | Hunold | 9000 | 10395 |
| 104 | Ernst | 6000 | 6930 |
| 105 | Austin | 4800 | 5544 |

107 rows returned

1. **Run your query in the file lab\_03\_02.sql.**

|  |  |  |  |
| --- | --- | --- | --- |
| **EMPLOYEE\_ID** | **LAST\_NAME** | **SALARY** | **New Salary** |
| 100 | King | 24000 | 27720 |
| 101 | Kochhar | 17000 | 19635 |
| 102 | De Haan | 17000 | 19635 |
| 103 | Hunold | 9000 | 10395 |
| 104 | Ernst | 6000 | 6930 |
| 105 | Austin | 4800 | 5544 |

107 rows returned

1. **Modify your query lab\_03\_02.sql to add a column that subtracts the old salary from the new salary. Label the column Increase. Save the contents of the file as lab\_03\_04.sql. Run the revised query.**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **EMPLOYEE\_ID** | **LAST\_NAME** | **SALARY** | **New Salary** | **Increase** |
| 100 | King | 24000 | 27720 | 3720 |
| 101 | Kochhar | 17000 | 19635 | 2635 |
| 102 | De Haan | 17000 | 19635 | 2635 |
| 103 | Hunold | 9000 | 10395 | 1395 |
| 104 | Ernst | 6000 | 6930 | 930 |

107 rows returned

1. **Write a query that displays the last name (with the first letter uppercase and all other letters lowercase) and the length of the last name for all employees whose name starts with the letters J, A, or M. Give each column an appropriate label. Sort the results by the employees’ last names.**

|  |  |
| --- | --- |
| **Name** | **Length** |
| Abel | 4 |
| Ande | 4 |
| Atkinson | 8 |
| Austin | 6 |
| Johnson | 7 |

16 rows returned

**Rewrite the query so that the user is prompted to enter a letter that starts the last name. For example, if the user enters H when prompted for a letter, then the output should show all employees whose last name starts with the letter H.**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| [**Results**](http://127.0.0.1:8080/apex/f?p=4500:1003:3095169076915553::NO:::) | [**Explain**](http://127.0.0.1:8080/apex/f?p=4500:1003:3095169076915553::NO:::) | [**Describe**](http://127.0.0.1:8080/apex/f?p=4500:1003:3095169076915553::NO:::) | [**Saved SQL**](http://127.0.0.1:8080/apex/f?p=4500:1003:3095169076915553::NO:::) | [**History**](http://127.0.0.1:8080/apex/f?p=4500:1003:3095169076915553::NO:::) |  |

|  |  |  |  |
| --- | --- | --- | --- |
|  |  |  |  |

**no data found**

1. **The HR department wants to find the length of employment for each employee. For each employee, display the last name and calculate the number of months between today and the date on which the employee was hired. Label the column MONTHS\_WORKED. Order your results by the number of months employed. Round the number of months up to the closest whole number.**

|  |  |
| --- | --- |
| **LAST\_NAME** | **MONTHS\_WORKED** |
| Kumar | 262 |
| Banda | 262 |
| Ande | 263 |
| Markle | 263 |
| Geoni | 264 |

107 rows returned

1. **Create a report that produces the following for each employee: <employee last name> earns <salary> monthly but wants <3 times salary>. Label the column Dream Salaries.**

|  |
| --- |
| **Dream Salaries** |
| King earns $24,000.00 monthly but wants $72,000.00. |
| Kochhar earns $17,000.00 monthly but wants $51,000.00. |
| De Haan earns $17,000.00 monthly but wants $51,000.00. |
| Hunold earns $9,000.00 monthly but wants $27,000.00. |
| Ernst earns $6,000.00 monthly but wants $18,000.00. |
| Austin earns $4,800.00 monthly but wants $14,400.00. |

107 rows returned

1. **Create a query to display the last name and salary for all employees. Format the salary to be 15 characters long, left-padded with $ symbol. Label the column SALARY.**

|  |  |
| --- | --- |
| **LAST\_NAME** | **SALARY** |
| King | $$$$$$$$$$24000 |
| Kochhar | $$$$$$$$$$17000 |
| De Haan | $$$$$$$$$$17000 |
| Hunold | $$$$$$$$$$$9000 |
| Ernst | $$$$$$$$$$$6000 |
| Austin | $$$$$$$$$$$4800 |

107 rows returned

1. **Display each employee’s last name, hire date, and salary review date, which is the first Monday after six months of service. Label the column REVIEW. Format the dates to appear in the format similar to “Monday, the Thirty-First of July, 2000.”**

|  |  |  |
| --- | --- | --- |
| **LAST\_NAME** | **HIRE\_DATE** | **REVIEW** |
| King | 17-JUN-87 | Monday, the Twenty-First of December, 1987 |
| Kochhar | 21-SEP-89 | Monday, the Twenty-Sixth of March, 1990 |
| De Haan | 13-JAN-93 | Monday, the Nineteenth of July, 1993 |
| Hunold | 03-JAN-90 | Monday, the Ninth of July, 1990 |
| Ernst | 21-MAY-91 | Monday, the Twenty-Fifth of November, 1991 |
| Austin | 25-JUN-97 | Monday, the Twenty-Ninth of December, 1997 |

107 rows returned

1. **Display the last name, hire date, and day of the week on which the employee started. Label the column DAY. Order the results by the day of the week, starting with Monday.**

|  |  |  |
| --- | --- | --- |
| **LAST\_NAME** | **HIRE\_DATE** | **DAY** |
| Kaufling | 01-MAY-95 | MONDAY |
| OConnell | 21-JUN-99 | MONDAY |
| Patel | 06-APR-98 | MONDAY |
| Errazuriz | 10-MAR-97 | MONDAY |
| Bernstein | 24-MAR-97 | MONDAY |

107 rows returned

**11. Create a query that displays the employees’ last names and commission amounts. If an**

**employee does not earn commission, show “No Commission.” Label the column COMM.**

|  |  |
| --- | --- |
| **LAST\_NAME** | **COMM** |
| King | No Commission |
| Kochhar | No Commission |
| De Haan | No Commission |
| Hunold | No Commission |
| Ernst | No Commission |
| Austin | No Commission |
| Pataballa | No Commission |
| Lorentz | No Commission |
| Greenberg | No Commission |
| Faviet | No Commission |
| Chen | No Commission |
| Sciarra | No Commission |
| Urman | No Commission |
| Popp | No Commission |
| Raphaely | No Commission |
| Khoo | No Commission |
| Baida | No Commission |
| Tobias | No Commission |
| Himuro | No Commission |
| Colmenares | No Commission |
| Weiss | No Commission |
| Fripp | No Commission |
| Kaufling | No Commission |
| Vollman | No Commission |
| Mourgos | No Commission |
| Nayer | No Commission |
| Mikkilineni | No Commission |
| Landry | No Commission |
| Markle | No Commission |
| Bissot | No Commission |
| Atkinson | No Commission |
| Marlow | No Commission |
| Olson | No Commission |
| Mallin | No Commission |
| Rogers | No Commission |
| Gee | No Commission |
| Philtanker | No Commission |
| Ladwig | No Commission |
| Stiles | No Commission |
| Seo | No Commission |
| Patel | No Commission |
| Rajs | No Commission |
| Davies | No Commission |
| Matos | No Commission |
| Vargas | No Commission |
| Russell | .4 |
| Partners | .3 |
| Errazuriz | .3 |
| Cambrault | .3 |
| Zlotkey | .2 |
| Tucker | .3 |
| Bernstein | .25 |
| Hall | .25 |
| Olsen | .2 |
| Cambrault | .2 |
| Tuvault | .15 |

107 rows returned

**12. Create a query that displays the first eight characters of the employees’ last names and**

**indicates the amounts of their salaries with asterisks. Each asterisk signifies a thousand dollars. Sort the data in descending order of salary. Label the column EMPLOYEES\_AND\_THEIR\_SALARIES.**

|  |
| --- |
| **EMPLOYEES\_AND\_THEIR\_SALARIES** |
| King \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* |
| Kochhar \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* |
| De Haan \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* |
| Russell \*\*\*\*\*\*\*\*\*\*\*\*\*\* |
| Partners \*\*\*\*\*\*\*\*\*\*\*\*\* |
| Hartstei \*\*\*\*\*\*\*\*\*\*\*\*\* |
| Greenber \*\*\*\*\*\*\*\*\*\*\*\* |

107 rows returned

**13. Using the DECODE function, write a query that displays the grade of all employees based on the value of the column JOB\_ID, using the following data:**

**Job Grade**

**AD\_PRES A**

**ST\_MAN B**

**IT\_PROG C**

**SA\_REP D**

**ST\_CLERK E**

**None of the above 0**

|  |  |
| --- | --- |
| **JOB\_ID** | **GRADE** |
| AC\_ACCOUNT | 0 |
| AC\_MGR | 0 |
| AD\_ASST | 0 |
| AD\_PRES | A |
| AD\_VP | 0 |
| AD\_VP | 0 |

107 rows returned

**14. Rewrite the statement in the preceding exercise using the CASE syntax.**

|  |  |
| --- | --- |
| **JOB\_ID** | **GRADE** |
| AC\_ACCOUNT | 0 |
| AC\_MGR | 0 |
| AD\_ASST | 0 |
| AD\_PRES | A |
| AD\_VP | 0 |
| AD\_VP | 0 |

107 rows returned