|  |  |
| --- | --- |
| **Topic** | Oracle SQL Language Fundamentals I |
| **Document Name** | SQL01-EX-01-05 |
|  |  |

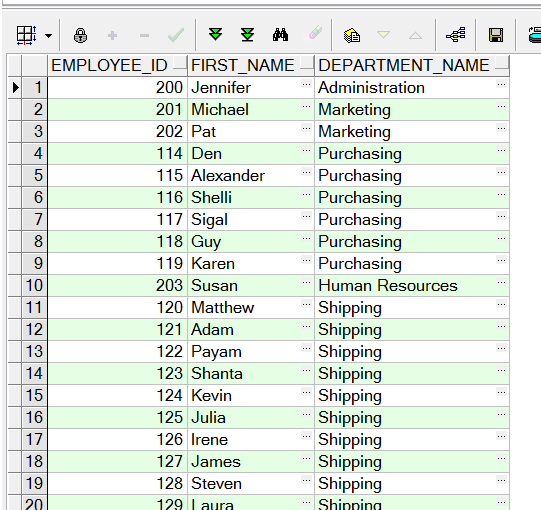
|  |  |  |  |
| --- | --- | --- | --- |
| **Document Difficulty Level** | | | |
| **Beginner** | **Junior** | **Senior** | **Expert** |
| ■ | □ | □ | □ |

# Oracle SQL Language Fundamentals I

## Exercise SQL01-EX-01:

**Definiton :** Write an SQL query that selects employee’s id, employee’s first name and employee’s department name for all employees. (Please use HR.EMPLOYEES and HR.DEPARTMENTS tables.)

**Sample Output :**



**Objectives** : To learn relations on tables and SQL language keyword JOIN.

**Exercise Keywords:** INNER JOIN, JOIN.

SELECT

e.EMPLOYEE\_ID,

e.FIRST\_NAME,

d.DEPARTMENT\_NAME

FROM

hr.EMPLOYEES e

JOIN

hr.DEPARTMENTS d

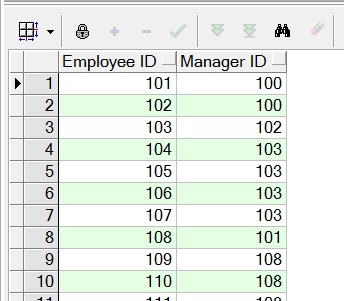
ON

e.DEPARTMENT\_ID = d.DEPARTMENT\_ID;

## Exercise SQL01-EX-02:

**Definiton :** Create a report that displays the employee’s id and their manager’s id. (Please use HR.EMPLOYEES table)

**Sample Output :**



**Objectives** : To learn SQL join logic like SELF JOIN.

SELECT

e.EMPLOYEE\_ID AS EMPLOYEE\_ID,

m.EMPLOYEE\_ID AS MANAGER\_ID

FROM

hr.EMPLOYEES e

LEFT JOIN

hr.EMPLOYEES m

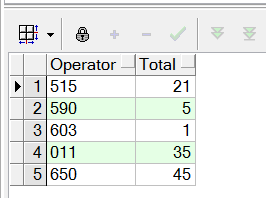
ON

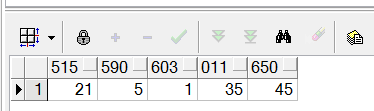
e.MANAGER\_ID = m.EMPLOYEE\_ID;

## Exercise SQL01-EX-03:

**Definiton :** For example; first three character of PHONE\_NUMBER column gives us a operator of employee. Create a report that displays the operators and their total subscriber. But we want two different displays with diffrent queries. (Please use HR.EMPLOYEES table)

**Sample Output :**





**Objectives** : To learn basic SQL keywords like COUNT, SUM, CASE.

SELECT

SUBSTR(PHONE\_NUMBER, 1, 3) AS OPERATOR,

COUNT(\*) AS TOTAL\_SUBSCRIBERS

FROM

hr.EMPLOYEES

GROUP BY

SUBSTR(PHONE\_NUMBER, 1, 3)

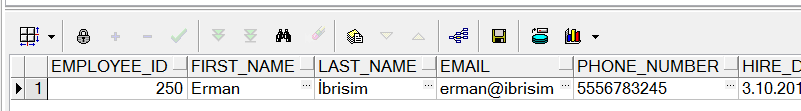
ORDER BY

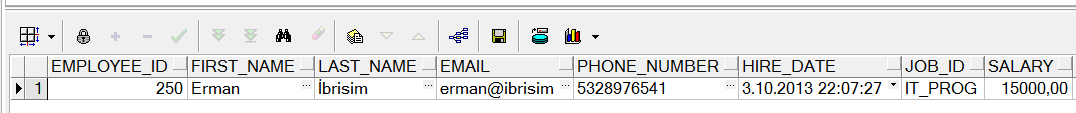
TOTAL\_SUBSCRIBERS DESC;

## Exercise SQL01-EX-04:

**Definiton :** Create a table (table name like HR.EMP) from HR.EMPLOYEES table. Insert a new row to HR.EMP table and update this employee’s phone number and salary. Delete your new row and display the HR.EMP table. Finally drop your table HR.EMP.

**Sample Output :**





**Objectives** : To learn basic SQL keywords like INSERT, UPDATE, DELETE, DROP and CREATE TABLE from table.

CREATE TABLE HR.EMP AS

SELECT \* FROM HR.EMPLOYEES WHERE 1=0;

INSERT INTO HR.EMP (EMPLOYEE\_ID, LAST\_NAME, FIRST\_NAME, EMAIL, PHONE\_NUMBER, HIRE\_DATE, JOB\_ID, SALARY, COMMISSION\_PCT, MANAGER\_ID, DEPARTMENT\_ID)

VALUES (999, 'Doe', 'Jane', 'JDOE@HR.COM', '123-456-7890', SYSDATE, 'IT\_PROG', 5000, NULL, NULL, 90);

UPDATE HR.EMP

SET PHONE\_NUMBER = '987-654-3210',

SALARY = 5500

WHERE EMPLOYEE\_ID = 999;

DELETE FROM HR.EMP

WHERE EMPLOYEE\_ID = 999;

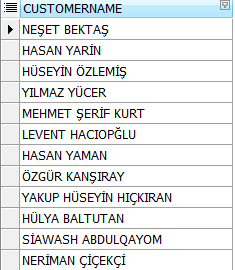
SELECT \* FROM HR.EMP;

DROP TABLE HR.EMP;

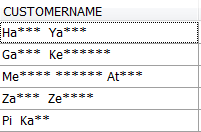
## Exercise SQL01-EX-05:

**Definiton :**

Select employees’ first name and last name as masked with “\*” character as shown in sample output below.



**Sample Output :**



**Objectives** : To learn basic SQL functions like length, substr, instr, trim, initcap, rpad, lpad, regexp\_replace, regexp\_substr

SELECT

EMPLOYEE\_ID,

SUBSTR(FIRST\_NAME, 1, 1) || RPAD('\*', LENGTH(FIRST\_NAME) - 1, '\*') AS MASKED\_FIRST\_NAME,

SUBSTR(LAST\_NAME, 1, 1) || RPAD('\*', LENGTH(LAST\_NAME) - 1, '\*') AS MASKED\_LAST\_NAME

FROM

HR.EMPLOYEES;