Practice Exercise

This document provides a list of exercises to be practiced by learners. Please raise a feedback in Talent Next, should you have any queries.

|  |  |
| --- | --- |
| Skill | Oracle SQL |
| Document Type | Lab Practice Exercises |
| Author | L & D |
| Current Version | 1.0 |
| Current Version Date | 15-June-2021 |
| Status | Active |

Document Control

|  |  |  |  |
| --- | --- | --- | --- |
| Version | Change Date | Change Description | Changed By |
| 1.0 | 15-June-2021 | Added new problems on DDL commands, SQL Constraints, DML operations, Aggregate and Date Functions, Subqueries and Joins | Anup Kini |

Contents

[Practice Exercise 1](#_Toc75793855)

[Document Control 2](#_Toc75793856)

[Problem Statement 1: Getting started with SQL DDL commands 4](#_Toc75793857)

[Problem Statement 2: Working with SQL Constraints 5](#_Toc75793858)

[Problem Statement 3: Working with DML and DQL Operations 6](#_Toc75793859)

[Problem Statement 4: Working with Aggregate and Date Functions 7](#_Toc75793860)

[Problem Statement 5: Refining Selections by SQL clauses – DISTINCT, ORDER BY, GROUP BY and LIKE 8](#_Toc75793861)

[Problem Statement 6: Working Subqueries in SQL 9](#_Toc75793862)

[Problem Statement 7: SQL: Join multiple tables 10](#_Toc75793863)

Note: Every Problem Statement start in a new page

Problem Statement 1: Getting started with SQL DDL commands

**Important Instructions:**

* Refer the **Lab Access Guide - PEP Up** document for accessing the oracle sql live online playground.
* Refer <https://livesql.oracle.com/apex/f?p=590:1000> link to access online playground.
* By default, the schema used is **My Schema.**

Write the SQL query for following operations:

1. Create an **Employees** table with following columns,
   * EmpId number
   * FirstName varchar2(20)
   * LastName varchar2(20)
   * EmailId varchar2(50)
   * Gender char(1),
   * MobileNo char(10)
   * DateOfJoining date default sysdate
   * DeptId int
2. Modify the Employees table to add DOB (Date of Birth) column with type as date and Salary column with type as number(10,2).
3. Modify the Employees table to modify Salary column type as number(9,2).
4. Modify the Employees table to drop Salary column from the table.
5. Modify the Employees table to add BloodGroup column with type as varchar2(5) and set it to unused column and finally drop the unused columns.
6. Remove the Employees table.

Problem Statement 2: Working with SQL Constraints

Write the SQL commands for following operations:

1. Create an Employees table with following columns,
   * EmpId number
   * FirstName varchar2(20)
   * LastName varchar2(20)
   * EmailID varchar2(50)
   * Gender char(1),
   * MobileNo char(10)
   * DateOfJoining date default sysdate
   * DeptId int
2. Add **“not null”** constraint to FirstName column in **Employees** table.
3. Modify the Employees table to add unique constraint on EmailID and MobileNo columns.
4. Modify the Employees table to add primary key constraint on EmpId column.
5. Modify the Employees table to add check constraint on Gender column as it should accept only ‘M’ or ‘F’ values.
6. Create a **Departments** table with following columns,
   * DeptId number – primary key constraint
   * DeptName varchar2(20)
7. Modify the **Employees** table to add DeptId as a **foreign key** constraint, also set on delete to null.

Problem Statement 3: Working with DML and DQL Operations

Write the SQL commands for following operations:

1. Create an **Employees** table with following columns,
   * EmpID int
   * FirstName varchar(255)
   * LastName varchar(255)
   * Salary int
2. Insert following records into **Employees** table,

(1,'Laya','Eluri', 20000)

(2,'Sena','John', 25000)

(3,'Don','Jane', 20000)

(4,'Marvel','John', 28000)

(5,'mike','Moni', 30000)

(6,'mike','Soni', 3000)

(7,'micky','mickey', 5000)

(8,'ms','Dhoni', 8000)

1. Perform following operations on the **Employees** table:
   1. Add a new employee – (9,'kohli','Virat', 60000)
   2. Select all employees and print the firstname and Salary.
   3. Select all employees with salary less than 20k.
   4. Update salary of employee firstname ‘mike’ to 9999.
   5. Delete all employees with salary less than 8000.
2. Perform following DQL operations on the **Employees** table:
   1. Display all the employee details who have first name ‘John’.
   2. Display the salary of employees in the range of 10000 to 50000.
   3. Find employees having letters ‘Jo’ in their first name.

Problem Statement 4: Working with Aggregate and Date Functions

For the Problem Statement refer **\Code Snippet\scripts.sql** script file provided to you and execute the script on the Console.

Write the SQL Select Queries on the tables:

1. Display the SAL column of **EMP** table rounded off to the nearest thousand.
2. Display the last day of the month for every HIREDATE in EMP table.
3. Display today’s date in the following format: 7th Jan 2005.
4. Display the average SAL for all the employees whose job is CLERK.
5. Display the maximum SAL department wise.
6. Display the annual SAL for each employee.
7. Display the names of all employees hired in the month of February (of any year).
8. Display the names of all employees with the initial letter only in capital.
9. Display the first three characters of the names of all employees.
10. Display the names of all employees replacing any ‘A’ with ‘a’.

Problem Statement 5: Refining Selections by SQL clauses – DISTINCT, ORDER BY, GROUP BY and LIKE

For the Problem Statement refer **\Code Snippet\scripts.sql** script file provided to you and execute the script on the Console.

Write the SQL Select Queries on the tables:

1. Display the JOB column with duplicate values suppressed.
2. Display the rows where ENAME begins with the letter ‘A’.
3. Display the maximum SAL department wise.
4. Display the names of all employees who have a letter ‘R’ as a third letter in their name.
5. Display the minimum SAL jobwise.
6. Display all employees in descending order of ENAME.
7. Display the rows where JOB column ends with the letter ‘T’.
8. Display the DEPTNO column with duplicate values suppressed.

Problem Statement 6: Working Subqueries in SQL

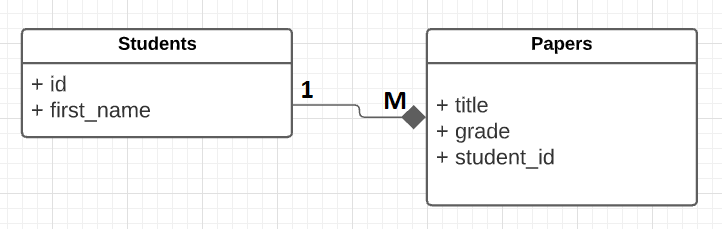
For the Problem Statement refer **\Code Snippet\scripts.sql** script file provided to you and execute the script on the Console.

Write the SQL Select Queries on the tables:

1. Display the ENAME and JOB for all employees who belong to the same DEPTNO as employee ‘KING’.
2. Find the name of the employee who is receiving the maximum salary.
3. Display the ENAME and SAL for the employee who’s getting the minimum SAL.
4. Display the third highest salary from EMP table.
5. Display the DNAME and the corresponding ENAME. All rows of DEPT table are to be displayed even if a particular DEPTNO has no employees.
6. Display the Enames and the corresponding Dnames from EMP and DEPT table.
7. Display the names of all MANAGERs who are not in department 30.

Problem Statement 7: SQL: Join multiple tables

Consider the following schema:



One Student can have multiple papers, but A paper can only belong to one Student.

In this schema FK - student\_id in PAPERS table which is referencing from PK - id from STUDENTS table.ac

1. Create above tables and establish a relationship between them.
2. Insert the following data to the respective tables:

**STUDENTS DATA**

(1, 'Caleb'), (2, 'Samantha'), (3, 'Raj'), (4, 'Carlos'), (5, 'Lisa');

**PAPERS DATA**

(1, 'My First Book Report', 60),

(1, 'My Second Book Report', 75),

(2, 'Russian Lit Through the Ages', 94),

(2, 'De Montaigne and The Art of The Essay', 98),

(4, 'Borges and Magical Realism', 89);

1. Write Join queries for below output:

|  |  |  |
| --- | --- | --- |
| **first\_name** | **Title** | **grade** |
| Samantha | De Montaigne and The Art of The Essay | 98 |
| Samantha | Russian Lit Through the Ages | 94 |
| Carlos | Borges and Magical Realism | 89 |
| Caleb | My Second Book Report | 75 |
| Caleb | My First Book Report | 60 |

1. Write Join queries for below output:

|  |  |  |
| --- | --- | --- |
| **first\_name** | **Title** | **grade** |
| Caleb | My First Book Report | 60 |
| Caleb | My First Book Report | 75 |
| Samantha | Russian Lit Through The Ages | 94 |
| Samantha | De Montaigne and The Art of The Essay | 98 |
| Raj | NULL | NULL |
| Carlos | Borges and Magical Realism | 89 |
| Lisa | NULL | NULL |

1. Write Join queries for below output:

|  |  |  |
| --- | --- | --- |
| **first\_name** | **Average** | **Passing status** |
| Samantha | 96.0000 | PASSING |
| Carlos | 89.0000 | PASSING |
| Caleb | 67.5000 | FAILING |
| Raj | 0 | FAILING |
| Lisa | 0 | FAILING |