Oracle Coding Challenge

LDMS

Author: Debadutta Rath

Contents

[Introduction 1](#_Toc164164091)

[Logical Data Model 1](#_Toc164164092)

[Creating Data base objects and Populating tables with records 2](#_Toc164164093)

[Sequence of execution 2](#_Toc164164094)

[Executable Objects 2](#_Toc164164095)

[Reports 4](#_Toc164164096)

[Conclusion 4](#_Toc164164097)

# 

# Introduction

**This document outlines the instructions needed to execute the coding challenge as described in the attached document.**



# Logical Data Model

**A diagram of a job flow

Description automatically generated**

# Creating Data base objects and Populating tables with records

## Sequence of execution

Installobjects.SQL

Create\_objects.sql

InsertRecords.sql

**Create\_objects.sql**

This script creates the following database objects

Tables: Employees, Departments and related constraints

Sequences: DEPTSEQ, EMPSEQ

Packages : pkg\_employees\_api

**InsertRecords.sql**

This script inserts departments and employees records.

**Additional Scripts**

**Reset.sql:** Clears the database objects if needed. In this case run the scripts in the following sequence

SQL>@reset.sql

SQL>@Installobjects.sql

# Executable Objects

Log Into SQLPLUS. All the objects need to be ran in SQLPLUS.

|  |  |  |
| --- | --- | --- |
| Scenario | Example | Remarks |
| Create an appropriate executable database object to allow an Employee to be created | BEGIN  pkg\_employees\_api.**pr\_create\_employee**(pi\_emp\_name => 'Deb'  ,pi\_job\_title=>'SEO'  ,pi\_manager\_id=>90001  ,pi\_date\_hired=>to\_date('01/01/1995','dd/mm/yyyy')  ,pi\_salary=>100000  ,pi\_department\_id=>1);  END; | Throws exception if any error occurs |
| Create an appropriate executable database object to allow the Salary for an Employee to be increased or decreased by a percentage | Increase  ========  Execute pkg\_objects\_api.pr\_change\_salary\_by\_pct(pi\_employee\_id => 90005,pi\_sal\_pct => 10);  Decrease  ==========  Execute pkg\_objects\_api.**pr\_change\_salary\_by\_pct**(pi\_employee\_id => 90005,pi\_sal\_pct => -10); |  |
| Create an appropriate executable database object to allow the transfer of an Employee to a different Department | Execute pkg\_employees\_api.**pr\_transfer\_employee**(pi\_employee\_id => 90005,pi\_department\_id => 4); |  |
| Create an appropriate executable database object to return the Salary for an Employee. | select pkg\_employees\_api.**f\_get\_salary**(90005) from dual; |  |

# Reports

1. **Report name:** RepEmployee4Dept

Write a report to show all Employees for a Department

Run the report as follows:

SQL> RepEmployee4Dept

Enter individual department number to print the employee details.

If nothing entered, the report will display employee details for all the department.

The report output will be stored as a file “EmpDeptReport.lst” in the current directory.

1. **Report name:** RepEmployee4Dept

Write a report to show the total of Employee Salary for a Department

Run the report as follows:

SQL> RepEmpSalary4Dept

Enter individual department number to print the employee details.

If nothing entered, the report will display employee details for all the department.

The report output will be stored as a file “RepEmpSalary4Dept.lst” in the current directory.

# Conclusion

I have completed the exercise at best of my understanding and within the permitted time frame. Although this project has a lots of scope to improve, I hope this shows a glance of my ability to fulfil the role. Should you have any feedback, please let me know.