



ADVENTIST UNIVERSITY OF CENTRAL AFRICA

Exercise: Employment System Design 2024

Course: Database Development with PL/SQL INSY 8311

Instructor: Eric Maniraguha | ericmaniraguha2024@gmail.com | [LinkedIn Profile](#)

Assignment Date:

I. Objective:

Design and implement an employment system using Oracle SQL that manages employee information, including their roles, types, projects, and associated organizations.

II. Problem Statement:

You are required to create an employment system using Oracle SQL. The system will include multiple tables related to employees and their associated data. Your task is to create the database schema, insert data, perform various SQL operations, and optimize queries using indexing and views.

III. Tasks:

Create the database schema:

Design and create the following tables in Oracle SQL

1. **employee_type**
 - **Purpose:** Categorizes employees (e.g., permanent, contract).
 - **Student Task:** Assign Employee_Type_ID when entering employee data.
2. **address_type**
 - **Purpose:** Defines types of addresses (e.g., home, office).
 - **Student Task:** Determine address types for employee records.
3. **organization**
 - **Purpose:** Stores details about organizations (contact info, location).
 - **Student Task:** Populate with organization data; link employees via Organization_ID.
4. **person**
 - **Purpose:** Holds personal details (names, contact info).
 - **Student Task:** Insert personal information; use Person_ID to associate with employees.
5. **employee_role**
 - **Purpose:** Defines employee roles within the organization.
 - **Student Task:** Ensure valid Role_ID is assigned to employees.
6. **employee**

- **Purpose:** Main table for employee details, including salary, bonus, hire date, and location.
- **Student Task:** Create employee entries with foreign keys (Person_ID, Employee_Type_ID, Role_ID, Organization_ID). Include salary, bonus, hire date, and location data.

7. project

- **Purpose:** Stores project information related to the organization.
- **Student Task:** Manage employee assignments to projects using Organization_ID.

IV. How to Deal with These Tables

- **Database Design:** Understand relationships (one-to-many, many-to-many) to design queries and ensure data integrity.
- **Data Entry:** Ensure foreign keys are valid before entering data. Verify that personal details, employee types, and roles exist

V. SQL Queries:

- Use INSERT to add records.
- Use SELECT to retrieve data (e.g., employees in an organization).
- Use JOIN to combine data across tables.

Data Management: Maintain data consistency and integrity by checking relationships (e.g., no invalid references).

Practical Application: Use this schema for projects involving employee management. Create realistic entries and practice SQL queries to analyze employee distribution or project assignments.

VI. Documentation:

- Write a README.md file explaining your design choices, table relationships, and instructions on how to execute the scripts.

VII. Additional Guidelines:

- Ensure that the employee table includes a bonus column that can hold null values.
- Use IDENTITY columns for primary keys where applicable to simplify the generation of unique identifiers.
- Use descriptive comments in your SQL scripts to explain the purpose of each command.
- Test all scripts and ensure they run without errors.

VIII. Submission:

- Submit all SQL scripts and the README.md file.
- Ensure that your code is well-documented and follows best practices for SQL development.

Good Luck!