

Database Application Development

Assignment

Part 2

Objective:

In this assignment, you create a simple HR application using the C++ programming language and Oracle. This assignment helps students learn a basic understanding of application development using C++ programming and a Oracle database.

Submission:

Your submission will be a single text-based .cpp file including your C++ program for the Database Application assignment.

AS_P2_ID.cpp

Your submission needs to be commented.

Instruction:

In this assignment, we complete the application from the first part to insert, update, and delete the employees' information.

Note: For each query in your assignment, make sure you handle the errors and display the proper message including the error code and the error message.

```
try{
    ...
}
catch (SQLException& sqlExcp) {
    cout << sqlExcp.getErrorCode() << ": " << sqlExcp.getMessage();
}
```

You will implement the following functions:

void insertEmployee(struct Employee *emp);

This function receives a pointer to a variable of type Employee. This function gets the employee's information from the user and stores the values in the corresponding members of the structure pointed by the *emp* pointer.

See the following example:

```
***** HR Menu *****
1) Find Employee
2) Employees Report
3) Add Employee
4) Update Employee
5) Remove Employee
0) Exit
Enter an option (0-5): 3

----- New Employee Information -----
Employee Number: 1099
Last Name: Adam
First Name: Sarah
Extension: x2222
Email: s@email.com
Office Code: 1
Manager ID: 1002
Job Title: Cashier
```

The new employee is added successfully.

void insertEmployee(Connection *conn, struct Employee emp);

This function receives an OCCI pointer (a reference variable to an Oracle database) and a variable of type Employee and inserts the given employee information stored in the parameter ***emp*** to the employees table.

Before you call the function ***insertEmployee()***, first call the function ***getEmployee()*** to get the employee's information.

In the function ***insertEmployee()***, call ***findEmployee()*** to see whether the employee number of the given employee exists. If an employee with the same employee number exists display a proper message.

Otherwise, insert the employee information into the employee table and display a proper message.

For simplicity, for new employees the office code is 1 and the manager (reportsto) is 1002. When asking employee's info, do not ask the user to enter the office code and the manager ID. Just display these values. The values in red are the values entered by the user.

After executing the statement make sure you execute commit to make all changes permanent and then terminate the statement.

```
conn->commit();
conn->terminateStatement(stmt);
```

See the following example:

If the employee number does not exist and the employee info inserted successfully:

***** HR Menu *****

- 1) Find Employee
- 2) Employees Report
- 3) Add Employee
- 4) Update Employee
- 5) Remove Employee
- 0) Exit

Enter an option (0-5): 3

----- New Employee Information -----

Employee Number: 1099

Last Name: Adam

First Name: Sarah

Extension: x2222

Email: s@email.com

Office Code: 1

Manager ID: 1002

Job Title: Cashier

The new employee is added successfully.

If the employee number exists:

***** HR Menu *****

- 1) Find Employee
- 2) Employees Report
- 3) Add Employee
- 4) Update Employee
- 5) Remove Employee
- 0) Exit

Enter an option (0-5): 3

----- New Employee Information -----

Employee Number: 1088

Last Name: Adam

First Name: Sarah

Extension: x2222

Email: s@email.com

Office Code: 1

Manager ID: 1002

Job Title: Cashier

An employee with the same employee number exists.

*void updateEmployee(Connection *conn, int employeeNumber);*

Before you call **updateEmployee()**, get the employee number from the user.

This function receives an OCI pointer (a reference variable to an Oracle database) and an integer number as the employee number and updates the phone extension for the given employee. In function **updateEmployee()**, call function **findEmployee()** to see if the employee exists in table employees. If employee does exist, display the employee's last name and first name and then ask the user to enter the new phone extension. Store the new extension in table employees for the given employee number.

After executing the statement make sure you execute commit to make all changes permanent and then terminate the statement.

```
conn->commit();
conn->terminateStatement(stmt);
```

If the employee with the given ID exists:

```
***** HR Menu *****
1) Find Employee
2) Employees Report
3) Add Employee
4) Update Employee
5) Remove Employee
0) Exit
Enter an option (0-5): 4
Employee Number: 1099
Last Name: Adam
First Name: Sarah
Extension: x6666
The employee's extension is updated successfully.
```

If the employee with the given ID does not exist:

```
***** HR Menu *****
1) Find Employee
2) Employees Report
3) Add Employee
4) Update Employee
5) Remove Employee
0) Exit
Enter an option (0-5): 4
Employee Number: 1000
The employee with ID 1000 does not exist.
```

void deleteEmployee(Connection *conn, int employeeNumber);

Before you call **deleteEmployee()**, get the employee number from the user.

This function receives an OCCI pointer (a reference variable to an Oracle database) and an integer number as the employee number and deletes a row with the given employee number from table employees. In function **deleteEmployee()**, call function **findEmployee()** to see if the employee with the given employee number exists. If the employee does not exist display a proper message.

If the employee exists, delete the row from table employees and display a proper message.

After executing the statement make sure you execute commit to make all changes permanent and then terminate the statement.

```
conn->commit();
conn->terminateStatement(stmt);
```

See the following example:

If the employee with the given ID exists:

```
***** HR Menu *****
1) Find Employee
2) Employees Report
3) Add Employee
4) Update Employee
5) Remove Employee
0) Exit
Enter an option (0-5): 5
Employee Number: 1099
The employee with ID 1099 is deleted successfully.
```

If the employee with the given ID does not exist:

```
***** HR Menu *****
1) Find Employee
2) Employees Report
3) Add Employee
4) Update Employee
5) Remove Employee
0) Exit
Enter an option (0-5): 5
Employee Number: 1000
The employee with ID 1000 does not exist.
```