

# Database Application Development

## Project/Assignment

### FINAL SUBMISSION (Part 2)

#### Objective:

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

#### Submission:

*Your submission will be a single text-based .cpp file including your C++ program for the Database Application assignment. The file must include a comment header of student name and ID number.*

AS\_Final.txt

Your submission needs to be commented.

Note: you can submit combine code of part1 and part2 as final.txt also

#### Instruction:

In this assignment, we complete the application from the first part1 (milestone1) to insert, update, and delete the employee information.

You need to implement the following functions:

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

This function receives a connection pointer and a structure of type Employee and inserts the given employee information stored in the parameter ***emp*** to employee table.

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

“An employee with the same employee number exists.” and return to the menu.

Otherwise, insert the employee information into the employee table and display the following message:

“The new employee is added successfully.”

**Note:** For simplicity, assume that the office code of the new employees is 1 and the manager id (reportsTo) is 1002 by default.

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

This function receives a connection pointer 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, ask the user to enter the new phone extension. Store the new extension in table employees for the given employee number.

***void deleteEmployee(\*conn, int employeeNumber);***

This function receives a connection pointer 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:

“The employee does not exist.”

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

“The employee is deleted.”

**Function *main()***

From the menu in the first part of the assignment, complete options 3 to 5.

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

**Add an Employee**

If the user chooses option 3, prompt the user to enter the new employee information and store them into a variable of type Employee structure. Then, call function ***insertEmployee()*** to insert the new employee information in table employees.

Employee Number: 1818  
Last Name: Adam  
First Name: Sarah  
Email: sadam@email.com  
extension: x4411

**Job Title: Sales Rep**

**NOTE:** You do not need to ask the user to enter values for the members *reportsTo* and *phone*.

### **Update an Employee**

If the user chooses option 4, ask the user to enter the employee number:

Employee Number: **1216**

Then, call function ***updateEmployee()*** to update the phone extension for the row with the employee number **1216**. In this function, the user is asked to enter the new extension:

New Extension: **x2111**

The extension column of the row with the employee number 1216 will be updated with the new value x2111.

### **Delete an Employee**

If the user chooses option 5, ask the user to enter the employee number:

Employee Number: **1818**

Then, call function ***deleteEmployee()*** to remove the employee from table employees.

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

`Error_code` is a number returned if the query execution is not successful.