Final Project Report

Database Management –Spring 2022

May 15th, 2022

Ankita Amarnath Kashyap

Student ID: 02010214

Table of Contents

1. Executive Summary 1
2. Project Description 2
3. Database Design: ER Diagram 3
   3.1. Employee 3
   3.2. DevelopmentCenter 4
   3.3. Unit Table 4
   3.4. Project Table 5
   3.5. Activity Table 5
   3.6. Relations: 6
4. Normalization 7
5. Stored Procedures 10
6. Triggers 11
7. Database Implementation 12
8. Appendices 13

1. Executive Summary

The primary objective of this project is to establish a database management system within the organization which would allow the employers/managers to keep a track of the company's resources.

The main function of this project is to manage the details of Infosys employees and Projects with Infosys. This project tracks all the relevant information of the employees in the organization, the unit (or department) they belong to, which employee works from what location and, is the person on bench (in case he/she is not related to any project) or are they in any project and, their participation in activities at any office campus. It shows all the details for project such as Billable Amount, it's Profit, Start date, End Date, etc. This could help in prioritizing and focusing work on required project, as per organization's needs.

This could prove to be an efficient method for the management to record, analyze and process all the information they need on their Employees, Projects, Units (Departments), Development Centers (Offices), Activities and their overall productivity at the organization.

2. Project Description

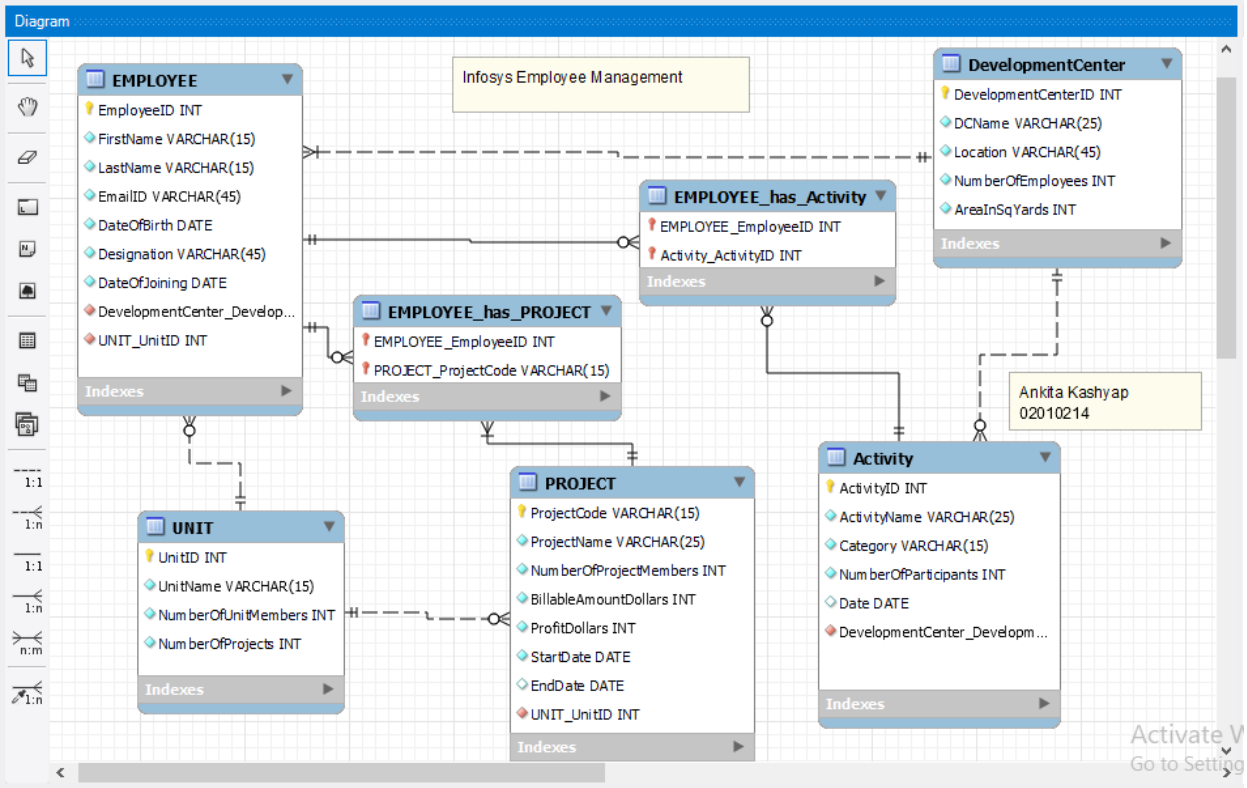
Infosys is a service-based company. It serves a very big range of clients since the year 1981. Currently they have more than 1,626 different clients that they are working with. To serve such a huge number, they need a humongous number of employees.

To store the data related to this large number of employees and organization, we are using database for Infosys. In this project, we are focusing on data related to employees, projects that Infosys has and its details, Units (departments) that are functional, Development Center details (Infosys Offices), extra activities that employees can take part into.

Business requirements:

- To get all the Employee related data which will be linked to his/her location, unit, employee's productivity in terms of working in a project or not and, also their engagements in extra activities can be tracked from this database
- Details related to Clients/Projects that are with Infosys can be easily tracked with this project. Details like what is the profit Infosys earns, what is the bill amount it charges its clients, when did the project started, till when should Infosys server a particular client, and also number of employees associated with this client
- Extra activities like Corporate social responsibility activities, entertainment activities and even tournaments that Infosys arranges for its workers can be tracked by this database
- This project's end users will be Managers, HR and other Top executives at Infosys
- This project aims to bring all the scattered data together and help the management ease their process of tracking data related to everyday use at their organization

3. Database Design: ER Diagram



3.1. Employee

This table holds details of every employee like employee ID, name, Date of birth, etc.

Column Name	Data Type	Keys/Constraints	Remarks
EmployeeID	INT	Primary Key, Not Null Auto_Increment	Unique identity given to each employee
FirstName	VARCHAR(15)	UNIQUE(FirstName, LastName), Not Null	First name of the employee. It is a unique key when combined with Last Name.
LastName	VARCHAR(15)	UNIQUE(FirstName, LastName), Not Null	Last name of the employee. It is a unique key when combined with First Name.
EmailID	VARCHAR(45)	Not Null	Email Address of the employee
DateOfBirth	DATE	CHECK (DateOfBirth < DateOfJoining) this is implemented with TRIGGER, Not Null	Date of Birth (DOB) of the employee. It should always be less than Date of joining

Designation	VARCHAR(45)	Not Null	Designation/Post of employee
DateOfJoining	DATE	CHECK (DateOfBirth < DateOfJoining) this is implemented with TRIGGER, Not Null	Date when the employee joined Infosys. This has to be greater than DOB
DevelopmentCenterID	INT	Foreign Key, Not Null, On Update Cascade, On Delete Cascade	Reference from DevelopmentCenter table
UnitID	INT	Foreign Key, Not Null, On Update Cascade, On Delete Cascade	Reference from Unit table

3.2. DevelopmentCenter

Infosys calls its offices as Development Centers (DC). This table holds details like name, location, area of every office

Column Name	Data Type	Keys/Constraints	Remarks
DevelopmentCenterID	INT	Primary Key, Not Null Auto_Increment	Unique identity given to each DC
DCName	VARCHAR(25)	Unique Key, Not Null	Unique Name given to each DC
Location	VARCHAR(45)	Not Null	City of particular DC
NumberOfEmployees	INT	CHECK (NumberOfEmployees > 0) , Not Null	This field should not be zero as there has to be minimum 1 employee for a DC to be functional
AreaInSqYards	INT	Not Null	Area of DC in sq. yards

3.3. Unit Table

Infosys refers to its departments as Units. This table holds the values of each units and its details like its ID, Name, number of projects and employees linked to each unit.

Column Name	Data Type	Keys/Constraints	Remarks
UnitID	INT	Primary Key, Not Null Auto_Increment	Unique identity given to each unit
UnitName	VARCHAR(15)	Unique Key, Not Null	Unique Name given to each unit
NumberOfUnitMembers	INT	Not Null	Number of employees in a unit
NumberOfProjects	INT	Not Null	Number of projects linked to a unit

3.4. Project Table

This table holds details of client projects. It has values like Project name, ID, Profit earned, Billable amount, Number of employees, etc.

Column Name	Data Type	Keys/Constraints	Remarks
ProjectCode	VARCHAR(15)	Primary Key, Not Null	Unique code assigned to each project. This is not auto incremented
ProjectName	VARCHAR(25)	Unique Key, Not Null	Unique Name given to each Project
NumberOfProjectMembers	INT	Not Null	Number of employees in a project
BillableAmountDollars	INT	Not Null	This the project cost quoted to the client
ProfitDollars	INT	Not Null	Profit earned after considering all the expenses
StartDate	DATE	Not Null	Date when project started with Infosys
EndDate	DATE	Null	This value can be NULL as there are chances that no end date is decided for a project
UnitID	INT	Foreign Key, Not Null, On Update Cascade, On Delete Cascade	Reference from Unit table

3.5. Activity Table

This table holds all the extra activities like tournaments, CSR activities, games and other fun events that take place in any DC.

Column Name	Data Type	Keys/Constraints	Remarks
ActivityID	INT	Primary Key, Not Null Auto_Increment	Unique identity given to each activity
ActivityName	VARCHAR(25)	Unique Key, Not Null	Unique Name given to each activity
Category	VARCHAR(15)	Not Null	This field specifies the type of activity such as tournaments, CSR activities, games and other fun events

NumberOfParticipants	INT	Not Null	Number of employees participating in any activity
Date	DATE	Not Null	Date on which that event is scheduled
DevelopmentCenterID	INT	Foreign Key, Not Null, On Update Cascade, On Delete Cascade	Reference from DevelopmentCenter table

3.6. Relations:

1. DevelopmentCenter: Employee
   An Employee will have one and only one DC but a DC can have one or many employees
   - Maximum: 1:N
   - Minimum: 1:1
2. DevelopmentCenter: Activity
   A DC can host zero or multiple Activities but a particular activity will be hosted at one and only one DC
   - Maximum: 1:N
   - Minimum: 1:0
3. Unit: Employee
   A unit can have zero or many employees but an employee will be linked to and and only one Unit
   - Maximum: 1:N
   - Minimum: 1:0
4. Project: Employee
   A project can have many employees but it will at least have one employee. An employee can belong to either no project or can even have multiple projects
   - Maximum: N: M
   - Minimum: 1:1 for project
   - Minimum: 1:0 for employee
5. Unit: Project
   A unit can have no projects at all or can even have multiple projects. But a project can belong to one and only one unit
   - Maximum: 1:N
   - Minimum: 1:0
6. Employee: Activity
   An employee can take part in none of the activities or can even have multiple activities. Similarly, an activity can have zero to many participants
   - Maximum: N: M
   - Minimum: 1:0

4. Normalization

3.1. Employee table

(EmployeeID, FirstName, LastName, EmailID, DateOfBirth, Designation, DateOfJoining)

STEP 1: Identify every functional dependency

- EmployeeID -> (FirstName, LastName, EmailID, DateOfBirth, Designation, DateOfJoining)
- (FirstName, LastName) -> (EmployeeID, EmailID, DateOfBirth, Designation, DateOfJoining)

STEP 2: Identify every candidate key

- Candidate Keys: EmployeeID, (FirstName, LastName)

STEP 3: This is in BCNF

Result:

- Employee (EmployeeID, FirstName, LastName, EmailID, DateOfBirth, Designation, DateOfJoining, *DevelopmentCenterID, UnitID*)
  Where Employee. DevelopmentCenterID MUST EXIST in DevelopmentCenter.DevelopmentCenterID
  Where Employee.UnitID MUST EXIST in Unit.UnitID
- Primary Key: EmployeeID
- Candidate Keys: EmployeeID, (FirstName, LastName)
- Foreign Key:
  i. DevelopmentCenterID is the foreign key referring to DevelopmentCenter.DevelopmentCenterID
  ii. UnitID is the foreign key referring to Unit.UnitID

3.2. DevelopmentCenter

(DevelopmentCenterID, DCName, Location, NumberOfEmployees, AreaInSqYards)

STEP 1: Identify every functional dependency

- DevelopmentCenterID -> (DCName, Location, NumberOfEmployees, AreaInSqYards)
- DCName -> (DevelopmentCenterID, Location, NumberOfEmployees, AreaInSqYards)

STEP 2: Identify every candidate key

- Candidate Keys: DevelopmentCenterID, DCName

STEP 3: This is in BCNF

Result:

- DevelopmentCenter (DevelopmentCenterID, DCName, Location, NumberOfEmployees, AreaInSqYards)
- Primary Key: DevelopmentCenterID
- Candidate Keys: DevelopmentCenterID, DCName

3.3. Unit Table

Unit (UnitID, UnitName, NumberOfUnitMembers, NumberOfProjects)

STEP 1: Identify every functional dependency

- UnitID -> (UnitName, NumberOfUnitMembers, NumberOfProjects)
- UnitName -> (UnitID, NumberOfUnitMembers, NumberOfProjects)

STEP 2: Identify every candidate key

- Candidate Keys: UnitID, UnitName

STEP 3: This is in BCNF

Result:

- Unit (UnitID, UnitName, NumberOfUnitMembers, NumberOfProjects)
- Primary Key: UnitID
- Candidate Keys: UnitID, UnitName

3.4. Project Table

Project (ProjectCode, ProjectName, NumberOfProjectMembers, BillableAmountDollars, ProfitDollars, StartDate, EndDate)

STEP 1: Identify every functional dependency

- ProjectCode -> (ProjectName, NumberOfProjectMembers, BillableAmountDollars, ProfitDollars, StartDate)
- ProjectName -> (ProjectCode, NumberOfProjectMembers, BillableAmountDollars, ProfitDollars, StartDate, EndDate)

STEP 2: Identify every candidate key

- Candidate Keys: ProjectCode, ProjectName

STEP 3: This is in BCNF

Result:

- Project (ProjectCode, ProjectName, NumberOfProjectMembers, BillableAmountDollars, ProfitDollars, StartDate, EndDate, *UnitID*)
- Primary Key: ProjectCode
- Foreign Key: UnitID

3.5. Activity Table

Activity (ActivityID, ActivityName, Category, NumberOfParticipants, Date)

STEP 1: Identify every functional dependency

- ActivityID -> (ActivityName, Category, NumberOfParticipants, Date)
- ActivityName -> (<u>ActivityID</u>, Category, NumberOfParticipants, Date)

STEP 2: Identify every candidate key

- Candidate Keys: ActivityID, ActivityName

STEP 3: This is in BCNF

Result:

- Activity (ActivityID, ActivityName, Category, NumberOfParticipants, Date, *DevelopmentCenterID*)
  Were Activity. DevelopmentCenterID MUST EXIST in DevelopmentCenter.DevelopmentCenterID
- Primary Key: ActivityID
- Foreign Key: DevelopmentCenterID

5. Stored Procedures

Stored Procedures are used to remove redundancy of code. When we require same set of code multiple times, we use stored procedures to avoid repetition of code.

Stored Procedures are of three types:

- Simple Stored Procedures
- Stored Procedures with variables
- Stored Procedures with parameters

For this project, I have used Stored Procedures with variables.

I purpose to use this code was to avoid repeating the same code in future.

Suppose I expand this project in future, I might use Update, Delete or Insert Statements. This will change my result in future. That time I will just have to call my Procedure named 'GetTAEmployees' and I will not have to write Select statement again.

6. Triggers

Triggers are used to prevent the code to run into error during compilation.
If we add constraints, complier will prevent the user to Insert, Update or Delete unwanted data. But this will stop compilation as the entire code will go into error stage.
There are three types of triggers:

- Before
- Instead of
- After

And they can be used in Insert, Update or Delete statements.

I have used Trigger Before Insert on Employee table to avoid such abrupt failure of code.

Result Grid | Filter Rows: | Edit: | Export/Import: | Wrap Cell Content:

EmployeeID	FirstName	LastName	EmailID	DateOfBirth	Designation	DateOfJoining	DevelopmentCenterID	UnitID
1001004	Kanchan	Maurya	kanchan.maurya@infosys.com	1995-02-28	Systems Engineer	2017-11-11	2	1007
1001005	Lakshmi	Ashok	lakshmi.ashok@infosys.com	1995-02-28	Technology Analyst	2017-11-11	2	1003
1001006	Neha	Bhushan	neha.bhushan@infosys.com	1997-01-23	Systems Engineer	2020-11-11	2	1001
1001007	Vidhi	Bhushan	ERROR!! CHANGE DATE OF BIRTH OR JOINING	1997-01-23	Systems Engineer	1995-11-11	2	1002
NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

employee 1

7. Database Implementation

A. In this project I have created entity relationship diagram (ERD) first, then normalized the equations and then I manually wrote the code to create Database and tables. Then I used INSERT statements to add data in the tables
I have even used Stored Procedures with variables in my code

B. Queries

1. **Using Employee table, show EmployeeID, First Name, Last Name, EmailID of all the employees with designation 'Technology Analyst'**
    - This query helps to list employees based on their designation
    - I have used **Stored Procedures with variables** to write this query
2. **Using Project table, list all the projects in decreasing order of their profit amounts. Also list their start and end dates**
    - This query helps Managers to track the profit earned by all the projects and to decide which project is most profitable and which one needs to be focused on, so that more profits can be earned
    - I have used Select statement, Order By (DESC) to write this query
3. **Using Project table, check the total Billable Amount in Dollars from all the projects**
    - This Query helps Managers to know the total Billable Amount from all the projects
    - I have used Select statement, Aggregate function SUM and aliasing for new column name to write this query
4. **List all the activities taking place at Mysore DC**
    - This Query helps keep a track of activities taking place at different Locations
    - I have used Nested Select statement (Sub-query) to write this query
5. **List all the employees taking part in Activity named CSR Mumbai. Also list their following details EmployeeID, FirstName, LastName, EmailID, DateOfBirth, Designation**
    - This Query helps Managers to know which employees are taking part in a particular Activity
    - I have used Multiple Nested Select statement (Sub-query) to write this query
6. **Write a query to show the project name and project code for which Employee 'Ankita Kashyap' works**
    - This Query helps find out the project details for any particular employee
    - I have used Multiple Nested Select statement (Sub-query) to write this query
7. **List all the employees and all their details who belong to Unit named 'Java'**
    - This Query helps get data of all the employees belonging to a specific Unit
    - I have used Nested Select statement (Sub-query) to write this query

Note: Results are attached in Appendices

8. Appendices

7.1 Results for Queries

1. **Using Employee table, show EmployeeID, First Name, Last Name, EmailID of all the employees with designation 'Technology Analyst'**

```
/* Stored Procedures with variables */
DROP PROCEDURE IF EXISTS GetTAEmployees;
DELIMITER //
CREATE PROCEDURE GetTAEmployees()
BEGIN
DECLARE post VARCHAR(45);
SET post = 'Technology Analyst';
SELECT EmployeeID, FirstName, LastName, EmailID FROM Employee
WHERE Designation = post;
END //
DELIMITER ;
CALL GetTAEmployees();
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

EmployeeID	FirstName	LastName	EmailID
1001002	Megha	Biradar	megha.biradar@infosys.com
1001003	Subodh	Dongre	subodh.dongre@infosys.com
1001005	Lakshmi	Ashok	lakshmi.ashok@infosys.com

2. **Using Project table, list all the projects in decreasing order of their profit amounts. Also list their start and end dates**

```
/* 2. Using Project table, list all the projects in decreasing order of their profit amounts. Also list their start and end dates */
/* This query helps Managers to track the profit earned by all the projects and to decide
which project is most profitable and which one needs to be focused on, so that more profits can be earned */
SELECT ProjectCode, ProjectName, ProfitDollars, StartDate, EndDate
  FROM Project
ORDER BY ProfitDollars DESC;
```

Result Grid | Filter Rows: | Edit: | Export/Import: | Wrap Cell Content:

ProjectCode	ProjectName	ProfitDollars	StartDate	EndDate
CC1	GHI	50000000	2020-06-11	NULL
Mck1	ABC	40000000	2010-06-11	NULL
Nv1	DEF	10000000	2010-06-11	2025-06-11
Lz1	JKL	1000000	2020-06-11	NULL
MNO1	MNO	1000000	2020-06-11	NULL
PQR1	PQR	1000000	2020-06-11	NULL
NULL	NULL	NULL	NULL	NULL

3. **Using Project table, check the total Billable Amount in Dollars from all the projects**

```
/* 3. Using Project table, check the total Billable Amount in Dollars from all the projects */
/* This Query helps Managers to know the total Billable Amount from all the projects */
SELECT SUM(BillableAmountDollars) AS TotalBillableAmount
FROM Project;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

TotalBillableAmount
176000000

4. **List all the activities taking place at Mysore DC**

```
/* 4. List all the activities taking place at Mysore DC */
/* This Query helps keep a track of activities taking place at different Locations */
SELECT ActivityID, ActivityName
FROM Activity
WHERE DevelopmentCenterID IN
                        (SELECT DevelopmentCenterID
                        FROM DevelopmentCenter
                        WHERE DCName = 'Mysore DC');
```

Result Grid | Filter Rows: | Edit: | Export/Import: | Wrap Cell Content:

ActivityID	ActivityName
2000	Basketball
2001	Bowling
NULL	NULL

5. **List all the employees taking part in Activity named CSR Mumbai. Also list their following details EmployeeID, FirstName, LastName, EmailID, DateOfBirth, Designation**

```
SELECT EmployeeID, FirstName, LastName, EmailID, DateOfBirth, Designation
FROM Employee
WHERE EmployeeID IN
                (SELECT EmployeeID
                FROM Employee_has_Activity
                WHERE ActivityID IN
                                (SELECT ActivityID
                                FROM Activity
                                WHERE ActivityName = 'CSR Mumbai'));
```

Result Grid | Filter Rows: | Edit: | Export/Import: | Wrap Cell Content:

EmployeeID	FirstName	LastName	EmailID	DateOfBirth	Designation
1001000	Ankita	Kashyap	ankita.kashyap@infosys.com	1996-03-25	Senior Systems Engineer
1001002	Megha	Biradar	megha.biradar@infosys.com	1992-04-05	Technology Analyst
NULL	NULL	NULL	NULL	NULL	NULL

6. **Write a query to show the project name and project code for which Employee 'Ankita Kashyap' works**

```
/* 6. Write a query to show the project name and project code for which Employee 'Ankita Kashyap' works */
/* This Query helps find out the project details for any particular employee */
SELECT ProjectName, ProjectCode FROM Project
   WHERE ProjectCode IN
                (SELECT ProjectCode FROM Employee_has_Project
                    WHERE EmployeeID IN
                                        (SELECT EmployeeID FROM Employee
                                            WHERE FirstName = 'Ankita' AND LastName  = 'Kashyap'));
```

Result Grid | Filter Rows: | Edit: | Export/Import: | Wrap Cell Content:

ProjectName	ProjectCode
ABC	Mck1
NULL	NULL

7. **List all the employees and all their details who belong to Unit named 'Java'**

```
/* 7. List all the employees and all their details who belong to Unit named 'Java' */
/* This Query helps get data of all the employees belonging to a specific Unit */
SELECT EmployeeID, FirstName, LastName, EmailID, DateOfBirth,
Designation, DateOfJoining, DevelopmentCenterID, UnitID
FROM Employee WHERE UnitID IN
                        (SELECT UnitID FROM Unit
                        WHERE UnitName = 'Java');
```

Result Grid | Filter Rows: | Edit: | Export/Import: | Wrap Cell Content:

EmployeeID	FirstName	LastName	EmailID	DateOfBirth	Designation	DateOfJoining	DevelopmentCenterID	UnitID
1001002	Megha	Biradar	megha.biradar@infosys.com	1992-04-05	Technology Analyst	2020-06-11	5	1001
1001006	Neha	Bhushan	neha.bhushan@infosys.com	1997-01-23	Systems Engineer	2020-11-11	2	1001
NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL