

Assignment 4/Final Project Documentation

Contents

1. Introduction
2. Tables
3. ER Diagram
4. UML Diagram
5. Creating Tables
6. Inserting and Updating Data to Tables
7. Functions, Procedures, Triggers
8. Users and view levels
9. Database Security
10. Back-up
11. Replication and Hosting
12. Author's note

Introduction

Designing a SQL database for keeping records of current students and alumni of a university to keep track of their employment opportunities. The database holds data belonging to the current students, alumni and the university staff for jobs, and information about jobs (in detail) – along with recruiters and the company.

Tables

Table 1 – Jobs

Field	Description
ID	Primary Key – Identifier for each job
Title	Title of job
Organization	Name of company providing the job
Department	Field (CS, Biology, etc) of job
Position_Type	Type of employment (full time, part time, etc)
Location	Location
App_Deadline	Deadline to apply

Table 2 – Job_Description

Fields	Description
JobID	Foreign Key – Identifier for each job
Job_Desc	Description of job
Salary	Salary for the job
Department	Field (CS, Biology, etc) of job
No_of_Positions	Number of positions available for the job
Location	Location
Position_Type	Type of employment (full time, part time, etc)
Company_ID	Foreign key - ID of company providing job

Table 3 – Job_Requirements

Fields	Description
Job_ID	Foreign Key – Identifier for each job
Required_Docs	Required documents for job
Opportunity_Requirements	Eligibility requirements for job
App_Deadline	Deadline to apply

Table 4 – Recruiters

Fields	Description
ID	Primary key – Identifier for each job recruiter
Name	Name of recruiter
BranchID	Foreign key – Branch identifier of recruiter's job location
Email	Email contact of recruiter
Phone_Num	Phone number of recruiter

Table 5 – Univ_Staff

Fields	Description
Staff_ID	Primary key – Identifier for each university employee
Name	Name of university employee
Current_Pos	Current working position of employee
Department	Field of job
Salary	Salary of employee
Email	Email contact of employee
Phone_Num	Phone number of employee
City	Location (Current)
Province_State	Location (Current)
Country	Location (Current)

Table 6 – Company_info

Fields	Description
Company_ID	Primary key – Identifier for each company providing jobs
Name	Name of company
Department	Primary field of work of company
Website	Company's website

Table 7 – Company_Branch

Fields	Description
Branch_ID	Primary key – Branch identifier of a company
City	Location
Province_State	Location
Country	Location
CompanyID	Foreign key - Identifier for each company providing jobs

Table 8 – Alumni

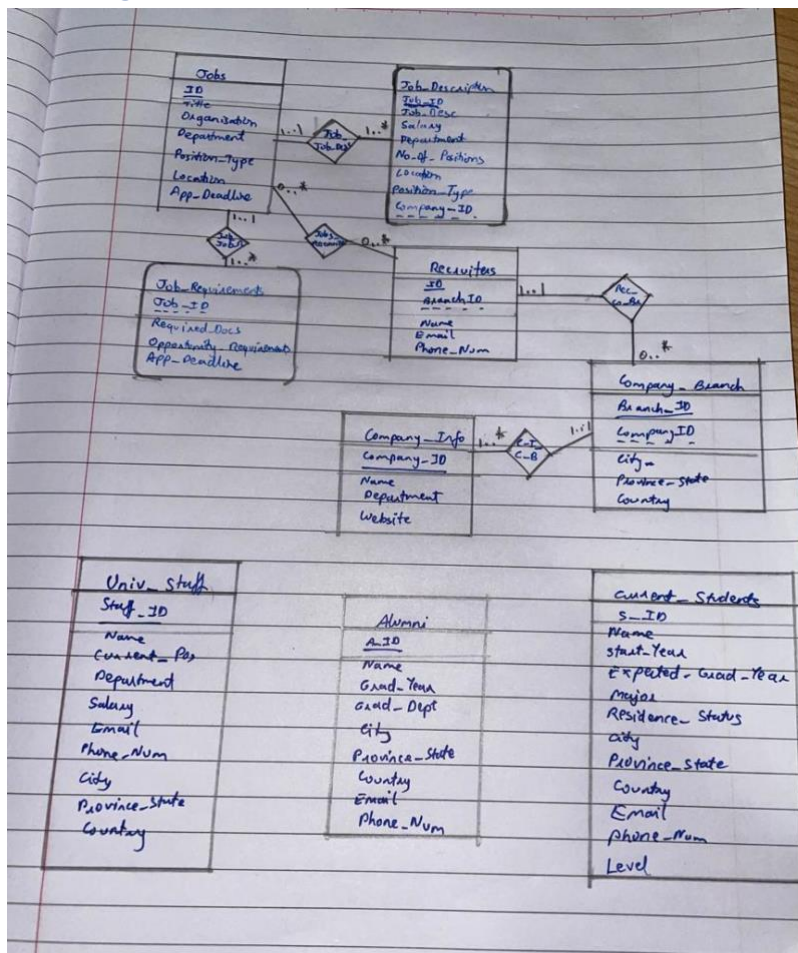
Fields	Description
A_ID	Primary key – Identifier for university alumni
Name	Name of the alumni
Grad_Year	Graduation year of the alumni
Grad_Dept	Major at graduation
City	Location (Current)
Province_State	Location (Current)
Country	Location (Current)
Email	Email contact of alumni
Phone_Num	Phone number of alumni

Table 9 – Current_Students

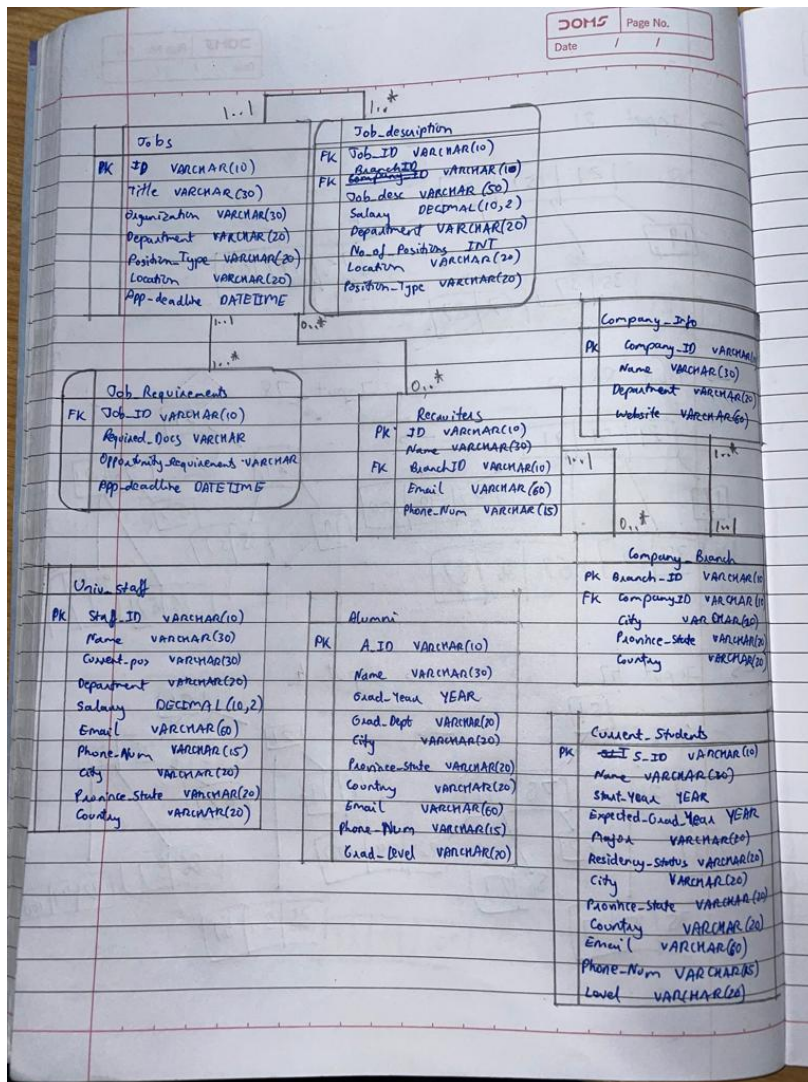
Fields	Description
S_ID	Primary key – Identifier for university student
Name	Name of the student

Start_Year	Start year at university
Expected_Grad_Year	Expected graduation year of student
Major	Field of study
Residence_Status	Domestic/International
City	Location (Current)
Province_State	Location (Current)
Country	Location (Current)
Email	Email contact of student
Phone_Num	Phone number of student

ER Diagram



UML Diagram



Creating Tables

CREATE TABLE Jobs (

ID VARCHAR (10),

Organisation VARCHAR (30),

Department VARCHAR (20),

Position_Type VARCHAR (20),

Location VARCHAR (20),

App_Deadline DATETIME,

PRIMARY KEY (ID)

);

CREATE TABLE Job_Description (

```
Job_ID VARCHAR (10),
Job_Desc VARCHAR (50),
Salary DECIMAL (10,2),
Department VARCHAR (20),
No_of_Postions INT,
Location VARCHAR (20),
Position_Type VARCHAR (20),
Company_ID VARCHAR (10),
FOREIGN KEY (Job_ID) REFERENCES Jobs (ID),
FOREIGN KEY (Company_ID) REFERENCES Company_Info (Company_ID)
);
```

```
CREATE TABLE Job_Requirements (
Job_ID VARCHAR (10),
Required_Docs VARCHAR,
Opportunity_Requirements VARCHAR,
App_Deadline DATETIME,
FOREIGN KEY (Job_ID) REFERENCES Jobs (ID)
);
```

```
CREATE TABLE Recruiters (
ID VARCHAR (10),
Name VARCHAR (30),
BranchID VARCHAR (10),
Email VARCHAR (60),
Phone_Num VARCHAR (15),
PRIMARY KEY (ID),
FOREIGN KEY (BranchId) REFERENCES Company_Branch (ID)
);
```

```
CREATE TABLE Univ_Staff (
```

```
Staff_ID VARCHAR (10),
Name VARCHAR (30),
Current_pos VARCHAR (30),
Department VARCHAR (20),
Salary DECIMAL (10,2)
Email VARCHAR (60),
Phone_Num VARCHAR (15),
City VARCHAR (20),
Province_State VARCHAR (20)
Country VARCHAR (20)
PRIMARY KEY (Staff_ID)
);
```

```
CREATE TABLE Company_Info (
Company_ID VARCHAR (10),
Name VARCHAR (30),
Department VARCHAR (20),
Website VARCHAR (60),
PRIMARY KEY (COMPANY_ID)
);
```

```
CREATE TABLE Company_Branch (
Branch_ID VARCHAR (10),
City VARCHAR (20),
Province_State VARCHAR (20),
Country VARCHAR (20),
Company_ID VARCHAR (20),
PRIMARY KEY (Branch_ID),
FOREIGN KEY (Company_ID) REFERENCES Company_Info (Company_ID)
);
```

```
CREATE TABLE Alumni (  
A_ID VARCHAR (10),  
NAME VARCHAR (30),  
Grad_year YEAR,  
Grad_Dept VARCHAR (20),  
City VARCHAR (20),  
Province_State VARCHAR (20),  
Country VARCHAR (20),  
Email VARCHAR (60),  
Phone_Num VARCHAR (15),  
PRIMARY KEY (A_ID),  
);
```

```
CREATE TABLE Current_Students (  
S_ID VARCHAR (10),  
Name VARCHAR (30),  
Start_Year YEAR  
Expected_Grad_Year YEAR  
Major VARCHAR (20)  
Residency_Status VARCHAR (20),  
City VARCHAR (20),  
Province_State VARCHAR (20),  
Country VARCHAR (20),  
Email VARCHAR (60),  
Phone_Num VARCHAR (15),  
PRIMARY KEY (S_ID)  
);
```

Inserting and Updating Data to Tables

```
INSERT INTO `Univ_Staff` (`Staff_ID`, `Name`, `Current_pos`, `Department`, `Salary`, `Email`,  
`Phone_Num`, `City`, `State`, `Country`) VALUES
```



```
('P101','John Hush','Professor','Computer Science','120000',  
'johnbush@gmail.com','6479081234','London','Ontario','Canada'),  
(  
'P102','Swae Lee','Accountant','Finance','200000',  
'swaelee@gmail.com','6489788234','Oshawa','Ontario','Canada');
```

```
INSERT INTO `Alumni` (`A_ID`,`NAME`,`Grad_year`,`Grad_Dept`,`City`,`State`,`Country`,`Email`,  
`Phone_Num`,`Level`) VALUES
```

```
('10001','Scott Travis','2018','Physics','Peterborough','Ontario','Canada','scotttravis@gmail.com',  
'6478971562','Graduate'),
```

```
('10002','Charles James','2017','Chemistry','Toronto','Ontario','Canada',  
'charlesjames@gmail.com','6489771582','Undergraduate');
```

```
INSERT INTO `Current_Students` (`S_ID`,`Name`,`Start_Year`,`Expected_Grad_Year`,`Major`,  
`Residence_Status`,`City`,`State`,`Country`,`Email`,`Phone_Num`,`Level`) VALUES
```

```
('100112','Christian Roland','2021','2025','Business','Domestic','Peterborough','Ontario','Canada',  
'christian@gmail.com','6541239876','Undergraduate'),
```

```
('101526','Leon Kennedy','2020','2024','Nursing','International','Mumbai','Maharashtra','India',  
'leon@hotmail.com','6541233876','Undergraduate');
```

```
INSERT INTO Jobs (ID, Title, Organisation, Department, Position_Type, Location, App_deadline)  
VALUES
```

```
('101','Data Analyst','TCS','Security','Full-Time','Toronto','2022/09/22'),
```

```
('102','Financial Advisor','CIBC','Admin','Part-Time','Peterborough','2023/02/07');
```

```
INSERT INTO Job_Description (Job_ID, Salary, Department, No_Of_Position, Location, Position_Type,  
Company_ID)
```

```
VALUES
```

```
('101','100000','Security','5','Toronto','Full-Time','TCST'),
```

```
('102','25000','Administration','35','Peterborough','Part-Time','CIBCP');
```

```
INSERT INTO Job_Requirements (Job_ID, Required_Docs, Opportunity_Requirements, App_deadline)
```

```
VALUES
```

```
('101','CV, Resume, Cover letter','MSc Data-Analytics','2022/09/22'),
```

```
('102','CV, Resume, Cover letter','MBA','2023/02/07').
```

```
INSERT INTO Job_Requirements (ID, Name, BranchID, Email, Phone_Num)
VALUES
('101', 'Rocky Malcolm', 'TCST001', 'rockymalcolm@gmail.com', '2988769968'),
('102', 'Robert Carlos', 'CIBCP002', 'rcarlos@hotmail.com', '7057481001');
```

```
INSERT INTO Company_Info (Company_ID, Name, Department, Website)
VALUES
('TCST', 'Rocky Malcolm', 'Security', 'TCS.org'),
('CIBCP', 'Robert Carlos', 'Administration', 'CIBCbanking.org');
```

```
INSERT INTO Company_Branch (Branch_ID, City, Province/State, Country, CompanyID)
VALUES
('TCST001', 'Toronto', 'Ontario', 'Canada' 'TCST'),
('CIBCP002', 'Peterborough', 'Ontario', 'Canada' 'TCST');
```

Functions, Procedures, Triggers

Functions:

Enables employers/recruiters to find quickly any people matching job profile (even useful for university administration)

Function 1: To Search for Alumni/University Staff

Arguments: Location (Current), Degree, Graduation_year, etc

Returns list of people matching the criterion

Enables university administration to search for current students efficiently

Function 2: To search for current students

Arguments: Major, Residency_status, Start_year, Expected_grad_year, etc

Returns list of students matching criterion

Procedures:

Procedure 1: Increment salary

Arguments: set of people from a SQL query/just a table, rate to increment by

Description: increments salary by given rate for the given set of people

Triggers:

Trigger 1: Delete from Current_Students and Insert into Alumni

Description: When real date year crosses the expected graduation year for a current student, student instance is deleted from the current_students table and is inserted into alumni table. This change is recorded into an alumni_transfer log table.

Trigger 2: Delete from Jobs

Description: When the application deadline of a job posting passes, the job instance is deleted from the Jobs table, and this deletion is recorded in a job_deadline_passed log table.

Trigger 3: Insert into Job_applied_for (log table)

Description: When a person applies for a job, their ID and the job ID, along with the datetime they applied for the job is inserted into the log table.

Users and view levels

Users:

Database Administrator – Conceptual Level

Security Administrator – Physical Level

University Management – External Level

Other University Staff – End Users

University Alumni – End Users

Current University Students – End Users

Recruiters from companies – End Users

Database Security

Database Administrator – Fully Trusted User

Security Administrator – Fully Trusted User

University Management – Large Scale Read Access

Other University Staff – Regular Database Users

University Alumni – Regular Database Users

Current University Students – Regular Database Users

Recruiters from companies – Regular Database Users

Each user has their own password access to the database. Additionally, Security administrator and DBA have an extra security key to authorize major data transfers, and for other important database related work.

Back-up

The database would need an onsite, offsite and cloud server for being stored.

The database would need updates everyday for loading in any new job postings, or job applications in the database. For the university's reference, the data would need to be up to date. In case of breakdown, the data (especially personal information) would be prioritized for build.

In the event a large amount of data (>30%) would need to be imported to a new storage, security keys from both the Database Administrator and Security Administrator would need to be used for permission. The DBA and SA would have highest access to the database schema. However, for any major data related events, the security key from both DBA and SA would be needed.

Replication and Hosting

We would need a large server (obviously) to store our database. Ideally, the server should be capable of holding a database containing at least a million entries.

In case a university has a lot of money to put into this database, they can just use the available infrastructure of any supercomputer from top500 list. This would account for holding large data and give fast processing.

Author's note

Jobs, Alumni, Students, Staff Tracking (JASS) university database by Krutarth Ghuge for COIS 3400H Assignment 4/Final Project Summer 2022, Trent University.

Collaboration:

Table design: Samaksh Monga and Rahul Makhija

SQL Queries: Prabal Kochhar and Ansh Adgaonkar