BUAN.6320.004
Database Foundations for Business Analytics

Opportunity Pool
Semester Project Group 12

Group Members:

Aman Sahu	Aman.Sahu@UTDallas.edu
Atharva Gupte	axg220265@UTDallas.edu
Chukwunonso Nnodum	Chukwunonso.Nnodum@UTDallas.edu
Jayesh Haryani	Jayesh.Haryani@UTDallas.edu
Keshav Sarraf	kxs230012@utdallas.edu
Malav Preyas Sheth	MalavPreyas.Sheth@UTDallas.edu

Project outline and objective

We propose the development of "Opportunity Pool," a comprehensive job database management system designed to simplify and enhance the job search and recruitment processes for both job seekers and employers. This centralized platform will provide users with access to a wealth of job-related information, enabling them to search, filter, and analyze data according to their specific needs. The system will feature a user-friendly interface to ensure ease of access and understanding.

Key Features

Comprehensive Job Data: The database will centralize job-related data, making it easy for job seekers to filter and sort job listings based on various criteria, simplifying the job-hunting process.

Data Backup and Recovery: Implement regular data backup and recovery procedures to safeguard against data loss or system failures. Database cluster will be implemented.

Data Security: Ensuring the security and confidentiality of user data through database authentication system.

Scalability: Design the system to be scalable, accommodating a growing user base and expanding the database.

Skills and Roles Exploration: Job hunters can explore various skills and roles within the database to gain insights into career development and growth opportunities.

Employer Access: Employers can use the system to access a vast pool of potential employees, streamlining their hiring processes and finding the right candidates efficiently.

Professional Networking: The system will feature an opportunity pool that allows users to build professional networks by connecting with others' profiles, filtered by criteria such as schools and companies.

Job Posting Interface: Employers can post job openings through the user interface, expanding their reach to a large number of job hunters. Job seekers can stay updated with new opportunities.

Intelligent Matching: The database will offer insights into matching relevant candidates with job opportunities, based on skills, experience, and job requirements. These insights can be shared with employers to enhance their hiring decisions.

User-Friendly Interface: A user-friendly and intuitive interface for both job seekers and employers to navigate and interact with the system effortlessly.

Search and Filter Capabilities: Advanced search and filtering options, including location, salary, experience level, and more, to provide users with tailored job search results.

Cost-Efficiency: Ensure the system's cost-effectiveness in terms of maintenance and scalability, making it accessible to a wide range of users and employers.

Timeline
Phase one: Planning and design
Phase two: Collection of data and insertion in the database
Phase three: Development
Phase five: Documentation and deployment

Tenatave Tables and their attributes

Entities (tables)	**Attributes (columns)**
User table	UserId, FName, LName, Age, Sex, Contact, Country, Loc, MostRecentJobTitle, EmployementType, MostRecentCompany,
Opportunities	Job code, nature, Details
Employer	Name, Opportunities, locations,
Employment Type	Should contain key Value pair Eg- 1 for the public servant 2 Government job 3 Corporate
Job Details	Description of job
Experience details	(link experiences with other job descriptions)
Locations	Country, State, City, Zipcode
Job skill set	Tech skills, Soft skills, Analytical skills
Job seeker details	Compensation, role, experience required, etc
Existing Employees	Employee code, name, sex, age, graduation uni, linked profile
Ancient job history (if any)	Employee code, company name, designation
UTD Alumni	Employee code, major, internships
Resume	File, Mapping(from Employee)
TBD	TBD

BUAN.6320.004
Database Foundations for Business Analytics

Opportunity Pool
Semester Project Group 12

Group Members:

Aman Sahu	Aman.Sahu@UTDallas.edu
Atharva Gupte	axg220265@UTDallas.edu
Chukwunonso Nnodum	Chukwunonso.Nnodum@UTDallas.edu
Jayesh Haryani	Jayesh.Haryani@UTDallas.edu
Keshav Sarraf	kxs230012@utdallas.edu
Malav Preyas Sheth	MalavPreyas.Sheth@UTDallas.edu

Phase II. Design and Modeling

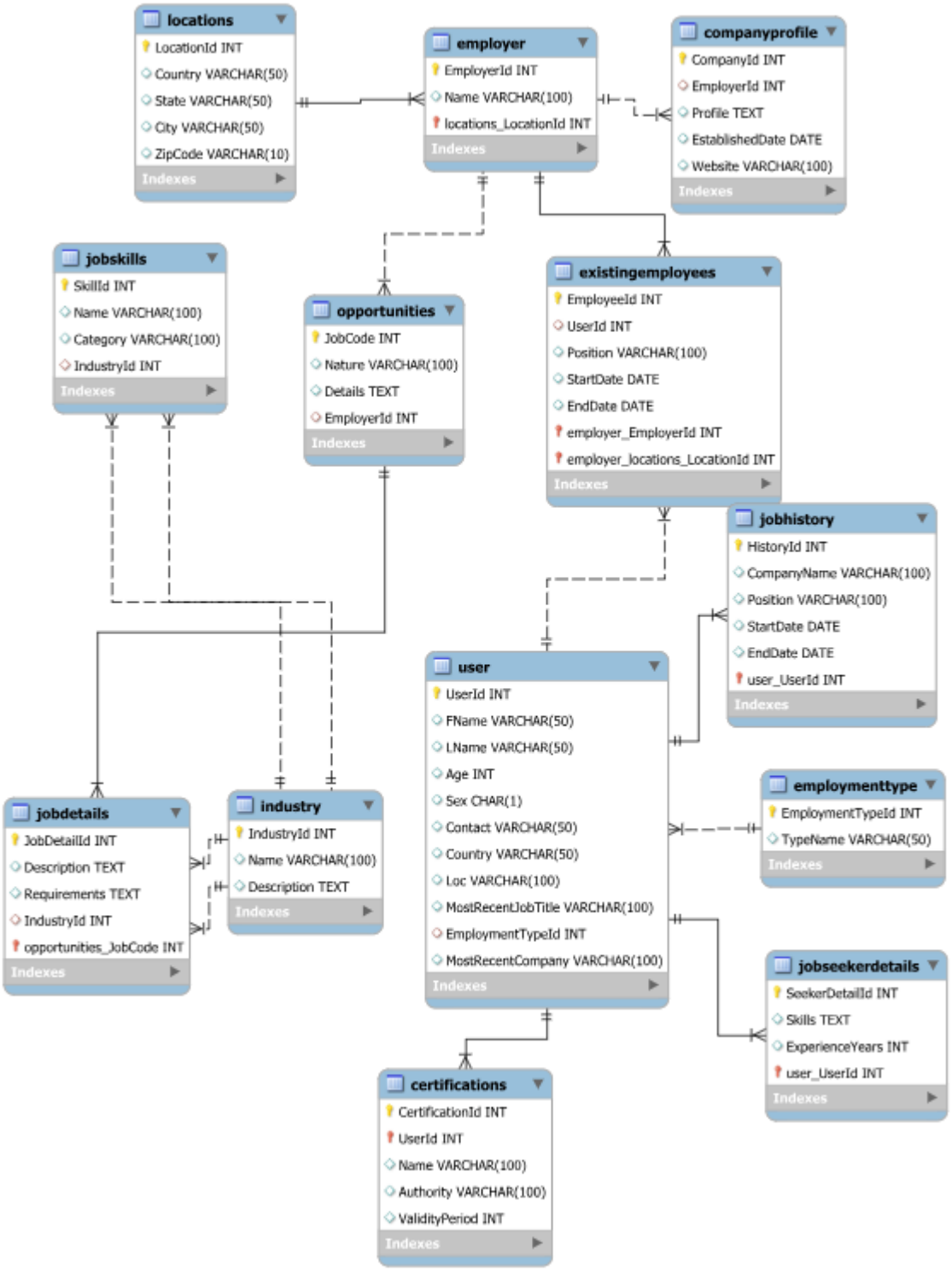
1. Executive Summary

In this project report, we delve into the logic design and modeling of our project. Section 1, provides an introduction to the project. Section 2 unveils our ER/EER diagram, along with all underlying assumptions, derived from Section 1. Continuing to Section 3, we present the relational schema, resulting from the transformation of the aforementioned ER/EER diagram. In Section 4, we meticulously document functional dependencies and normalize all tables to meet the third normal form (3NF) standards. To conclude, a concise summary is offered at the end of this report.

We propose the development of "Opportunity Pool," a comprehensive job database management system designed to simplify and enhance the job search and recruitment processes for both job seekers and employers. This centralized platform will provide users with access to a wealth of job-related information, enabling them to search, filter, and analyze data according to their specific needs. The system will feature a user-friendly interface to ensure ease of access and understanding.

2. Conceptual Design

2.1 ER Diagram



2.2 (Min, Max) Notation for Relationship

Relationship	(min, max) notation	Description
Job History to User	(1, *)	Each User can have one or more Job History records, but each Job History record can only be associated with one User record.
Company Profile to Employer	(1, 1)	Each Employer must have one Company Profile record, and each Company Profile record can only be associated with one Employer record.
Opportunities to Employer	(1, *)	Each Employer can have one or more Opportunities records, but each Opportunities record can only be associated with one Employer record.
Job Skills to Existing Employees	(1, *)	Each Existing Employee can have one or more Job Skill records, but each Job Skill record can only be associated with one Existing Employee record.
Existing Employees to Opportunities	(1, *)	Each Existing Employee can be associated with one or more Opportunities records, but each Opportunities record can only be associated with one Existing Employee record.
User to Certifications	(1, *)	Each User can have one or more Certification records, but each Certification record can only be associated with one User record.

The (min, max) notation in the image you provided specifies the minimum and maximum number of times that an entity in one table can participate in a relationship with an entity in another table.

For example, the relationship between the Job History and User tables has a (min, max) notation of (1, *). This means that each User can have one or more Job History records, but each Job History record can only be associated with one User record.

Another example is the relationship between the Company Profile and Employer tables. This relationship has a (min, max) notation of (1, 1). This means that each Employer must have one Company Profile record, and each Company Profile record can only be associated with one Employer record.

This EER diagram in itself contains all the necessary details of data format for each table with table names, attributes and the data type it is associated with

3. Functional Dependencies:

User table:

- UserId → FName, LName, Age, Sex, Contact, Country, Loc, MostRecentJobTitle, EmploymentTypeId, MostRecentCompany
- EmploymentTypeId → TypeName

JobSeekerDetails table:

- UserID → Skills, ExperienceYears

CompanyProfile table:

- EmployerId → Profile, EstablishedDate, Website

JobSkills table:

- EmployeeId → Skill

Opportunities table:

- EmployerId → Category, JobCode, UserId, IndustryId, Nature, Position, Details, StartDate, EndDate

JobHistory table:

- UserId → CompanyName, Position, StartDate, EndDate

Certifications table:

- UserId → Name, Authority, ValidityPeriod

Additional functional dependencies:

- LocationId → Country, State, Name
- OpportunityId → EmployerId
- JobSeekerDetailsID → UserID
- HistoryId → UserId
- CertificationId → UserId

These functional dependencies can be used to normalize the database and eliminate data redundancy.
For example, the dependency UserId → FName, LName, Age, Sex, Contact, Country, Loc, MostRecentJobTitle, EmploymentTypeId, MostRecentCompany means that the values for all of these attributes can be determined from the UserId alone.
This means that we could create a separate table for each of these attributes, and the UserId could be used to join these tables together. This would eliminate the need to store all of these attributes in the User table, which would save space and make the database more efficient.

4. Relationship Schema Diagram :

The following is a detailed relational schema for the given diagram:

Tables:
locations
company profile
job skills
existing employees
opportunities
job history
user
employment type
jobseeker details
certifications

Primary Keys:
locations: LocationId
companyprofile: CompanyId
jobskills: SkillId
existingemployees: EmployeeId
opportunities: OpportunityId
jobhistory: HistoryId
user: UserId
employmenttype: EmploymentTypeId
jobseekerdetails: SeekerDetailsID
certifications: CertificationId

Foreign Key Relationships:
companyprofile.EmployerId references user.UserId
jobskills.EmployeeId references existingemployees.EmployeeId
existingemployees.EmployerId references user.UserId
existingemployees.LocationId references locations.LocationId
opportunities.EmployerId references user.UserId
jobhistory.UserId references user.UserId
user.EmploymentTypeId references employmenttype.EmploymentTypeId
jobseekerdetails.UserID references user.UserId
certifications.UserId references user.UserId

Attributes:
locations: LocationId, Country, State, Name, Profile
companyprofile: CompanyId, EmployerId, Profile, EstablishedDate, Website
jobskills: SkillId, EmployeeId, Name
existingemployees: EmployeeId, EmployerId, City, LocationId
opportunities: OpportunityId, EmployerId, Category, JobCode, UserId, IndustryId, Nature, Position, Details, StartDate, EndDate
jobhistory: HistoryId, UserId, CompanyName, Position, StartDate, EndDate
user: UserId, FName, LName, Age, Sex, Contact, Country, Loc, MostRecentJobTitle, EmploymentTypeId, MostRecentCompany
employmenttype: EmploymentTypeId, TypeName

jobseekerdetails: SeekerDetailsID, UserID, Skills, ExperienceYears
certifications: CertificationId, UserId, Name, Authority, ValidityPeriod

Examples:
A location can have multiple company profiles, but a company profile can only belong to one location.
An employee can have multiple job skills, but a job skill can only belong to one employee.
An employee can work for multiple companies, but a company can only employ one employee per job position.
An opportunity can belong to only one company, but a company can have multiple opportunities.
A job seeker can have multiple job histories, but a job history can only belong to one job seeker.
A user can have only one employment type, but an employment type can be associated with multiple users.
A job seeker can have multiple certifications, but a certification can only belong to one job seeker.
This relational schema can be used to create a database that can store and manage all of the data in the given diagram. The foreign key relationships between the tables ensure that the data is consistent and accurate.

Data Format for Every Relation

Table	Attribute	Data Type
	LocationId	INT
	Country	VARCHAR(50)
	State	VARCHAR(50)
	Name	VARCHAR(100)
Locations	Profile	TEXT
	CompanyId	INT
	EmployerId	INT
	Profile	TEXT
	EstablishedDate	DATE
Company Profile	Website	VARCHAR(100)
	SkillId	INT
	EmployeeId	INT
Job Skills	Name	VARCHAR(100)
	EmployeeId	INT
	EmployerId	INT
	City	VARCHAR(50)
Existing Employees	LocationId	INT
	OpportunityId	INT
	EmployerId	INT
	Category	VARCHAR(100)
	JobCode	INT
	UserId	INT
	IndustryId	INT
	Nature	VARCHAR(100)
	Position	VARCHAR(100)
	Details	TEXT
	StartDate	DATE
End Date	EndDate	DATE
	HistoryId	INT
	UserId	INT
	CompanyName	VARCHAR(100)
	Position	VARCHAR(100)
Job History	StartDate	DATE

	EndDate	DATE
User	UserId	INT
	FName	VARCHAR(50)
	LName	VARCHAR(50)
	Age	INT
	Sex	CHAR(1)
	Contact	VARCHAR(50)
	Country	VARCHAR(50)
	Loc	VARCHAR(50)
	MostRecentJobTitle	VARCHAR(100)
	EmploymentTypeId	INT
	MostRecentCompany	VARCHAR(100)
Employment Type	EmploymentTypeId	INT
	TypeName	VARCHAR(50)
Job Seeker Details	SeekerDetailsID	INT
	UserID	INT
	Skills	TEXT
	ExperienceYears	INT
Certifications	CertificationId	INT
	UserId	INT
	Name	VARCHAR(100)
	Authority	VARCHAR(100)
	ValidityPeriod	INT

5. Normalization:

User

1NF

UserID	Fname	Lname	Age	Sex	Contact	Country	Loc	MostRecentJobTitle	EmploymentTypeID	MostRecentCompany
1	John	Doe	35	M	123456789	USA	NY	Software Engineer	1	Microsoft
2	Jane	Smith	28	F	987654321	UK	London	Data Scientist	2	Google

2NF

User Table

UserID	Fname	Lname	Age	Sex	Contact	Country	Loc
1	John	Doe	35	M	123456789	USA	NY
2	Jane	Smith	28	F	987654321	UK	London

Employment Table

Employment ID	UserID	Most RecentJobTitle	Employment Type ID	Most Recent Company
1	1	Software Engineer	1	Microsoft
2	2	Data Scientist	2	Google

3NF

User Table retained same from the second normal form

Employment Type Table

Employment ID	Employment Type
1	Software Engineer
2	Data Scientist

Company Table

CompanyID	Company Name
1	Microsoft
2	Google

Employment Table

Employment ID	UserID	EmploymentTypeID	CompanyID
1	1	1	1
2	2	2	2

The 3NF form is the most normalized form of the table. This is because it eliminates all transitive dependencies. A transitive dependency is a type of dependency where a non-prime attribute is dependent on another non-prime attribute. In the 2NF form, the EmploymentTypeID attribute is dependent on the MostRecentJobTitle attribute. However, in the 3NF form, the EmploymentTypeID attribute is no longer dependent on the MostRecentJobTitle attribute, as it is now stored in a separate table called Employment Type

JobSeekerDetails

1NF
The table is already in 1NF

2NF
In the "JobSeekerDetails" table, all non-prime attributes (Skills and ExperienceYears) are fully dependent on the entire primary key, which is SeekerDetailsID. Therefore, the table is already in 2NF.

3NF
To achieve 3NF, we need to eliminate any transitive dependencies. A transitive dependency occurs when a non-prime attribute is dependent on another non-prime attribute, rather than the primary key.
In the "JobSeekerDetails" table, there is a transitive dependency between Skills and ExperienceYears. This is because Skills are often related to a job seeker's experience years. To eliminate this transitive dependency, we can create a separate table to store the skills for each job seeker.

Job Seeker Skills Table

Job Seeker Skills ID	Job Seeker ID	Skill
1	1	Java
2	1	Python
3	2	C++
4	2	Go

JobSeekerDetails Table

SeekerDetailsID	UserID	ExperienceYears
1	1	5
2	2	3

By separating the "Skills" attribute into a distinct table, we eliminate the transitive dependency and ensure that each table only contains atomic values that are directly related to its primary key. This normalization

improves the overall integrity and consistency of the database. Therefore, the 3NF form of the "JobSeekerDetails" table is as follows:

3NF JobSeekerDetails Table

SeekerDetailsID	UserID	ExperienceYears
1	1	5
2	2	3

Job Seeker Skills Table

JobSeekerSkillsID	JobSeekerID	Skill
1	1	Java
2	1	Python
3	2	C++
4	2	JavaScript

Employment Type

The table, "Employment Type," is already in 1NF and 2NF.
1NF (First Normal Form)
A table is in 1NF when it adheres to the following conditions:
No repeating groups: Each attribute should have a single value for each row.
Atomic values: Each attribute should hold a single value, not multiple values.
No derived attributes: Attribute values should be directly stored, not calculated from other attributes.
The "Employment Type" table satisfies all three conditions:
There are no repeating groups. Each row contains unique values for "EmploymentTypeID" and "TypeName".
Each attribute holds a single value. There are no multi-valued attributes.
Attribute values are directly stored. They are not calculated from other attributes.

2NF (Second Normal Form)
To achieve 2NF, we need to eliminate any partial dependencies. A partial dependency occurs when a non-prime attribute is dependent on only a part of the primary key.
In the "Employment Type" table, the "TypeName" attribute is completely dependent on the primary key, which is "EmploymentTypeID". Therefore, the table is already in 2NF.

3NF (Third Normal Form)

A table is in 3NF when it satisfies both 1NF and 2NF, and additionally, it has no transitive dependencies. A transitive dependency occurs when a non-prime attribute is dependent on another non-prime attribute, rather than the primary key.
In the "Employment Type" table, there are no transitive dependencies. Therefore, the table is already in 3NF.
The "Employment Type" table is already in 1NF, 2NF, and 3NF, and it doesn't require any further normalization.

Company Profile

1NF
The table is already in 1NF, due to the following conditions:

No repeating groups: Each attribute has a single value for each row.
Atomic values: Each attribute holds a single value, not multiple values.
No derived attributes: Attribute values are directly stored, not calculated from other attributes.

2NF
To achieve 2NF, we need to eliminate any partial dependencies. A partial dependency occurs when a non-prime attribute is dependent on only a part of the primary key.
In the "Company Profile" table, all non-prime attributes (Profile, EstablishedDate, and Website) are fully dependent on the entire primary key, which is CompanyID. Therefore, the table is already in 2NF.

3NF
To achieve 3NF, we need to eliminate any transitive dependencies. A transitive dependency occurs when a non-prime attribute is dependent on another non-prime attribute, rather than the primary key.
In the "Company Profile" table, there are no transitive dependencies. Therefore, the table is already in 3NF.

Employment Type

1NF
The table, "Employment Type," is already in 1NF and 2NF.
A table is in 1NF when it adheres to the following conditions:
No repeating groups: Each attribute should have a single value for each row.
Atomic values: Each attribute should hold a single value, not multiple values.
No derived attributes: Attribute values should be directly stored, not calculated from other attributes.
The "Employment Type" table satisfies all three conditions:
There are no repeating groups. Each row contains unique values for "EmploymentTypeID" and "TypeName".
Each attribute holds a single value. There are no multi-valued attributes.
Attribute values are directly stored. They are not calculated from other attributes.

2NF
To achieve 2NF, we need to eliminate any partial dependencies. A partial dependency occurs when a non-prime attribute is dependent on only a part of the primary key.
In the "Employment Type" table, the "TypeName" attribute is completely dependent on the primary key, which is "EmploymentTypeID". Therefore, the table is already in 2NF.

3NF
A table is in 3NF when it satisfies both 1NF and 2NF, and additionally, it has no transitive dependencies. A transitive dependency occurs when a non-prime attribute is dependent on another non-prime attribute, rather than the primary key.
In the "Employment Type" table, there are no transitive dependencies. Therefore, the table is already in 3NF.
The "Employment Type" table is already in 1NF, 2NF, and 3NF, and it doesn't require any further normalization.

Employer Table

2NF

EmployerID	Name	LocationID
emp_id_1	emp_name_1	loc_1
emp_id_2	emp_name_2	loc_2

3NF

EmployerID	Name	LocationID
emp_id_1	emp_name_1	loc_1
emp_id_2	emp_name_2	loc_2

Location Table

1NF

LocationID	Country	State	City	zip
loc_1	country_1	state_1	city_1	zip_code_1
loc_2	country_2	state_2	city_2	zip_code_2

2NF

Country	State	City	zip
country_1	state_1	city_1	zip_code_1
country_2	state_2	city_2	zip_code_2

3NF

Country	State	City	zip
country_1	state_1	city_1	zip_code_1
country_2	state_2	city_2	zip_code_2

Existing Employee Table

1NF

Emp_ID	Employer_ID	User_ID	Start_date	End_date
emp_1	employer_1	user_1	start_1	end_1
emp_2	employer_2	user_2	start_2	end_2

2NF

Emp_ID	User_ID	Start_date	End_date
emp_1	user_1	start_1	end_1
emp_2	user_2	start_2	end_2

3NF

Emp_ID	User_ID	Start_date	End_date

emp_1	user_1	start_1	end_1
emp_2	user_2	start_2	end_2

Job History

1NF

Job_history_id	User_id	Company_name	Position	Start_date	End_date
J_hist_1	user_1	Company_1	pos_1	start_1	end_1
J_hist_2	user_2	Company_2	pos_2	start_2	end_2

2NF

Job_history_id	User_id	Company_name	Position	Start_date	End_date
J_hist_1	user_1	Company_1	pos_1	start_1	end_1
J_hist_2	user_2	Company_2	pos_2	start_2	end_2

3NF

Job_history_id	User_id	Company_name	Position	Start_date	End_date
J_hist_1	user_1	Company_1	pos_1	start_1	end_1
J_hist_2	user_2	Company_2	pos_2	start_2	end_2

Certification Table

1NF

Cert_id	user_id	name	Authority	Validity_period
cert_1	user_1	emp_name_1	authority_1	val_per_1
cert_2	user_2	emp_name_2	authority_2	val_per_2

2NF

Cert_id	user_id	Authority	Validity_period
cert_1	user_1	authority_1	val_per_1
cert_2	user_2	authority_2	val_per_2

3NF

Cert_id	user_id	Authority
cert_1	user_1	authority_1
cert_2	user_2	authority_2

Job Details Table

1NF

job_details_id	job_code	description	requirements	industry_id
job_det_1	job_code_1	job_desc_1	job_req_1	industry_1
job_det_2	job_code_2	job_desc_2	job_req_2	industry_2

2NF

job_details_id	job_code	description	requirements
job_det_1	job_code_1	job_desc_1	job_req_1
job_det_2	job_code_2	job_desc_2	job_req_2

3NF

job_details_id	description	requirements
job_det_1	job_desc_1	job_req_1
job_det_2	job_desc_2	job_req_2

Job Skill Table

1NF

skill_id	name	industry	category
skill_id_1	job_name_1	industry_1	job_category_1
skill_id_2	job_name_2	industry_2	job_category_2

2NF

skill_id	name	category
skill_id_1	job_name_1	job_category_1
skill_id_2	job_name_2	job_category_2

3NF

skill_id	name	category
skill_id_1	job_name_1	job_category_1
skill_id_2	job_name_2	job_category_2

Opportunities Table

1NF

job_code	nature	details	employer_Id
job_code_1	job_nature_1	job_det_1	emp_1
job_code_2	job_nature_2	job_det_2	emp_2

2NF

job_code	nature	employer_Id
job_code_1	job_nature_1	emp_1
job_code_2	job_nature_2	emp_2

3NF

job_code	nature	employer_Id
job_code_1	job_nature_1	emp_1
job_code_2	job_nature_2	emp_2

Industry Table

1NF

industry_id	name	description
industry_id_1	name_1	desc_1
industry_id_2	name_2	desc_2

2NF

industry_id	name	description
industry_id_1	name_1	desc_1
industry_id_2	name_2	desc_2

3NF

industry_id	name	description
industry_id_1	name_1	desc_1
industry_id_2	name_2	desc_2

6. Conclusion

In this report, we discuss and design the relational schema of the Opportunity Pool Job Management System. Our ER diagram and its associated relational schema show the conceptual and logical designs of the system. We also define data types and formats for each attribute in the relation. The next step is to implement this database. In the future, we may change some designs due to practical difficulties and other requirements.

Database Foundations for Business Analytics

(BUAN-6359)

GROUP – 12

Topic – Online Job Database

Group Members:

Aman Sahu	axs220513@utdallas.edu
Atharva Gupte	axg220265@UTDallas.edu
Chukwunonso Nnodum	Chukwunonso.Nnodum@UTDallas.edu
Jayesh Haryani	jxh230004@utdallas.edu
Keshav Sarraf	kxs230012@utdallas.edu
Malav Preyas Sheth	MalavPreyas.Sheth@UTDallas.edu

Brief Summary of the Project –

The development of “Opportunity Pool," a comprehensive job database management system designed to simplify and enhance the job search and recruitment processes for both job seekers and employers. This centralized platform will provide users with access to a wealth of job-related information, enabling them to search, filter, and analyze data according to their specific needs. The system will feature a user-friendly interface to ensure ease of access and understanding.

Features of the Project –

Comprehensive Job Data: The database will centralize job-related data, making it easy for job seekers to filter and sort job listings based on various criteria, simplifying the job-hunting process.

Data Backup and Recovery: Implement regular data backup and recovery procedures to safeguard against data loss or system failures. The database cluster will be implemented.

Data Security: Ensuring the security and confidentiality of user data through a database authentication system.

Scalability: Design the system to be scalable, accommodating a growing user base and expanding the database.

Skills and Roles Exploration: Job hunters can explore various skills and roles within the database to gain insights into career development and growth opportunities.

Employer Access: Employers can use the system to access a vast pool of potential employees, streamlining their hiring processes and finding the right candidates efficiently.

Professional Networking: The system will feature an opportunity pool that allows users to build professional networks by connecting with others' profiles, filtered by criteria such as schools and companies.

Job Posting Interface: Employers can post job openings through the user interface, expanding their reach to a large number of job hunters. Job seekers can stay updated with new opportunities.

Intelligent Matching: The database will offer insights into matching relevant candidates with job opportunities, based on skills, experience, and job requirements. These insights can be shared with employers to enhance their hiring decisions.

Table of Contents –

List of Tables –

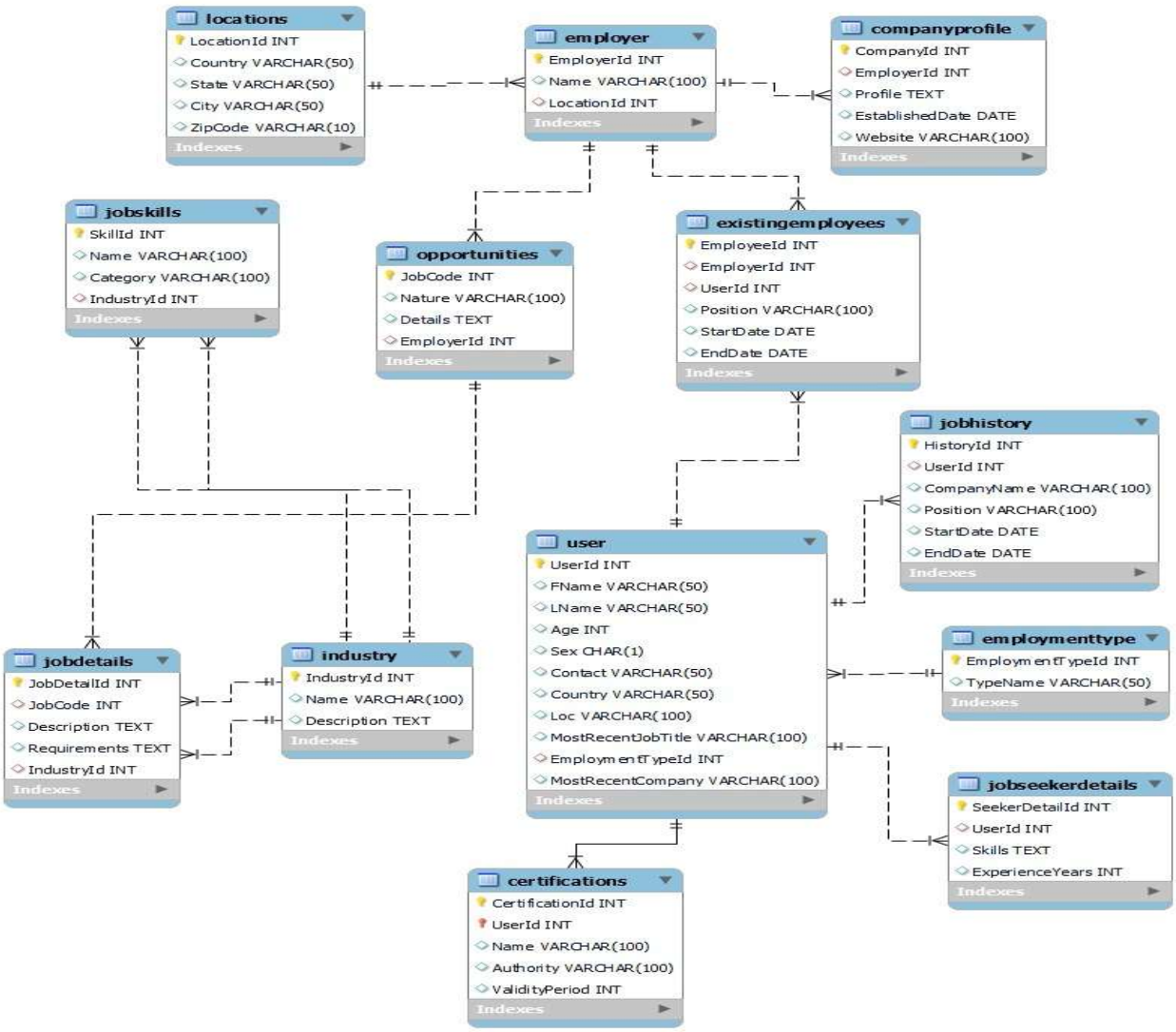
S.no	***Table***
1	User
2	Existing employee
3	Employment type
4	Location
5	Job details
6	Job skills
7	Education
8	Company profile
9	Opportunities
10	Certifications
11	employer
12	Job seeker details
13	Industry
14	Review

Phase 3 – Implementation

0. Pre-Illumination

This report outlines the implementation phase of the database project, focusing on the creation of the database, table setup, data population, SQL queries, and. Our project utilizes the MySQL database management system. Part 1 is the modified relational schema. Part 2 is the creation of the database, including tables, all other structures as well as constraints, data type, and format, Part 3 is the query scenario design and implementation.

1. Relational Schema



2. Creation of Database with SQL Database

2.1 Table Creation –

```
/* JobDetails: Contains detailed information about job opportunities.
JobSkills: Lists various skills associated with job opportunities.
JobSeekerDetails: Stores additional information about job seekers.
ExistingEmployees: Holds information about employees currently working
 with an employer.
JobHistory: Stores the job history of users.
Education: Records the educational background of users.
ProfessionalNetwork: Manages connections between different users for n
etworking purposes.
JobApplications: Tracks the job applications submitted by users.
Reviews: Allows users to post reviews about employers or job experienc
es.
Company Profile: Applicants know about the job they are applying for.
Industry: Type of industry the job is in.
Certifications: Applicant skills add on certifications.
*/


-- User Table
-- Employment Type Table

-------------------------------------------------------------------------
CREATE TABLE IF NOT EXISTS employmenttype
  (
     employmenttypeid INT auto_increment PRIMARY KEY,
     typename         VARCHAR(50)
  );


-------------------------------------------------------------------------
CREATE TABLE IF NOT EXISTS user
  (
     userid             INT auto_increment PRIMARY KEY,
     fname              VARCHAR(50),
     lname              VARCHAR(50),
     age                INT,
     sex                CHAR(1),
     contact            VARCHAR(50),
     country            VARCHAR(50),
     loc                VARCHAR(100),
     mostrecentjobtitle VARCHAR(100),
     employmenttypeid   INT,
     mostrecentcompany  VARCHAR(100),
     FOREIGN KEY (employmenttypeid) REFERENCES employmenttype(employme
```

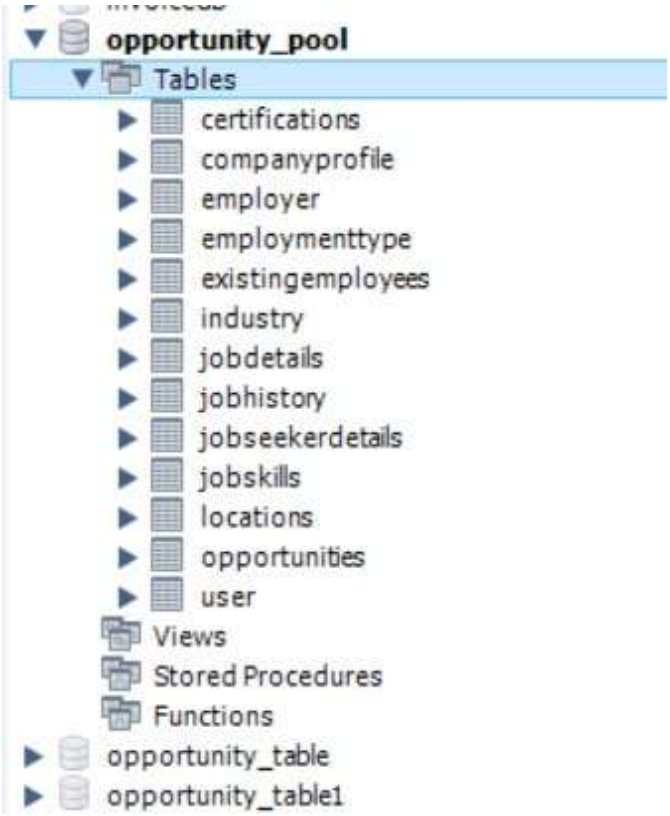
```
nttypeid)
  );

-------------------------------------------------------------------------
- Opportunities Table
CREATE TABLE IF NOT EXISTS opportunities
  (
     jobcode     INT auto_increment PRIMARY KEY,
     nature      VARCHAR(100),
     details     TEXT,
     employerid INT,
     FOREIGN KEY (employerid) REFERENCES employer(employerid)
  );

-------------------------------------------------------------------------
- Employer Table
CREATE TABLE IF NOT EXISTS employer
  (
     employerid INT auto_increment PRIMARY KEY,
     name        VARCHAR(100),
     locationid INT,
     FOREIGN KEY (locationid) REFERENCES locations(locationid)
  );

-------------------------------------------------------------------------
- Locations Table
CREATE TABLE IF NOT EXISTS locations
  (
     locationid INT auto_increment PRIMARY KEY,
     country     VARCHAR(50),
     state       VARCHAR(50),
     city        VARCHAR(50),
     zipcode     VARCHAR(10)
  );

-------------------------------------------------------------------------
- JobDetails Table
CREATE TABLE IF NOT EXISTS jobdetails
  (
     jobdetailid  INT auto_increment PRIMARY KEY,
     jobcode      INT,
     description  TEXT,
     requirements TEXT,
     FOREIGN KEY (jobcode) REFERENCES opportunities(jobcode)
  );

-------------------------------------------------------------------------
- JobSkills Table
```

```
CREATE TABLE IF NOT EXISTS jobskills
  (
     skillid  INT auto_increment PRIMARY KEY,
     name     VARCHAR(100),
     category VARCHAR(100)
  );

-----------------------------------------------------------------------
- JobSeekerDetails Table
CREATE TABLE IF NOT EXISTS jobseekerdetails
  (
     seekerdetailid  INT auto_increment PRIMARY KEY,
     userid          INT,
     skills          TEXT,
     experienceyears INT,
     FOREIGN KEY (userid) REFERENCES user(userid)
  );

-----------------------------------------------------------------------
- ExistingEmployees Table
CREATE TABLE IF NOT EXISTS existingemployees
  (
     employeeid INT auto_increment PRIMARY KEY,
     employerid INT,
     userid     INT,
     position   VARCHAR(100),
     startdate  DATE,
     enddate    DATE,
     FOREIGN KEY (employerid) REFERENCES employer(employerid),
     FOREIGN KEY (userid) REFERENCES user(userid)
  );

-----------------------------------------------------------------------
- JobHistory Table
CREATE TABLE IF NOT EXISTS jobhistory
  (
     historyid   INT auto_increment PRIMARY KEY,
     userid      INT,
     companyname VARCHAR(100),
     position    VARCHAR(100),
     startdate   DATE,
     enddate     DATE,
     FOREIGN KEY (userid) REFERENCES user(userid)
  );

-----------------------------------------------------------------------
-- Education Table
CREATE TABLE IF NOT EXISTS education
```

```
  (
     educationid INT auto_increment PRIMARY KEY,
     userid      INT,
     institution VARCHAR(100),
     degree      VARCHAR(100),
     startyear   INT,
     endyear     INT,
     FOREIGN KEY (userid) REFERENCES user(userid)
  );


-----------------------------------------------------------------------
-- ProfessionalNetwork Table
CREATE TABLE IF NOT EXISTS professionalnetwork
  (
     networkid      INT auto_increment PRIMARY KEY,
     userid1        INT,
     userid2        INT,
     connectiondate DATE,
     FOREIGN KEY (userid1) REFERENCES user(userid),
     FOREIGN KEY (userid2) REFERENCES user(userid)
  );


-----------------------------------------------------------------------
-- JobApplications Table
CREATE TABLE IF NOT EXISTS jobapplications
  (
     applicationid   INT auto_increment PRIMARY KEY,
     jobcode         INT,
     userid          INT,
     applicationdate DATE,
     status          VARCHAR(50),
     FOREIGN KEY (jobcode) REFERENCES opportunities(jobcode),
     FOREIGN KEY (userid) REFERENCES user(userid)
  );

-----------------------------------------------------------------------
-- Reviews Table
CREATE TABLE IF NOT EXISTS reviews
  (
     reviewid   INT auto_increment PRIMARY KEY,
     employerid INT,
     userid     INT,
     rating     INT,
     comment    TEXT,
     reviewdate DATE,
     FOREIGN KEY (employerid) REFERENCES employer(employerid),
```

```
    FOREIGN KEY (userid) REFERENCES user(userid)
  );

-----------------------------------------------------------------------
-- CompanyProfile Table
CREATE TABLE IF NOT EXISTS companyprofile
  (
    companyid       INT auto_increment PRIMARY KEY,
    employerid      INT,
    profile         TEXT,
    establisheddate DATE,
    website         VARCHAR(100),
    FOREIGN KEY (employerid) REFERENCES employer(employerid)
  );


-----------------------------------------------------------------------
-- Industry Table
CREATE TABLE IF NOT EXISTS industry
  (
    industryid  INT auto_increment PRIMARY KEY,
    name        VARCHAR(100),
    description TEXT
  );


-----------------------------------------------------------------------
-- Certifications Table
CREATE TABLE IF NOT EXISTS certifications
  (
    certificationid INT auto_increment,
    userid          INT NOT NULL,
    name            VARCHAR(100),
    authority       VARCHAR(100),
    validityperiod  INT, -- Validity period in years
    PRIMARY KEY (certificationid, userid),
    FOREIGN KEY (userid) REFERENCES user(userid)
  );
-----------------------------------------------------------------------
-
- Adding foreign key relationships for JobDetails and JobSkills to Ind
ustry
ALTER TABLE jobdetails
  ADD COLUMN industryid INT;

ALTER TABLE jobskills
  ADD COLUMN industryid INT;
```

```
ALTER TABLE jobdetails
  ADD FOREIGN KEY (industryid) REFERENCES industry(industryid);

ALTER TABLE jobskills
  ADD FOREIGN KEY (industryid) REFERENCES industry(industryid);
```



2.2 Database state

To ensure the database is populated for testing and development purposes, dummy data was inserted into each table. The following records were added to each table, maintaining data consistency and validity. One sample set of query is shown below;

insert into user (UserId, FName, LName, Age, Sex, Contact, Country, Loc, MostRecentJobTitle, EmploymentTypeId, MostRecentCompany)

values (8001, "Anil", "Kumble", 23, "M", 9999866893, "USA", "Dallas", "Bowling Staff", "16", "Dallas Sports");

insert into certifications (CertificationId, UserId, Name, Authority, ValidityPeriod)

values (1050, 8001, "Pitch assesment", "ICC", 6);

insert into locations(LocationId, Country, State, City, ZipCode) values (3001, "USA", "TX", "Dallas", 50103);

insert into employer (EmployerId, Name, LocationId) values (4001, "Apple", 3001);

insert into existingemployees (EmployeeId, EmployerId, Position)

values (7002, 4001, "Software Engineer B2");

insert into opportunities (JobCode, Nature, Details, EmployerId) values (3001, "Engineering", "Software Development Cycle", 4001);

insert into jobdetails (JobDetailId, JobCode, Description, Requirements)

values (2002, 3001, "Candidate should have proficiency in Java or Python or C++. Good knowledge of OOPS", "Must have Bachelor's degree in CS or realted field");

insert into companyprofile (CompanyId, EmployerId, Profile, Website) values (902, 4001, "Agile Based", "www.google.com/Agile");

insert into education (EducationId, UserId, Institution, Degree, StartYear, EndYear)

values (6001, 8001, "MIT", "MS in BA", 2022, 2024);

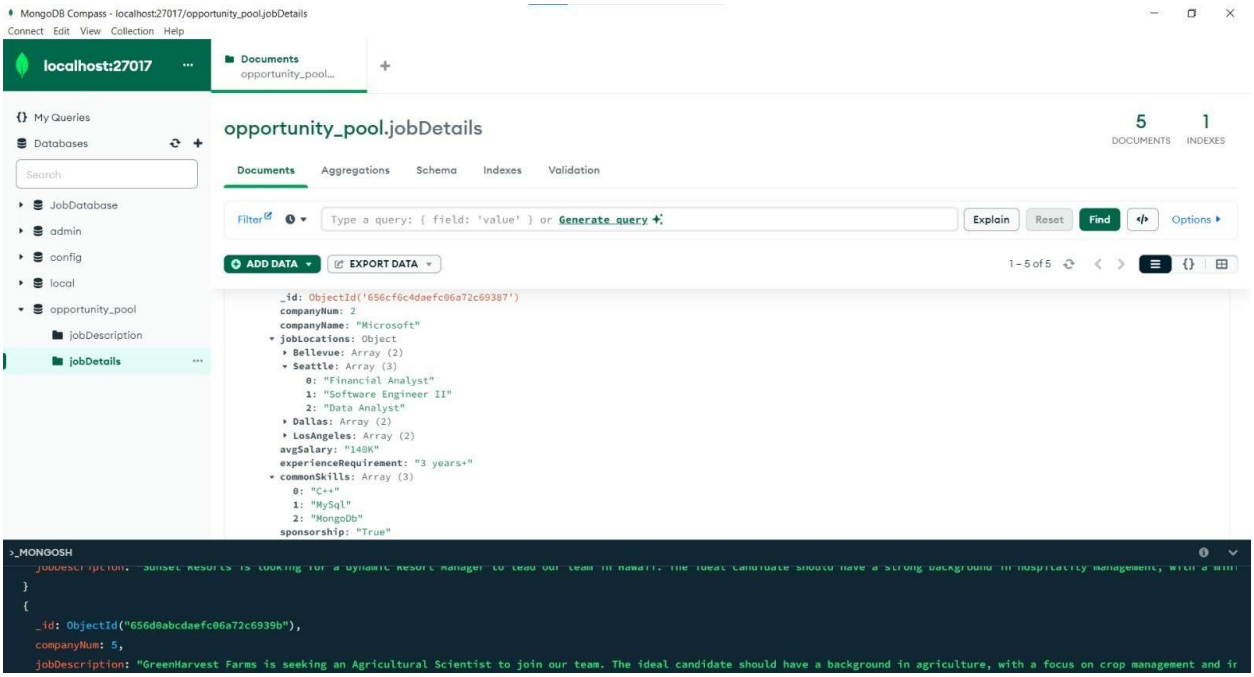
insert into jobskills (SkillId, Name, Category) values (90001, "Python", "Programming language");

We have attached the complete data entry at the end of the report.

2.3 Mongodb

Job details and job descriptions given the type of data becomes very difficult to contain data in SQL database. Thus, we migrated these two tables to mongoDb given the flexibility it provides to store data in unstructured format and we can store long strings along.

Below is attached a snippet of mongoDb.



MongoDb Query

use opportunity_pool

db.jobDescription.find()

db.jobDetails.find()

3 - Query Scenario Design (Including question and answers)

#1 Retrive users who are working in the field of software and hold any one certificate

```
SELECT u.fname,
       u.lname,
       u.age,
       u.sex,
       u.contact,
```

```
        u.country,
        u.mostrecentjobtitle,
        u.mostrecentcompany,
        c.NAME       AS Certification_Name,
        c.authority AS Certified_from
FROM    USER AS u
        INNER JOIN certifications AS c
                ON u.userid = c.userid
        INNER JOIN employmenttype AS et
                ON u.employmenttypeid = et.employmenttypeid
WHERE   et.typename = "information technology"
        AND u.userid IN (SELECT userid
                         FROM   certifications
                         GROUP  BY userid
                         HAVING Count(certificationid) >= 1);
```

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

FName	LName	Age	Sex	Contact	Country	MostRecentJobTitle	MostRecentCompany	Certification_Name	Certified_from
Olivia	Liu	30	F	9999888999	USA	UX Researcher	UserInsights Corp	UX Researcher Certification	UX Research Association
Ethan	Perez	28	M	9999000888	Canada	Software Developer	CodeCrafters Inc	Software Development Certification	Software Developers Association

#2 A user is applying to google, he/she wants to know existing employess at Google so he can know positions and other inforamtions

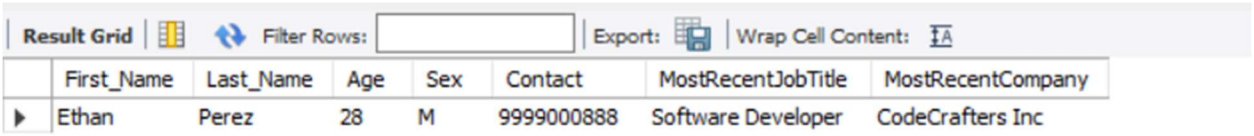
```
SELECT e.*
FROM   existingemployees AS e
       INNER JOIN employer AS emp
               ON e.employerid = emp.employerid
WHERE  emp.NAME = "google";
```

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

EmployeeId	EmployerId	UserId	Position	StartDate	EndDate
7004	4003	NULL	Data Scientist	NULL	NULL

#3 Retrieve users who have a degree in Computer Science and are employed in the Information Technology sector.

```
SELECT u.fname AS First_Name,
       u.lname AS Last_Name,
       u.age,
       u.sex,
       u.contact,
       u.mostrecentjobtitle,
       u.mostrecentcompany
FROM   USER AS u
WHERE  u.employmenttypeid = (SELECT employmenttypeid
                             FROM   employmenttype
                             WHERE  typename = "information technology
")
       AND u.userid IN (SELECT userid
                        FROM   education
                        WHERE  degree = "bachelor of computer science"
);
```

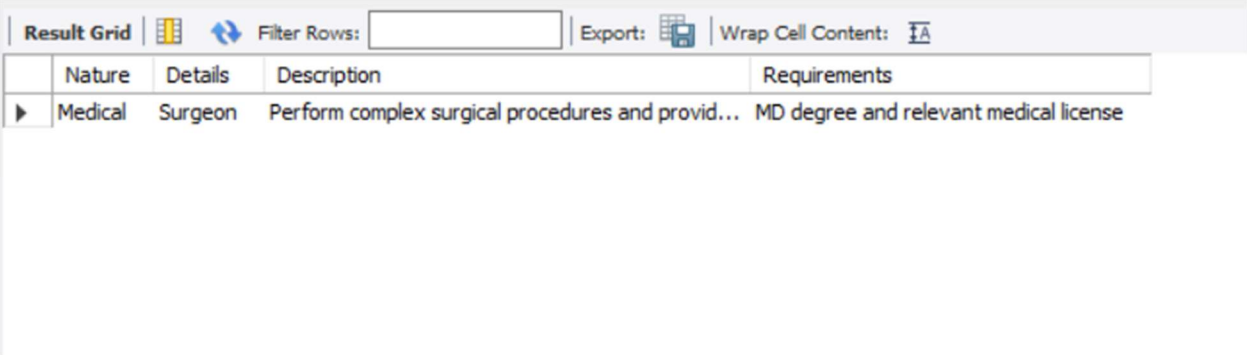


Result Grid | Filter Rows: | Export: | Wrap Cell Content:

First_Name	Last_Name	Age	Sex	Contact	MostRecentJobTitle	MostRecentCompany
Ethan	Perez	28	M	9999000888	Software Developer	CodeCrafters Inc

#4 retrive job details of surgeon

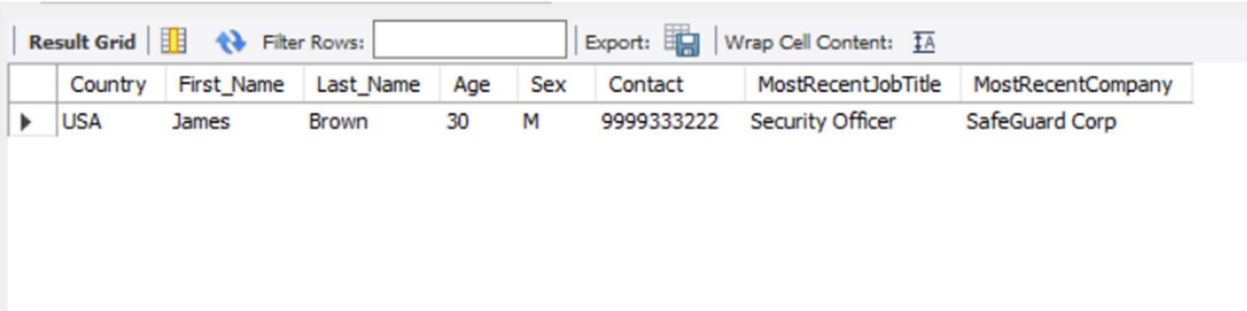
```
SELECT o.nature,
       o.details,
       j.description,
       j.requirements
FROM   jobdetails AS j
       JOIN opportunities AS o
         ON j.jobcode = o.jobcode
WHERE  o.details = "surgeon";
```



Nature	Details	Description	Requirements
Medical	Surgeon	Perform complex surgical procedures and provid...	MD degree and relevant medical license

#5 Retrive users that are from USA and Canada and their most recent job titile is security officer

```
SELECT country,
       fname AS First_Name,
       lname AS Last_Name,
       age,
       sex,
       contact,
       mostrecentjobtitle,
       mostrecentcompany
FROM   USER
WHERE  mostrecentjobtitle = "security officer"
       AND country IN ( "usa", "canada" );
```



Country	First_Name	Last_Name	Age	Sex	Contact	MostRecentJobTitle	MostRecentCompany
USA	James	Brown	30	M	9999333222	Security Officer	SafeGuard Corp

#6 Retrive users that are into sports industry(employer type) and are under age of 24. Also find the location of the job opportunities in this field.alter

```
SELECT fname AS First_Name,
       lname AS Last_Name,
       age,
       sex,
```

```
        contact,
        mostrecentjobtitle,
        mostrecentcompany
FROM    USER
WHERE   employmenttypeid = (SELECT employmenttypeid
                            FROM   employmenttype
                            WHERE  typename = "sports")
        AND age < 24;
```

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

First_Name	Last_Name	Age	Sex	Contact	MostRecentJobTitle	MostRecentCompany
Anil	Kumble	23	M	9999866893	Bowling Staff	Dallas Sports

#7 retrive employees age less than 30 seeking jobs in the field of agriculture

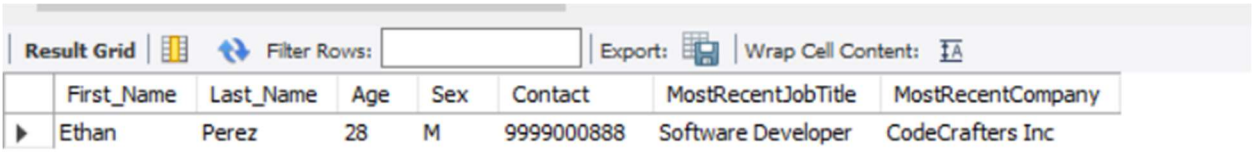
```
SELECT USER.fname,
       USER.lname,
       age,
       sex,
       country,
       mostrecentjobtitle
FROM   USER
       JOIN employmenttype
         ON USER.employmenttypeid = employmenttype.employmenttypeid
WHERE  typename = "agriculture"
       AND age < 30;
```

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

Fname	LName	Age	Sex	Country	MostRecentJobTitle
Ava	Lee	25	F	USA	Agricultural Scientist
Liam	Rodriguez	29	M	Canada	Environmental Scientist

#8 Retrieve users who have a degree in Computer Science and are employed in the Information Technology sector.

```
SELECT u.fname AS First_Name,
       u.lname AS Last_Name,
       u.age,
       u.sex,
       u.contact,
       u.mostrecentjobtitle,
       u.mostrecentcompany
FROM   USER AS u
WHERE  u.employmenttypeid = (SELECT employmenttypeid
                             FROM   employmenttype
                             WHERE  typename = "information technology
")
       AND u.userid IN (SELECT userid
                        FROM   education
                        WHERE  degree = "bachelor of computer science"
);
```



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

	First_Name	Last_Name	Age	Sex	Contact	MostRecentJobTitle	MostRecentCompany
▶	Ethan	Perez	28	M	9999000888	Software Developer	CodeCrafters Inc

#9 Retrieve users who have certifications related to healthcare and are from the United States.

```
SELECT u.fname      AS First_Name,
       u.lname      AS Last_Name,
       u.age,
       u.sex,
       u.contact,
       u.mostrecentjobtitle,
       u.mostrecentcompany,
       c.NAME       AS Certification_Name,
       c.authority AS Certified_from
FROM   USER AS u
       INNER JOIN certifications AS c
               ON u.userid = c.userid
```

```sql
WHERE  u.country = "usa"
       AND c.NAME IN (SELECT NAME
                      FROM   certifications
                      WHERE  NAME = "medical board certification");
```

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

First_Name	Last_Name	Age	Sex	Contact	MostRecentJobTitle	MostRecentCompany	Certification_Name	Certified_from
Emily	Davis	26	F	9999666555	Medical Doctor	NYC Hospital	Medical Board Certification	Medical Council

#10 Identifying Users by Age Group

#categorize users into age groups (e.g., "Under 25," "25-34," "35-44," etc.) based on their ages.

```sql
SELECT userid,
       fname,
       lname,
       CASE
         WHEN age < 25 THEN 'Under 25'
         WHEN age BETWEEN 25 AND 34 THEN '25-34'
         WHEN age BETWEEN 35 AND 44 THEN '35-44'
         ELSE '45 and above'
       END AS Age_Group
FROM   USER;
```

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

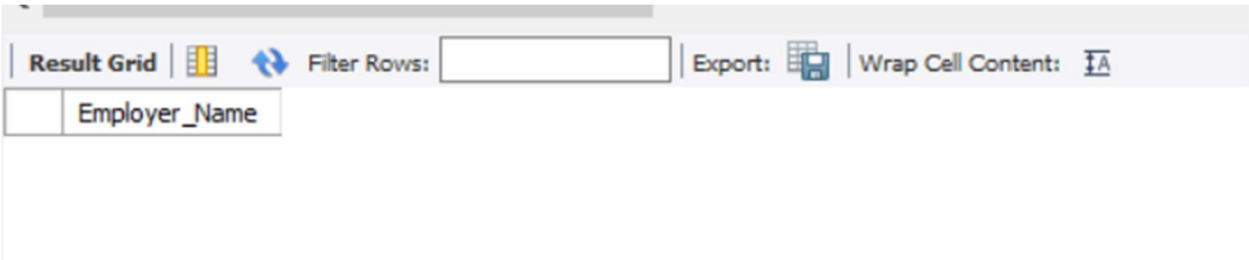
UserId	FName	LName	Age_Group
2	Maria	John	25-34
8001	Anil	Kumble	Under 25
8002	Sarah	Johnson	25-34
8003	Michael	Smith	35-44
8004	Emily	Davis	25-34
8005	David	Wilson	25-34
8006	Emma	Anderson	25-34
8007	James	Brown	25-34
8008	Olivia	Martin	25-34
8009	William	Thompson	25-34
8010	Ava	Lee	25-34
8011	Liam	Martinez	25-34
8012	Sophia	Garcia	25-34
8013	Mason	Lopez	25-34
8014	Aria	Perez	25-34
8015	Ethan	Wang	25-34
8016	Isabella	Hernandez	25-34
8017	Noah	Gomez	25-34
8018	Ava	Torres	25-34
8019	Liam	Rodriguez	25-34
8020	Olivia	Liu	25-34
8021	Lucas	Lee	25-34
8022	Zoe	Chen	25-34
8023	Liam	Nguyen	35-44
8024	Emma	Smith	25-34

Result 31

#11 Find Employers with No Job Openings

Identify employers who currently have no job openings.

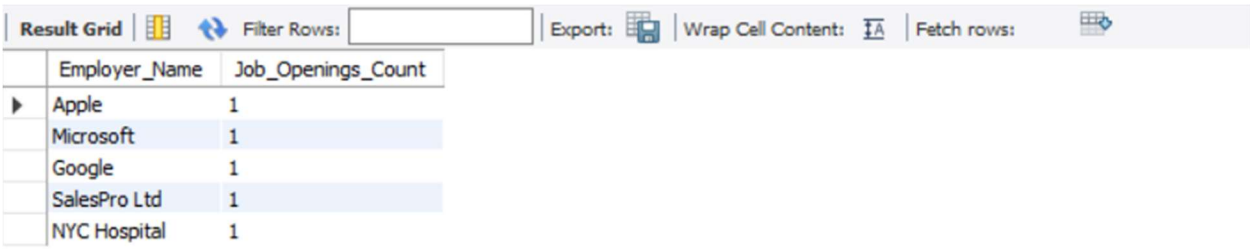
```
SELECT e.NAME AS Employer_Name
FROM   employer AS e
       LEFT JOIN opportunities AS o
              ON e.employerid = o.employerid
WHERE  o.jobcode IS NULL;
```



#12 Identify Employers with the Most Job Openings

Find the top 5 employers who have the most job openings available.

```
SELECT e.name           AS Employer_Name,
       Count(o.jobcode) AS Job_Openings_Count
FROM   employer AS e
       LEFT JOIN opportunities AS o
              ON e.employerid = o.employerid
GROUP  BY e.name
ORDER  BY job_openings_count DESC
LIMIT  5;
```



Employer_Name	Job_Openings_Count
Apple	1
Microsoft	1
Google	1
SalesPro Ltd	1
NYC Hospital	1

4. Conclusion

In conclusion, the development of the Job Database Project has been a comprehensive and successful endeavor, aimed at enhancing the efficiency and effectiveness of job management within our organization. The project has successfully leveraged the power of SQL to create a robust and scalable database.

Through the implementation of a well-structured schema, normalization techniques, and optimized queries, we have achieved a streamlined and organized database that facilitates seamless data retrieval, insertion, and modification. The project's user-friendly interface allows for easy navigation and interaction, ensuring that both technical and non-technical users can leverage the system with minimal training.

Overall, the Job Database Project represents a significant step forward in optimizing our job management processes. Its successful implementation promises to streamline our recruitment workflows, enhance collaboration among team members, and ultimately contribute to the overall success and growth of our organization. As we move forward, the project will continue to evolve, incorporating feedback and adapting to the changing needs of our dynamic business environment.

Complete data entry

```
INSERT INTO USER
            (userid,
             fname,
             lname,
             age,
             sex,
             contact,
             country,
             loc,
             mostrecentjobtitle,
             employmenttypeid,
             mostrecentcompany)
VALUES      (8002,
             "sarah",
             "johnson",
             29,
             "f",
             9999888777,
             "canada",
             "toronto",
             "project manager",
             10,
             "tech solutions inc"),
            (8003,
             "michael",
             "smith",
             35,
             "m",
             9999777666,
             "uk",
             "london",
             "sales manager",
             4,
             "salespro ltd"),
            (8004,
             "emily",
             "davis",
             26,
             "f",
             9999666555,
             "usa",
             "new york",
             "medical doctor",
             3,
```

```
 "nyc hospital"),
(8005,
 "david",
 "wilson",
 32,
 "m",
 9999555444,
 "canada",
 "vancouver",
 "teacher",
 14,
 "maplewood school"),
(8006,
 "emma",
 "anderson",
 28,
 "f",
 9999444333,
 "australia",
 "sydney",
 "chef",
 9,
 "taste buds restaurant"),
(8007,
 "james",
 "brown",
 30,
 "m",
 9999333222,
 "usa",
 "chicago",
 "security officer",
 8,
 "safeguard corp"),
(8008,
 "olivia",
 "martin",
 27,
 "f",
 9999222111,
 "uk",
 "manchester",
 "fashion designer",
 12,
 "elegant designs ltd"),
(8009,
 "william",
 "thompson",
```

```
             31,
             "m",
             9999111000,
             "canada",
             "montreal",
             "construction manager",
             11,
             "buildit inc"),
            (8010,
             "ava",
             "lee",
             25,
             "f",
             9999000999,
             "usa",
             "los angeles",
             "agricultural scientist",
             13,
             "greenfields research");

INSERT INTO certifications
            (certificationid,
             userid,
             NAME,
             authority,
             validityperiod)
VALUES      (1051,
             8002,
             "project management professional",
             "project management institute",
             3),
            (1052,
             8003,
             "certified sales professional",
             "sales excellence institute",
             2),
            (1053,
             8004,
             "medical board certification",
             "medical council",
             5),
            (1054,
             8005,
             "teaching certification",
             "education authority",
             4),
            (1055,
             8006,
```

```
             "culinary arts certification",
             "culinary institute",
             2),
            (1056,
             8007,
             "security guard license",
             "state security bureau",
             1),
            (1057,
             8008,
             "fashion design certification",
             "fashion institute",
             3),
            (1058,
             8009,
             "construction project management",
             "construction institute",
             3),
            (1059,
             8010,
             "agricultural research certification",
             "agricultural council",
             2),
            (1060,
             8002,
             "advanced project management",
             "project management institute",
             2);

INSERT INTO locations
            (locationid,
             country,
             state,
             city,
             zipcode)
VALUES      (3002,
             "canada",
             "on",
             "toronto",
             "m5h 2n2"),
            (3003,
             "uk",
             "england",
             "london",
             "wc2n 5du"),
            (3004,
             "usa",
             "ny",
```

```
             "new york",
             "10001"),
            (3005,
             "canada",
             "bc",
             "vancouver",
             "v6b 1g1"),
            (3006,
             "australia",
             "nsw",
             "sydney",
             "2000"),
            (3007,
             "usa",
             "il",
             "chicago",
             "60601"),
            (3008,
             "uk",
             "england",
             "manchester",
             "m1 1fn"),
            (3009,
             "canada",
             "qc",
             "montreal",
             "h3b 2e3"),
            (3010,
             "usa",
             "ca",
             "los angeles",
             "90001"),
            (3011,
             "germany",
             "bavaria",
             "munich",
             "80331");

INSERT INTO employer
            (employerid,
             NAME,
             locationid)
VALUES      (4002,
             "microsoft",
             3002),
            (4003,
             "google",
             3003),
```

```
            (4004,
             "salespro ltd",
             3004),
            (4005,
             "nyc hospital",
             3005),
            (4006,
             "taste buds restaurant",
             3006),
            (4007,
             "safeguard corp",
             3007),
            (4008,
             "elegant designs ltd",
             3008),
            (4009,
             "buildit inc",
             3009),
            (4010,
             "greenfields research",
             3010),
            (4011,
             "autotech gmbh",
             3011);

INSERT INTO existingemployees
            (employeeid,
             employerid,
             position)
VALUES      (7003,
             4002,
             "software engineer"),
            (7004,
             4003,
             "data scientist"),
            (7005,
             4004,
             "sales representative"),
            (7006,
             4005,
             "medical doctor"),
            (7007,
             4006,
             "chef"),
            (7008,
             4007,
             "security officer"),
            (7009,
```

```
             4008,
             "fashion designer"),
            (7010,
             4009,
             "construction manager"),
            (7011,
             4010,
             "agricultural scientist"),
            (7012,
             4011,
             "mechanical engineer");

INSERT INTO opportunities
            (jobcode,
             nature,
             details,
             employerid)
VALUES      (3002,
             "software development",
             "backend developer",
             4002),
            (3003,
             "data science",
             "data analyst",
             4003),
            (3004,
             "sales",
             "sales manager",
             4004),
            (3005,
             "medical",
             "surgeon",
             4005),
            (3006,
             "culinary arts",
             "head chef",
             4006),
            (3007,
             "security",
             "security supervisor",
             4007),
            (3008,
             "fashion design",
             "fashion designer",
             4008),
            (3009,
             "construction",
             "construction project manager",
```

```
             4009),
            (3010,
             "agricultural research",
             "research scientist",
             4010),
            (3011,
             "mechanical engineering",
             "mechanical engineer",
             4011);

INSERT INTO jobdetails
            (jobdetailid,
             jobcode,
             description,
             requirements)
VALUES      (2003,
             3002,
"responsible for designing and developing backend applications using j
ava and sql"
             ,
"bachelor's degree in computer science or related field"),
            (2004,
             3003,
"apply machine learning algorithms to analyze and extract insights fro
m data",
"master's degree in data science or related field"),
            (2005,
             3004,
             "lead and manage a sales team to achieve revenue targets"
,
             "bachelor's degree in sales or related field"),
            (2006,
             3005,
"perform complex surgical procedures and provide medical care to patie
nts",
"md degree and relevant medical license"),
            (2007,
             3006,
"create and innovate culinary dishes in a high-
end restaurant setting",
"culinary degree and extensive culinary experience"),
            (2008,
             3007,
"ensure the security and safety of premises, including monitoring and
responding to security threats"
             ,
"security training and certification"),
            (2009,
```

```
             3008,
"design and develop fashion collections, including clothing, accessori
es, and footwear"
             ,
"degree in fashion design or related field"),
            (2010,
             3009,
"plan, manage, and oversee construction projects from concept to compl
etion",
"bachelor's degree in civil engineering or related field"),
            (2011,
             3010,
"conduct research in agriculture, crop management, and sustainable far
ming practices"
             ,
"ph.d. in agriculture or related field"),
            (2012,
             3011,
"design and optimize mechanical systems, including machinery and mecha
nical components"
             ,
"bachelor's degree in mechanical engineering or equivalent");

INSERT INTO companyprofile
            (companyid,
             employerid,
             profile,
             website)
VALUES      (909,
             4004,
             "leading provider of marketing solutions and advertising
services",
             "www.advertisenow.com"),
            (910,
             4005,
             "data-driven insights and analytics for businesses",
             "www.datainsights.com"),
            (911,
             4006,
             "innovative technology solutions and software development
",
             "www.techinnovators.com"),
            (912,
             4007,
             "web design and development services for businesses",
             "www.webcraftersinc.com"),
            (913,
             4008,
```

```
             "pioneering in data science and analytics",
             "www.datagenius.com"),
            (914,
             4009,
             "software development and it solutions provider",
             "www.codemasters.com"),
            (915,
             4010,
             "technology solutions for operations and devops",
             "www.techopssolutions.com"),
            (916,
             4011,
             "automotive technology and engineering solutions",
             "www.autotechgmbh.com"),
            (917,
             4002,
             "consumer electronics and software innovation",
             "www.microsoft.com"),
            (918,
             4003,
             "internet services, search, and cloud computing",
             "www.google.com");

INSERT INTO education
            (educationid,
             userid,
             institution,
             degree,
             startyear,
             endyear)
VALUES      (6002,
             8002,
             "harvard university",
             "mba",
             2018,
             2020),
            (6003,
             8003,
             "oxford university",
             "ph.d. in computer science",
             2016,
             2021),
            (6004,
             8004,
             "stanford medical school",
             "doctor of medicine (md)",
             2012,
             2018),
```

```
            (6005,
             8005,
             "university of toronto",
             "bachelor of education (b.ed.)",
             2014,
             2016),
            (6006,
             8006,
             "le cordon bleu",
             "culinary arts diploma",
             2013,
             2015),
            (6007,
             8007,
             "security training institute",
             "certified security officer",
             2019,
             2019),
            (6008,
             8008,
             "fashion institute of design & merchandising",
             "bachelor of science in fashion design",
             2016,
             2020),
            (6009,
             8009,
             "engineering university",
             "bachelor of civil engineering",
             2010,
             2014),
            (6010,
             8010,
             "agricultural research institute",
             "ph.d. in agriculture",
             2015,
             2019),
            (6011,
             8002,
             "mit",
             "master of science in business analytics",
             2021,
             2023);

INSERT INTO jobskills
            (skillid,
             NAME,
             category)
VALUES      (90002,
```

```
             "data analysis",
             "data science"),
            (90003,
             "sales management",
             "sales"),
            (90004,
             "medical diagnosis",
             "medical"),
            (90005,
             "teaching and instruction",
             "education"),
            (90006,
             "culinary arts",
             "culinary"),
            (90007,
             "security management",
             "security"),
            (90008,
             "fashion design",
             "fashion"),
            (90009,
             "project management",
             "construction"),
            (90010,
             "agricultural research",
             "agriculture"),
            (90011,
             "mechanical engineering",
             "mechanical");

INSERT INTO USER
            (userid,
             fname,
             lname,
             age,
             sex,
             contact,
             country,
             loc,
             mostrecentjobtitle,
             employmenttypeid,
             mostrecentcompany)
VALUES      (8011,
             "liam",
             "martinez",
             27,
             "m",
             9999888111,
```

```
 "usa",
 "houston",
 "electrical engineer",
 11,
 "powertech inc"),
(8012,
 "sophia",
 "garcia",
 32,
 "f",
 9999777222,
 "spain",
 "barcelona",
 "marketing manager",
 4,
 "marketmasters sl"),
(8013,
 "mason",
 "lopez",
 30,
 "m",
 9999666333,
 "canada",
 "toronto",
 "dentist",
 3,
 "bright smiles clinic"),
(8014,
 "aria",
 "perez",
 28,
 "f",
 9999555444,
 "usa",
 "los angeles",
 "graphic designer",
 12,
 "creativeworks inc"),
(8015,
 "ethan",
 "wang",
 29,
 "m",
 9999444555,
 "china",
 "beijing",
 "civil engineer",
 5,
```

```
 "bridgebuilders ltd"),
(8016,
 "isabella",
 "hernandez",
 26,
 "f",
 9999333666,
 "mexico",
 "mexico city",
 "biomedical researcher",
 3,
 "healthtech research institute"),
(8017,
 "noah",
 "gomez",
 34,
 "m",
 9999222777,
 "usa",
 "chicago",
 "financial analyst",
 10,
 "financepro llc"),
(8018,
 "ava",
 "torres",
 31,
 "f",
 9999111888,
 "uk",
 "london",
 "architect",
 11,
 "architectsrus"),
(8019,
 "liam",
 "rodriguez",
 29,
 "m",
 9999000999,
 "canada",
 "vancouver",
 "environmental scientist",
 13,
 "ecosolutions inc"),
(8020,
 "olivia",
 "liu",
```

```
 30,
 "f",
 9999888999,
 "usa",
 "san francisco",
 "ux researcher",
 7,
 "userinsights corp"),
(8021,
 "lucas",
 "lee",
 33,
 "m",
 9999888222,
 "canada",
 "calgary",
 "mechanical engineer",
 11,
 "mechanotech ltd"),
(8022,
 "zoe",
 "chen",
 27,
 "f",
 9999777111,
 "australia",
 "melbourne",
 "interior designer",
 12,
 "designscape pty"),
(8023,
 "liam",
 "nguyen",
 35,
 "m",
 9999666555,
 "usa",
 "boston",
 "investment analyst",
 10,
 "investmentpros llc"),
(8024,
 "emma",
 "smith",
 28,
 "f",
 9999555333,
 "uk",
```

```
 "manchester",
 "pharmacist",
 3,
 "pharmacare ltd"),
(8025,
 "elijah",
 "brown",
 32,
 "m",
 9999444222,
 "usa",
 "dallas",
 "geologist",
 13,
 "geoexplorers inc"),
(8026,
 "mia",
 "davis",
 30,
 "f",
 9999333444,
 "canada",
 "montreal",
 "hr manager",
 4,
 "hrpros inc"),
(8027,
 "henry",
 "martinez",
 29,
 "m",
 9999222111,
 "usa",
 "san diego",
 "product manager",
 10,
 "productmasters inc"),
(8028,
 "aria",
 "garcia",
 26,
 "f",
 9999111999,
 "spain",
 "madrid",
 "electrical engineer",
 11,
 "energetics ltd"),
```

```
            (8029,
             "ethan",
             "perez",
             28,
             "m",
             9999000888,
             "canada",
             "ottawa",
             "software developer",
             7,
             "codecrafters inc"),
            (8030,
             "olivia",
             "smith",
             31,
             "f",
             9999888333,
             "usa",
             "miami",
             "legal counsel",
             4,
             "legaleagle attorneys");

INSERT INTO certifications
            (certificationid,
             userid,
             NAME,
             authority,
             validityperiod)
VALUES      (1061,
             8011,
             "electrical engineering certification",
             "engineering institute",
             2),
            (1062,
             8012,
             "digital marketing certification",
             "digital marketing association",
             3),
            (1063,
             8013,
             "oral surgery certification",
             "dental board",
             5),
            (1064,
             8014,
             "graphic design certification",
             "designers guild",
```

```
 2),
(1065,
 8015,
 "structural engineering certification",
 "engineering council",
 3),
(1066,
 8016,
 "biomedical research certification",
 "biomedical association",
 2),
(1067,
 8017,
 "financial analyst certification",
 "finance institute",
 3),
(1068,
 8018,
 "architectural license",
 "architects association",
 4),
(1069,
 8019,
 "environmental scientist certification",
 "environmental council",
 2),
(1070,
 8020,
 "ux researcher certification",
 "ux research association",
 3),
(1071,
 8021,
 "mechanical engineering certification",
 "engineering institute",
 2),
(1072,
 8022,
 "interior design certification",
 "designers guild",
 3),
(1073,
 8023,
 "investment analyst certification",
 "finance institute",
 2),
(1074,
 8024,
```

```
             "pharmacy license",
             "pharmacists association",
             4),
            (1075,
             8025,
             "geologist certification",
             "geologists society",
             3),
            (1076,
             8026,
             "hr management certification",
             "hr professionals institute",
             2),
            (1077,
             8027,
             "product management certification",
             "product managers association",
             3),
            (1078,
             8028,
             "electrical engineering certification",
             "engineering institute",
             2),
            (1079,
             8029,
             "software development certification",
             "software developers association",
             3),
            (1080,
             8030,
             "legal counsel certification",
             "bar association",
             4);

INSERT INTO locations
            (locationid,
             country,
             state,
             city,
             zipcode)
VALUES      (3012,
             "germany",
             "bavaria",
             "munich",
             "80331"),
            (3013,
             "france",
             "ile-de-france",
```

```
 "paris",
 "75001"),
(3014,
 "brazil",
 "sao paulo",
 "sao paulo",
 "01000"),
(3015,
 "india",
 "maharashtra",
 "mumbai",
 "400001"),
(3016,
 "china",
 "shanghai",
 "shanghai",
 "200000"),
(3017,
 "australia",
 "victoria",
 "melbourne",
 "3000"),
(3018,
 "mexico",
 "mexico city",
 "mexico city",
 "01000"),
(3019,
 "japan",
 "tokyo",
 "tokyo",
 "100-0001"),
(3020,
 "south korea",
 "seoul",
 "seoul",
 "04501"),
(3021,
 "italy",
 "lazio",
 "rome",
 "00100"),
(3022,
 "spain",
 "madrid",
 "madrid",
 "28001"),
(3023,
```

```
             "argentina",
             "buenos aires",
             "buenos aires",
             "1000"),
            (3024,
             "russia",
             "moscow",
             "moscow",
             "101000"),
            (3025,
             "south africa",
             "gauteng",
             "johannesburg",
             "2000"),
            (3026,
             "canada",
             "ab",
             "calgary",
             "t2p 3p6"),
            (3027,
             "usa",
             "ca",
             "san francisco",
             "94101"),
            (3028,
             "uk",
             "england",
             "birmingham",
             "b1 1aa"),
            (3029,
             "canada",
             "on",
             "ottawa",
             "k1p 1a4"),
            (3030,
             "usa",
             "fl",
             "miami",
             "33101"),
            (3031,
             "spain",
             "valencia",
             "valencia",
             "46001");

INSERT INTO employer
            (employerid,
             NAME,
```

```
             locationid)
VALUES       (4012,
              "siemens ag",
              3012),
             (4013,
              "l'oreal",
              3013),
             (4014,
              "banco do brasil",
              3014),
             (4015,
              "tata consultancy services",
              3015),
             (4016,
              "alibaba group",
              3016),
             (4017,
              "anz banking group",
              3017),
             (4018,
              "cemex",
              3018),
             (4019,
              "sony corporation",
              3019),
             (4020,
              "samsung electronics",
              3020),
             (4021,
              "ferrari",
              3021),
             (4022,
              "telefonica",
              3022),
             (4023,
              "ypf sa",
              3023),
             (4024,
              "gazprom",
              3024),
             (4025,
              "standard bank group",
              3025),
             (4026,
              "suncor energy",
              3026),
             (4027,
              "salesforce",
```

```
             3027),
            (4028,
             "rolls-royce",
             3028),
            (4029,
             "bank of montreal",
             3029),
            (4030,
             "american express",
             3030),
            (4031,
             "valencia city council",
             3031);

INSERT INTO existingemployees
            (employeeid,
             employerid,
             position)
VALUES      (7013,
             4012,
             "electrical engineer"),
            (7014,
             4013,
             "marketing coordinator"),
            (7015,
             4014,
             "financial analyst"),
            (7016,
             4015,
             "software developer"),
            (7017,
             4016,
             "data scientist"),
            (7018,
             4017,
             "civil engineer"),
            (7019,
             4018,
             "architect"),
            (7020,
             4019,
             "environmental scientist"),
            (7021,
             4020,
             "ux designer"),
            (7022,
             4021,
             "mechanical engineer"),
```

```
            (7023,
             4022,
             "interior designer"),
            (7024,
             4023,
             "investment analyst"),
            (7025,
             4024,
             "pharmacist"),
            (7026,
             4025,
             "geologist"),
            (7027,
             4026,
             "hr manager"),
            (7028,
             4027,
             "product manager"),
            (7029,
             4028,
             "electrical engineer"),
            (7030,
             4029,
             "software developer"),
            (7031,
             4030,
             "legal counsel"),
            (7032,
             4031,
             "city planner");

INSERT INTO companyprofile
            (companyid,
             employerid,
             profile,
             website)
VALUES      (919,
             4012,
             "global leader in electrification, automation, and digita
lization",
             "www.siemens.com"),
            (920,
             4013,
             "world's largest cosmetics and beauty company",
             "www.loreal.com"),
            (921,
             4014,
             "one of the largest banks in brazil",
```

```
             "www.bb.com.br"),
            (922,
             4015,
             "it services, consulting, and business solutions provider
",
             "www.tcs.com"),
            (923,
             4016,
             "leading technology conglomerate",
             "www.alibabagroup.com"),
            (924,
             4017,
             "australian multinational banking and financial services
company",
             "www.anz.com"),
            (925,
             4018,
             "global building materials company",
             "www.cemex.com"),
            (926,
             4019,
             "japanese multinational conglomerate",
             "www.sony.net"),
            (927,
             4020,
             "south korean multinational electronics company",
             "www.samsung.com"),
            (928,
             4021,
             "italian luxury sports car manufacturer",
             "www.ferrari.com"),
            (929,
             4022,
             "spanish multinational telecommunications company",
             "www.telefonica.com"),
            (930,
             4023,
             "argentine multinational oil company",
             "www.ypf.com"),
            (931,
             4024,
             "russian multinational energy corporation",
             "www.gazprom.com"),
            (932,
             4025,
             "south african financial services provider",
             "www.standardbank.co.za"),
            (933,
```

```
             4026,
             "canadian integrated energy company",
             "www.suncor.com"),
            (934,
             4027,
             "leading customer relationship management (crm) platform"
,
             "www.salesforce.com"),
            (935,
             4028,
             "british luxury car and aero engine manufacturer",
             "www.rolls-roycemotorcars.com"),
            (936,
             4029,
             "canadian multinational investment bank",
             "www.bmo.com"),
            (937,
             4030,
             "american multinational financial services corporation",
             "www.americanexpress.com"),
            (938,
             4031,
             "local government authority for valencia, spain",
             "www.valencia.es");

INSERT INTO education
            (educationid,
             userid,
             institution,
             degree,
             startyear,
             endyear)
VALUES      (6012,
             8011,
             "university of texas",
             "bachelor of science in electrical engineering",
             2013,
             2017),
            (6013,
             8012,
             "london school of economics",
             "master of science in marketing",
             2015,
             2016),
            (6014,
             8013,
             "university of sao paulo",
             "doctor of dental surgery (dds)",
```

```
 2008,
 2013),
(6015,
 8014,
 "art institute of los angeles",
 "bachelor of fine arts in graphic design",
 2010,
 2014),
(6016,
 8015,
 "university of beijing",
 "master of science in civil engineering",
 2014,
 2016),
(6017,
 8016,
 "national autonomous university of mexico",
 "ph.d. in biomedical research",
 2012,
 2018),
(6018,
 8017,
 "university of chicago",
 "master of business administration (mba)",
 2011,
 2013),
(6019,
 8018,
 "university college london",
 "bachelor of architecture",
 2013,
 2017),
(6020,
 8019,
 "university of british columbia",
 "master of environmental science",
 2015,
 2017),
(6021,
 8020,
 "university of melbourne",
 "master of interaction design",
 2017,
 2019),
(6022,
 8021,
 "tsinghua university",
 "master of science in mechanical engineering",
```

```
 2014,
 2016),
(6023,
 8022,
 "madrid polytechnic university",
 "bachelor of interior design",
 2012,
 2016),
(6024,
 8023,
 "harvard business school",
 "master of finance",
 2010,
 2012),
(6025,
 8024,
 "university of manchester",
 "ph.d. in pharmacy",
 2013,
 2018),
(6026,
 8025,
 "moscow state university",
 "master of science in geology",
 2010,
 2015),
(6027,
 8026,
 "university of montreal",
 "master of human resources management",
 2012,
 2014),
(6028,
 8027,
 "stanford graduate school of business",
 "master of business administration (mba)",
 2015,
 2017),
(6029,
 8028,
 "polytechnic university of madrid",
 "bachelor of science in electrical engineering",
 2013,
 2017),
(6030,
 8029,
 "carleton university",
 "bachelor of computer science",
```

```
             2014,
             2018),
            (6031,
             8030,
             "university of miami",
             "juris doctor (jd)",
             2017,
             2020);

INSERT INTO jobskills
            (skillid,
             NAME,
             category)
VALUES      (90012,
             "renewable energy",
             "engineering"),
            (90013,
             "content marketing",
             "marketing"),
            (90014,
             "oral surgery",
             "medical"),
            (90015,
             "curriculum development",
             "education"),
            (90016,
             "pastry arts",
             "culinary"),
            (90017,
             "risk management",
             "finance"),
            (90018,
             "3d modeling",
             "design"),
            (90019,
             "ecology",
             "environmental science"),
            (90020,
             "usability testing",
             "ux design"),
            (90021,
             "robotics engineering",
             "engineering"),
            (90022,
             "furniture design",
             "design"),
            (90023,
             "portfolio management",
```

```
             "finance"),
            (90024,
             "pharmacology",
             "medical"),
            (90025,
             "seismology",
             "geology"),
            (90026,
             "employee relations",
             "hr management"),
            (90027,
             "product development",
             "product management"),
            (90028,
             "power systems",
             "engineering"),
            (90029,
             "web development",
             "software development"),
            (90030,
             "legal research",
             "legal"),
            (90031,
             "urban planning",
             "urban planning");

INSERT INTO jobdetails
            (jobdetailid,
             jobcode,
             description,
             requirements)
VALUES      (2033,
             3012,
             "design electrical systems for industrial applications",
             "bachelor's degree in electrical engineering"),
            (2034,
             3013,
             "develop marketing campaigns and strategies",
             "bachelor's degree in marketing or related field"),
            (2035,
             3014,
             "analyze financial data and provide insights",
             "bachelor's degree in finance or related field"),
            (2036,
             3015,
             "develop software applications for clients",
             "bachelor's degree in computer science"),
            (2037,
```

```
             3016,
             "perform data analysis and machine learning tasks",
             "master's degree in data science or related field"),
            (2038,
             3017,
             "design and supervise construction projects",
             "bachelor's degree in civil engineering"),
            (2039,
             3018,
             "architectural design and project management",
             "bachelor's degree in architecture"),
            (2040,
             3019,
             "conduct environmental research and analysis",
             "bachelor's degree in environmental science"),
            (2041,
             3020,
             "user experience design and usability testing",
             "bachelor's degree in ux design or related field"),
            (2042,
             3021,
             "design mechanical systems and components",
             "bachelor's degree in mechanical engineering"),
            (2043,
             3022,
             "create interior designs for residential and commercial s
paces",
             "bachelor's degree in interior design"),
            (2044,
             3023,
             "analyze investment opportunities and financial markets",
             "bachelor's degree in finance"),
            (2045,
             3024,
             "dispense medications and provide pharmaceutical care",
             "pharmd or equivalent pharmacy degree"),
            (2046,
             3025,
             "conduct geological surveys and research",
             "bachelor's degree in geology"),
            (2047,
             3026,
             "manage hr functions and employee relations",
             "bachelor's degree in hr management or related field"),
            (2048,
             3027,
             "product planning and management",
             "bachelor's degree in business or related field"),
```

```
            (2049,
             3028,
             "design electrical systems for various applications",
             "bachelor's degree in electrical engineering"),
            (2050,
             3029,
             "develop software applications and solutions",
             "bachelor's degree in computer science"),
            (2051,
             3030,
             "provide legal advice and representation",
             "juris doctor (jd) or equivalent law degree"),
            (2052,
             3031,
             "city planning and urban development",
             "bachelor's degree in urban planning or related field");

INSERT INTO opportunities
            (jobcode,
             nature,
             details,
             employerid)
VALUES      (3032,
             "engineering",
             "electrical engineer",
             4012),
            (3033,
             "marketing",
             "digital marketing specialist",
             4013),
            (3034,
             "finance",
             "financial analyst",
             4014),
            (3035,
             "software development",
             "frontend developer",
             4015),
            (3036,
             "data science",
             "machine learning engineer",
             4016),
            (3037,
             "engineering",
             "civil engineer",
             4017),
            (3038,
             "architecture",
```

```
 "architect",
 4018),
(3039,
 "environmental science",
 "environmental scientist",
 4019),
(3040,
 "ux design",
 "ux/ui designer",
 4020),
(3041,
 "engineering",
 "mechanical engineer",
 4021),
(3042,
 "interior design",
 "interior designer",
 4022),
(3043,
 "finance",
 "investment analyst",
 4023),
(3044,
 "medical",
 "pharmacist",
 4024),
(3045,
 "geology",
 "geologist",
 4025),
(3046,
 "hr management",
 "hr manager",
 4026),
(3047,
 "product management",
 "product manager",
 4027),
(3048,
 "engineering",
 "electrical engineer",
 4028),
(3049,
 "software development",
 "software engineer",
 4029),
(3050,
 "legal",
```

```
 "legal counsel",
 4030),
(3051,
 "urban planning",
 "city planner",
 4031);
```