

JOB APPLICATION TRACKER

PROJECT DONE BY:

NAME : K.ANAND

REGISTER NUMBER : RA2211003020438

YEAR/SEM : IV/VII

**COLLEGE : SRM INSTITUTE OF SCIENCE AND
TECHNOLOGY, RAMAPURAM**

ABSTRACT

Tracking job applications has become a crucial for job seekers in today's quickly changing labour market, particularly for recent graduates, early professionals, and those applying to many positions at once. Managing numerous applications and keeping track of information such as job role, company name, application date, follow-up status, or interview schedule can become burden with the rise of online job portals, internships, and company-specific hiring platforms. By providing an organized, digital way to store, handle, and evaluate job application data using a relational database format, a job application tracker seeks to solve this issue.

The three main tables in the database are Jobs, Applications, and Status. Key data including the firm name, job title, and location are kept in the Jobs table. The date of application, job ID (connected by a foreign key), and current application status are all recorded in the Applications table. The Status table serves as a reference table with standard application stages like "Applied," "Interview Scheduled," "Offered," "Rejected," and so on. This relational model is straightforward but strong enough to address a range of questions that represent a job seeker's true needs.

INTRODUCTION

Job seekers frequently apply to several organizations at once in today's competitive job market using a variety of venues, including recruiting drives, job boards, and company portals. It might seem challenging and complicated to manually keep track of all these programs. It's easy for information to be confused or forgotten, including application dates, firm names, job roles, interview schedules, and offer statuses.

A straightforward database system called the Job Application Tracker project was created to centrally arrange and track all job application actions. This system enables users to record application details, maintain their status, and retrieve important information through structured queries by utilizing SQL and relational database principles. It makes it easier for users to keep track of where they have applied, what stage each application is in, and what they should do next.

With an emphasis on backend database development, this project showcases expertise in table design, relationships, SQL query data retrieval, and ER diagram usage. It offers a workable answer to a real-world issue and is perfect for students who want to show prospective employers that they are proficient with databases

SCOPE

The goal of the Job Application Tracker project is to use relational databases to manage numerous job applications in a centralized, effective, and organized manner. People apply to many different organizations in the modern job-search climate, and each one has its own application deadlines, interview procedures, and status reports. Managing and remembering these details becomes difficult and prone to mistakes if an inadequate system is not in place. The goal of this project is to create a database-driven system that makes it possible for users to efficiently save, edit, and retrieve comprehensive job application.

This project focuses on the backend aspect of application management, which includes creating pertinent SQL queries to communicate with the data and designing and implementing a database structure. To keep a clear focus on understanding database fundamentals, the frontend component—such as a graphical user interface or web integration.

The project's goal is to model a real-world scenario in which users may follow applications across different industries, businesses, jobs, and application statuses with the aid of structured data systems. It enables students to comprehend the importance of data modelling, query processing, and data integrity management by focusing only on SQL and relational database concepts.

OBJECTIVE

The Job Application Tracker project's primary objectives are:

- To build an organized database system for storing information about job applications.
- To make it simple to monitor the status of applications (e.g., Applied, Interview, Offer, Rejected).
- To make it easier to search and filter by status, role, or firm.
- To use SQL queries to offer insights (e.g., total applications, success rate).
- To exhibit in a practical setting a solid understanding of SQL and database concepts.

TOOLS AND TECHNOLOGIES

The Job Application Tracker project combines basic database development, productivity tools, and front-end design. A list of the technologies and tools used is provided below:

1. Database Management System, or MySQL
utilized for relational database creation and administration.

entails building tables, drafting DDL and DML SQL queries, and carrying out data operations.

guarantees the integrity, standardization, and consistency of data.

2. Structured Query Language (SQL)
fundamental language for database interaction.

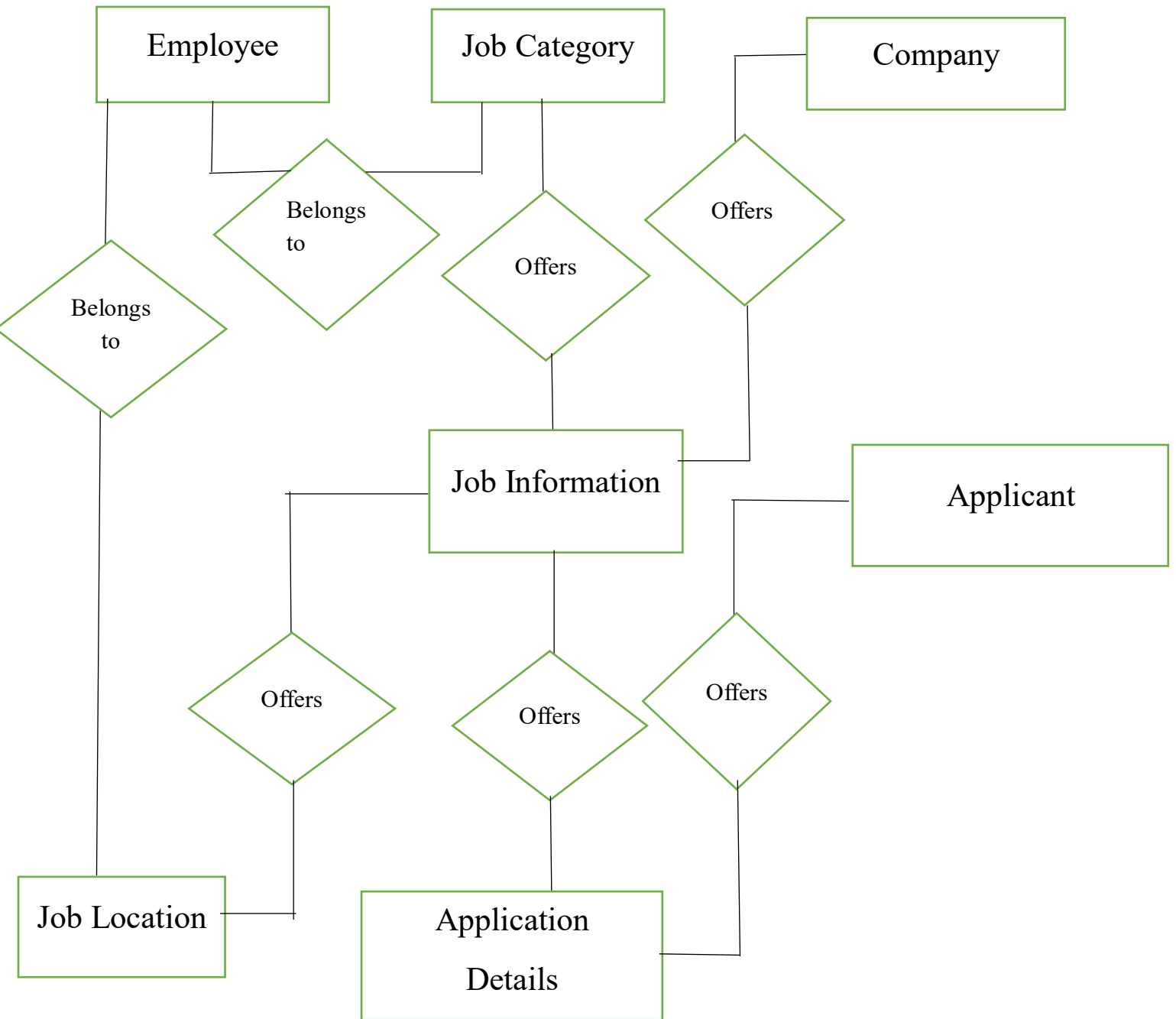
Table creation (CREATE), data insertion (INSERT), retrieval (SELECT), updating (UPDATE), and record deletion (DELETE) are all accomplished with this tool.

Analytical functions were performed using JOIN, WHERE, ORDER BY, and GROUP BY queries.

3. Tools for DB/ER Diagrams
The Entity-Relationship Diagram was created using programs like draw.io, Lucidchart, or dbdiagram.io aids in better understanding the data structure and visualizing the links between tables.

4. Google Docs and Microsoft Word
The abstract, introduction, tools, objectives, questions, and output are all used to document the project.

ER DIAGRAM



SQL QUERIES

```
mysql> create table company(company_id int, company_name varchar(20), location varchar(20));
Query OK, 0 rows affected (0.13 sec)

mysql> create table job(job_id int, job_title varchar(20), job_type varchar(20));
Query OK, 0 rows affected (0.03 sec)

mysql> create table application(application_id int, applied_date DATE, application_status varchar(20));
Query OK, 0 rows affected (0.04 sec)
```

```
mysql> insert into company values(101, 'Google', 'California');
Query OK, 1 row affected (0.04 sec)

mysql> insert into company values(102, 'TCS', 'India');
Query OK, 1 row affected (0.01 sec)

mysql> insert into company values(103, 'Amazon', 'Seattle');
Query OK, 1 row affected (0.01 sec)

mysql> insert into company values(104, 'CTS', 'Sholingnanallur');
Query OK, 1 row affected (0.01 sec)
```

```
mysql> insert into job values(101, 'Software engineer', 'Full time');
Query OK, 1 row affected (0.01 sec)

mysql> insert into job values(102, 'Data analyst', 'Internship');
Query OK, 1 row affected (0.00 sec)

mysql> insert into job values(103, 'Database developer', 'Full time');
Query OK, 1 row affected (0.04 sec)

mysql> insert into job values(104, 'Backend developer', 'Part time');
Query OK, 1 row affected (0.01 sec)
```

```
mysql> insert into application values(101, '2024-06-12', 'Applied');
Query OK, 1 row affected (0.01 sec)

mysql> insert into application values(102, '2024-04-13', 'Interviewing');
Query OK, 1 row affected (0.00 sec)

mysql> insert into application values(103, '2025-05-15', 'Offer');
Query OK, 1 row affected (0.01 sec)

mysql> insert into application values(104, '2025-07-11', 'Rejected');
Query OK, 1 row affected (0.01 sec)
```


OUTPUT SCREENSHOT

```
mysql> select * from company;
+-----+-----+-----+
| company_id | company_name | location |
+-----+-----+-----+
|          101 | Google       | California |
|          102 | TCS          | India      |
|          103 | Amazon       | Seattle    |
|          104 | CTS          | Sholinganallur |
+-----+-----+-----+
4 rows in set (0.01 sec)
```

```
mysql> select * from job;
+-----+-----+-----+
| job_id | job_title          | job_type |
+-----+-----+-----+
|      101 | Software engineer  | Full time |
|      102 | Data analyst       | Internship |
|      103 | Database developer | Full time |
|      104 | Backend developer  | Part time |
+-----+-----+-----+
4 rows in set (0.00 sec)
```

```
mysql> select * from application;
+-----+-----+-----+
| application_id | applied_date | application_status |
+-----+-----+-----+
|          101 | 2024-06-12   | Applied           |
|          102 | 2024-04-13   | Interviewing      |
|          103 | 2025-05-15   | Offer             |
|          104 | 2025-07-11   | Rejected          |
+-----+-----+-----+
4 rows in set (0.00 sec)
```

```
mysql> select * from application where application_status = 'Applied';
```

application_id	applied_date	application_status
101	2024-06-12	Applied

1 row in set (0.01 sec)

```
mysql> select * from application order by applied_date asc;
```

application_id	applied_date	application_status
102	2024-04-13	Interviewing
101	2024-06-12	Applied
103	2025-05-15	Offer
104	2025-07-11	Rejected

4 rows in set (0.01 sec)

```
mysql> update application set application_status = "Selected" where application_id = 104;
```

Query OK, 1 row affected (0.01 sec)

Rows matched: 1 Changed: 1 Warnings: 0

```
mysql> select * from application;
```

application_id	applied_date	application_status
101	2024-06-12	Applied
102	2024-04-13	Interviewing
103	2025-05-15	Offer
104	2025-07-11	Selected

4 rows in set (0.00 sec)

```
mysql> delete from application where application_id = 103;
```

Query OK, 1 row affected (0.01 sec)

```
mysql> select * from application;
```

application_id	applied_date	application_status
101	2024-06-12	Applied
102	2024-04-13	Interviewing
104	2025-07-11	Selected

3 rows in set (0.00 sec)

FUTURE ENHANCEMENTS

The Job Application Tracker can be enhanced in a number of ways to become more reliable, intelligent, and user-friendly as technology advances and the labour market becomes more competitive and dynamic. Here are some possible improvements in the future:

1. Integration with APIs and Job Portals

The system is currently maintained by hand. Through their APIs, the tracker may eventually be connected with websites like as Indeed, LinkedIn, or Naukri. This will decrease human labour and increase accuracy by enabling the automatic import of job postings, application revisions, and interview schedules.

2. Resume Auto-Fill and Parser

NLP approaches can be used to integrate a resume parser. The system has the ability to automatically fill in user profile fields like education, experience, and talents when a user submits a resume. This guarantees uniform formatting and saves time.

3. Recommendation Engine Driven by GenAI

Using Generative AI, the system is able to recommend:

Businesses based on the interests and capabilities of users

Positions in line with contemporary tendencies

Resuming enhancements with AI-generated recommendations

Additionally, chatbots can be used to assist users with inquiries such

as "What is the status of my latest application?" or "On the basis of my profile, suggest the top 5 companies to me."

4. Reminder and Notification System

By including an intelligent notification system (such as an SMS, email, or app notification), consumers can stay focused on:

Deadlines for applications

Schedules for interviews

Reminders for follow-up

This guarantees better time management and enhances the user experience.

5. Version of the Mobile App

A portable, lightweight version of the tracker might be created to make it easier to use. It will make the system more approachable and user-focused by assisting users in tracking and updating apps while they are on the road.

6. Dashboard for Analytics

One way to introduce an admin or user dashboard is with:

Application status pie charts (e.g., Applied, Interview, Offer)

Applications across time

Application to interview to offer conversion rates

Users can use these data to evaluate their success and enhance their

job search tactics.

7. Data encryption and security

Implementing authentication, authorization, and encryption capabilities is essential because the application may handle sensitive personal data, such as contact details and resumes. Data security and user trust are increased by utilizing technologies like OAuth2 and SSL.

8. Integration with the Cloud

Better scalability, availability, and performance are made possible by storing application data on the cloud (such as AWS, Azure, or Google Cloud). It will enable numerous users to safely track and access their data from any location.

CONCLUSION

The goal of the Job Application Tracker project was to make the job-search process easier and more structured for professionals and students. Through the development of this project, we produced a structured system that enables users to effectively and methodically save, manage, and track their job applications. In order to manage different things, such as users, businesses, job applications, and interviews, we used MySQL to construct a relational database during this project. Real-time operations were supported by the heavy usage of core database operations such as CREATE, INSERT, SELECT, UPDATE, and JOIN. This gave us practical expertise with application logic, query optimization, and database design.

To sum up, the Job Application Tracker project combines careful planning, realistic execution, and an optimistic outlook. It demonstrates how even a basic application can address real-world issues and provide value to a professional portfolio or resume when it is created with clarity and purpose. This project has contributed to the development of solid database basic knowledge, which is necessary for pursuing tech-related positions like database developer, data analyst, or backend developer.