

# **DATABASE DESIGN DOCUMENT – GoCareer**

## **1.Introduction**

### **1.1 Purpose**

This document provides the database design for the Job Search Website Management System, including the schema, relationships, indexing strategies, and data integrity constraints.

### **1.2 Scope**

The database will store and manage, job categories, jobs posting, , company name , job seeker detail , messages, and job application. It will ensure data integrity, security, and efficient retrieval to provide a seamless job search and recruitment experience.

### **1.3 Target Audience**

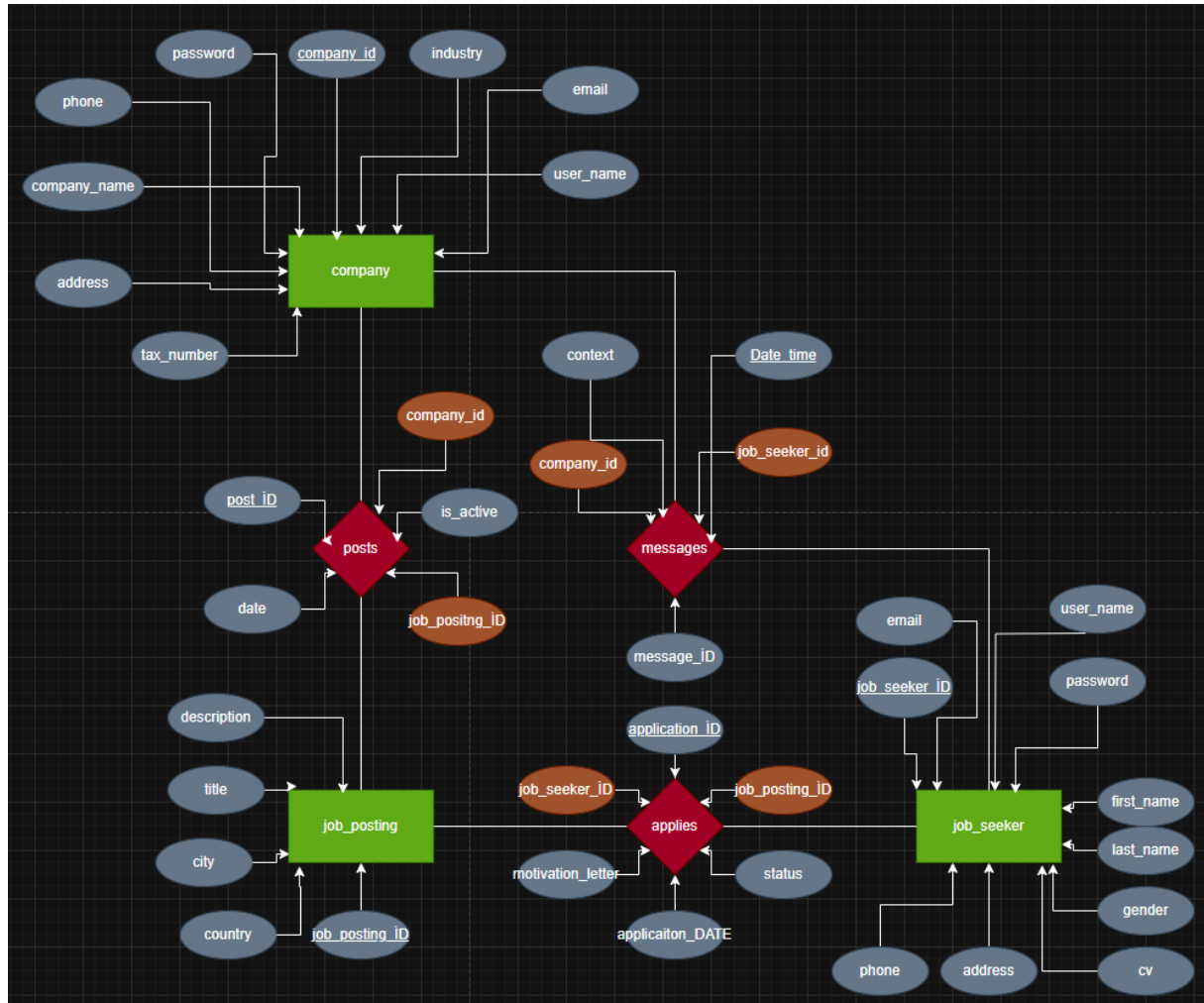
- **Job Seekers** → recent graduates, professionals who want to change jobs, freelancers and students looking for an internship.
- **Company & HR Professionals** → Companies, human resources managers and recruitment specialists.
- **Freelancers & Remote Workers** → Individuals looking for flexible or project-based jobs.
- **Career Consultants & Educational Institutions** → Career coaches and course providers who support job seekers.

## **2.Database Architecture**

### **2.1 Database Management System**

- a) **Type:** Relational Database Management System (RDBMS)
- b) **Preferred DBMS:** MySQL
- c) **Normalization:** Third Normal Form (3NF) to optimize storage and eliminate redundancy.

## ER DIAGRAM



## 3.Database Schema

### 3.1 Company Table

COLUMN	TYPE	CONSTRAINTS	DESCRIPTION
compnay_id	INT(PK)	AUTO_INCREMENT	EMPLOYERS ID
User_name	VARCHAR(255)	UNIQUE , NOT NULL	SYSTEM ACCOUNT NAME
company_name	VARCHAR(100)		EMPOYERS COMPANY NAME
industry	VARCHAR(100)		COMPANY'S INDUSTRY
password	VARCHAR	NOT NULL	PERSONEL ACCES THE SYSTEM
Tax_number	VARCHAR(20)	UNIQUE , NOT NULL	ALL COMPANY HAVE TAX NUMBER

Address	VARCHAR(100)	UNIQUE , NOT NULL	COMPANY'S ADDRESS
Email	VARCHAR(50)	UNIQUE, NOT NULL	COMPANY CONTACT EMAIL
Phone	VARCHAR(20)	UNIQUE ,NOT NULL	COMPANY CONTACT PHONE

### 3.2 Job\_seekers Table

COLUMN	TYPE	CONSTRAINTS	DESCRPTION
job_seeker_id	INT(PK)	AUTO_INCREMENT	JOB_SEEKER ID
user_name	INT(FOREIGN)	UNIQUE ,	SYSTEM ACCOUNT NAME
first_name	VARCHAR(255)	NOT NULL	JOB SEEKER NAME
Last_name	VARCHAR(255)	NOT NULL	JOB SEEKER SURNAME
password	VARCHAR(255)	NOT NULL	USER PASSWORD FOR THE ACCESS SYSTEM
gender	ENUM	("MALE","FEMALE")	JOB_SEEKER GENDER
Address	VARCHAR(100)	UNIQUE , NOT NULL	JOB_SEEKER ADDRESS
Email	VARCHAR(50)	UNIQUE, NOT NULL	JOB_SEEKER CONTACT EMAIL
Phone	VARCHAR(20)	UNIQUE ,NOT NULL	JOB_SEEKER CONTACT PHONE
cv	TEXT	NOT NULL	JOB SEEKER EXPERIENCE

### 3.3 Job Posting

COLUMN	TYPE	CONSTRAINTS	DESCRPTION
Job_posting_id	INT(PK)	AUTO_INCREMENT	JOB POSTING ID
Title	VARCHAR(100)	NOT NULL	POST'S TITLE
Descrpition	TEXT	NOT NULL	POST'S EXPLANATION ABOUT THE JOB
Country	VARCHAR(50)	NOT NULL	POST SHARED WHICH COUNTRY
City	VARCHAR(50)	NOT NULL	POST SHARED WHICH CITY

### 3.4 Post Table

COLUMN	TYPE	CONSTRAINTS	DESCRPTION
--------	------	-------------	------------

Post_id	INT(PK)	AUTO_INCREMENT	POST SHARED NUMBER
Company_id	INT(FOREIGN)	NOT NULL	WHICH COMPANY BELONG
title	VARCHAR(255)	NOT NULL	TITLE OF THE POST
date	DATE DEFAULT CURRENT DATE	NOT NULL	SHARED DATE TIME
is_active	ENUM	('active','not active')	POST STATUS

### 3.5 Applies Table

COLUMN	TYPE	CONSTRAINTS	DESCRIPTION
applies_id	INT(PK)	AUTO_INCREMENT	APPLICATION ID
job_posting_id	INT(FOREIGN)	NOT NULL	SHARED POST ID
job_seeker_id	INT(FOREIGN)	NOT NULL	JOB SEEKER ID
applies_date	DATE DEFAULT CURRENT DATE	NOT NULL	APPLICANT DATE TIME
status	ENUM	('accepted', 'rejected', 'hold')	APPLICANT STATUS
motivation_letter	TEXT		INTDORUCE THE JOB SEEKER HIMSELF/HERSELF

### 3.4 Messages Table

COLUMN	TYPE	CONSTRAINTS	DESCRIPTION
message_id	INT(PK)	AUTO_INCREMENT	MESSAGE ID
sender_id	INT	UNIQUE	WHO SEND A MESSAGE ID
receiver_id	INT	UNIQUE	WHO GIVE A MESSAGE ID
message_content	TEXT		WHAT IS MESSAGE INCLUDE?
Date_time	DATE	NOT NULL	WHEN MESSAGE SENT

## 4. Indexing Strategy

### a. Primary Keys

Primary keys are automatically indexed, ensuring fast lookups and enforcing uniqueness for each row. Every table has a primary key, which serves as its unique identifier. For example:

- COMPANY(company\_id), JOB\_SEEKER(job\_seeker\_id), JOB\_POSTING(job\_posting\_id), POST(post\_id), APPLIES(apply\_id), and MESSAGE(message\_id) all have automatically created primary key indexes.

#### **b. Foreign Key Indexing (Referential Integrity)**

Foreign keys establish relationships between tables and should be indexed to speed up JOIN operations. Since foreign keys are frequently used in queries that retrieve related data, indexing them improves query performance. For instance:

- POST(company\_id) is indexed to enable quick lookups of all posts by a particular company.
- POST(job\_posting\_id) is indexed to speed up job posting queries.
- APPLIES(job\_seeker\_id) is indexed to allow efficient retrieval of job applications made by a job seeker.
- APPLIES(job\_posting\_id) is indexed to enable quick lookups of applicants for a particular job posting.
- MESSAGE(sender\_id) and MESSAGE(receiver\_id) are indexed to improve message retrieval performance.

#### **c. Frequently Queried Columns**

Some columns are often used in search and filtering operations, so indexing them significantly improves performance. Examples include:

- email column in COMPANY and JOB\_SEEKER tables, which is used for authentication and needs fast lookups.
- status column in APPLIES, which allows for efficient filtering of applications based on their current status.
- is\_active column in POST, which helps retrieve active job postings quickly.

#### **d. Full-Text Indexing (Optimized Search)**

For text-heavy fields, FULLTEXT indexing is used to enhance search functionality, allowing efficient keyword-based searches instead of slower LIKE '%text%' queries. This is particularly useful for fields such as:

- description in JOB\_POSTING, enabling fast job searches by description.
- motivation\_letter in APPLIES, allowing employers to search applications based on motivation letters.
- content in MESSAGE, making it easier to search within chat histories.

## 5. Security Measures

### a. Data Encryption

- **Password Hashing:** User passwords are stored securely using bcrypt hashing to prevent unauthorized access.
- **Sensitive Data Encryption:** Critical user data (e.g., phone numbers, emails) can be encrypted using AES-256 encryption to protect it from breaches.

### b. Access Control & Authentication

- **Role-Based Access Control (RBAC):** Users are assigned roles (company, job\_seeker), and permissions are strictly enforced to prevent unauthorized access.

### c. Audit Logging & Monitoring

- **Action Logging:** User actions (e.g., logins, job applications, message exchanges) are logged to detect suspicious activities.

## 6. Scalability and Performance Optimizations

### a. Read Replication (Scaling Read Queries)

- **Tables:** JOB\_POSTING, COMPANY, JOB\_SEEKER
- **Reason:** These tables are frequently queried but not updated as often, making them ideal for read replicas.

### b. Caching (Reducing Query Load)

- **Tables:** JOB\_POSTING, COMPANY, MESSAGE
- **Reason:** Frequently accessed data, such as job listings, company profiles, and unread messages, can be cached for faster retrieval.

### c. Indexing Optimization (Faster Lookups)

- **Tables:** COMPANY, JOB\_SEEKER, APPLIES, MESSAGE, POST
- **Reason:**
  - Indexing email and phone\_number in COMPANY and JOB\_SEEKER speeds up authentication.
  - Indexing status in APPLIES helps track job application progress.
  - Indexing sender\_id and receiver\_id in MESSAGE speeds up chat queries.
  - Indexing is\_active in POST allows quick retrieval of active job postings.

## 7.Backup and Diaster Recovery Plan

- **Database Backup:** Daily full backups, hourly incremental backups, and encrypted offsite storage ensure data safety.
- **Disaster Recovery:** Multi-region deployment and automated failover minimize downtime.
- **Recovery Objectives:** RTO < 15 min, RPO < 5 min to quickly restore system functionality.
- **File Storage Backup:** Version-controlled and automatic snapshots prevent data loss

## 8.Conculusion

The database design for the Job Search Website Management System is structured to ensure efficiency, security, and scalability. By implementing a relational database model (MySQL) and following Third Normal Form (3NF) principles, the design optimizes data storage while minimizing redundancy.

- Well-defined schema covering users, employers, job seekers, job postings, applications, messages, notifications, and experience records.
- Efficient indexing strategies, including primary keys, foreign keys, and frequently queried columns, to enhance query performance.
- Security measures, such as password hashing, data encryption, and role-based access control (RBAC), to protect user data.