****

**NEW ERA UNIVERSITY COLLEGE**

|  |
| --- |
| Faculty of Computer Science & Information Computing Technology |
| Department of Information Computing Technology |

**Report of Human Resource Management System**

**TIST223 DATABASE CONCEPTS**

**GROUP ASSIGNMENT**

**Group 2:**

**ON WU XU 2290113-DCS**

**YAP KAH SENG 22900146-DCS**

**CHONG JIA WEI 2290206-DCS**

**LECTURER: TS. LAW TENG YI**

**DATE OF SUBMISSION: 19 August 2023**

**CONTENTS**

**1.0 Introdction 3**

**2.0 SQL Commands 6**

**2.1 List of Table, Attribute and Data 6**

**2.2 Data Definition Language (DDL) 15**

2.2.1 CREATE 15

2.2.2 ALTER 79

2.2.3 RENAME 138

**2.3 Data Manipulation Language (DML) 139**

2.3.1 INSERT 139

2.3.2 UPDATE 203

2.3.3 DELETE 217

2.3.4 SELECT 219

**3.0 Diagrams 226**

**3.1 ER Diagram 226**

**3.2 Table Relationship 227**

**3.3 Data Flow Diagram 230**

**3.4 Chen ER Diagram 231**

**3.5 Mock-ups 235**

1. **Introduction**

**1.1 Overview**

In the dynamic landscape of modern businesses, the efficient management of Human Resources (HR) plays a pivotal role in driving organizational success. A well-structured HR Management System is a fundamental tool that empowers businesses to streamline HR operations, enhance employee engagement, and optimize workforce productivity. This introduction provides an overview of the HR Management System, highlighting its essential features and benefits for businesses of all sizes.

The HR Management System described here is a comprehensive and integrated platform that centralizes all HR-related processes and data. It serves as a one-stop solution for HR professionals, enabling them to manage various processes of the employees seamlessly. From recruitment and onboarding to performance evaluation, talent development, and offboarding, the system covers the entire spectrum of HR functions.

At its core, the system maintains a database of individual employee profiles, housing essential details such as personal information, contact details, employment history, job roles, and responsibilities. This comprehensive employee database forms the foundation for all HR activities and facilitates quick access to crucial information when needed.

Recruitment and onboarding processes are significantly streamlined through the HR Management System. It allows HR teams to post job openings, receive applications, and track candidate progress throughout the selection process. Additionally, the system can store candidate resumes, interview feedback, and skill assessments, making it easier to identify the best-suited candidates for vacant positions.

Once employees are onboard, the system takes charge of attendance management, leave tracking, and time-off approvals. This ensures that records are accurate, and employee absences are managed efficiently. Furthermore, the system can handle employee benefits, such as health insurance, retirement plans, and other perks, fostering a positive and supportive work environment.

Performance evaluation and talent development are vital aspects of HR management. The HR Management System enables HR professionals to conduct performance reviews, set development goals, and track progress over time. This data-driven approach allows for better insights into employee growth and assists in identifying high-potential talents for career advancement.

Moreover, the system supports learning and development initiatives by maintaining a record of training programs attended by employees. This information helps HR teams to identify skill gaps and plan targeted training sessions to upskill the workforce.

A critical feature of the HR Management System is its compliance with labour laws and regulations. It helps HR professionals stay updated with changing legal requirements, maintain accurate records, and ensure adherence to employment laws.

One of the fundamental steps to access the HR Management System is through a secure login process. To ensure data privacy and system security, each user, such as HR managers, administrators, and employees, is provided with unique login credentials. These credentials typically consist of a username and password.

For HR managers and administrator, they need to visit the HR Management System website or access the system through an internal company portal. Then, enter the unique username and password provided by the system administrator or IT department. Upon successful authentication, the HR manager gains access to the system's comprehensive dashboard, featuring various modules to handle HR tasks.

For employees, they can access the HR Management System through a dedicated employee portal or mobile application. Then, they are required to enter their individual login credentials, usually consisting of a username and password. After successful authentication, employees can view and manage their personal information, request time off, access company policies, and participate in performance evaluations.

Once logged in, users are presented with an intuitive and user-friendly dashboard that provides an overview of essential HR information. Such as their name, gender, birth date, phone number, department id and position id.

The navigation within the HR Management System is designed to be straightforward and efficient. Users can easily access different modules with their position.

In conclusion, the HR Management System serves as a strategic tool for businesses to nurture their most valuable asset: their employees. By streamlining HR processes, fostering employee engagement, and promoting talent development, the system contributes to creating a motivated and high-performing workforce. As businesses continue to recognize the significance of human capital, adopting an efficient HR Management System becomes an indispensable choice for achieving long-term growth and success.

1. **SQL Commands**

**2.1 List of Table, Attributes and Data**

- Here is a list of the table names and attributes in the create table statement you provided:

1. Attendance\_Record

· attendance\_record\_id (Primary Key)

· record\_date

· time\_in

· time\_out

· OT\_Record\_id

2. Candidates

· Candidate\_id (Primary Key)

· Candidate\_name

· Email

· phone\_no

· Skill\_set\_id

· position\_Opening\_id

3. Contracts

· contract\_id (Primary Key)

· emp\_files\_id

· start\_date

· end\_date

· types

· status

4. Departments

· dept\_id (Primary Key)

· dept\_name

· dept\_manager

· location

· emp\_count

5. Edu\_History

· edu\_history\_id (Primary Key)

· emp\_files\_id

· school\_name

· major

· LEVEL

· graduate\_date

6. Emp\_Penalites

· emp\_penality\_id (Primary Key)

· emp\_files\_id

· TYPE

· reason

· DATE

· amount

7. Employee\_Attendance

· employee\_Attendance\_id (Primary Key)

· Leave\_Record\_id

· Leave\_req\_id

· OT\_Record\_id

· attendance\_record\_id

8. Employee\_Awards

· Employee\_Awards\_id (Primary Key)

· Employee\_Benefits\_id

· Title

· Reason

· date

· Amount

9. Employee\_Benefits

· Employee\_Benefits\_id (Primary Key)

· medical\_insurance

· sick\_leave

· vision\_insurance

· maternity\_leave

10. Employee\_Files

· emp\_files\_id (Primary Key)

· file\_name

· file\_size

· file\_path

· upload\_date

11. Employee\_Health\_Checkups

· Employee\_Health\_Checkups\_id (Primary Key)

· emp\_files\_id

· Check\_date

· height

· weight

· blood\_pressure

12. Employee\_Performance\_and\_Development

· emp\_pnd\_id (Primary Key)

· skillset\_id

· employee\_promotion\_id

· emp\_training\_id

· credit\_score

· credit\_score\_grader\_id

13. Employee\_Promotion

· emp\_promo\_id (Primary Key)

· promo\_reason

· current\_position\_id

· promoted\_position\_id

· promoted\_date

14. Employee\_Training

· emp\_training\_id (Primary Key)

· training\_skill\_id

· trainer\_id

· training\_date

· training\_duration

15. Employees

· emp\_id (Primary Key)

· name

· gender

· birth\_date

· phone\_no

· dept\_id

· position\_id

16. Employees\_Salary\_Details

· sal\_detail\_id (Primary Key)

· Employee\_Benefits\_id

· BASE\_SALARY

· BONUS

· ALLOWANCE

· DEDUCTIONS

· BENEFITS

17. Interview\_List

· Interview\_list\_id (Primary Key)

· Candidate\_id

· Interviewer\_id

· Interview\_date

· Interview\_result

18. Leave\_Record

· leave\_record\_id (Primary Key)

· leave\_req\_id

· leave\_start\_date

· leave\_end\_date

· leave\_approver\_id

19. Leave\_Request

· leave\_req\_id (Primary Key)

· requester\_id

· req\_reason

· req\_start\_date

· req\_end\_date

· approve\_status

20. OT\_Record

· OT\_Record\_id (Primary Key)

· record\_date

· time\_in

· time\_out

· ot\_hour

21. Position\_Application

· position\_apply\_id (Primary Key)

· emp\_id

· current\_position\_id

· position\_opening\_id

· application\_reason

· skillset\_id

22. Position\_Opening

· Position\_Opening\_id (Primary Key)

· Position\_id

· Dept\_id

· Experience\_Required

· Education\_Required

23. Positions

· position\_id (Primary Key)

· position\_name

· avg\_salary

· dept\_id

· emp\_count

24. Project\_Participation

· project\_participate\_id (Primary Key)

· emp\_pnd\_id

· project\_id

· comment

· start\_date

· end\_date

25. Projects

· project\_id (Primary Key)

· project\_title

· leader\_id

· start\_date

· end\_date

· comment

26. Resignations

· resign\_emp\_id (Primary Key)

· resign\_reason

· RESIGN\_DATE

· related\_contract\_id

· exit\_interviewer\_id

27. Salary\_History

· SAL\_HISTORY\_ID (Primary Key)

· Employee\_Benefits\_id

· gender

· TOTAL\_SALARY

· START\_DATE

· END\_DATE

28. Skills

· Skill\_id (Primary Key)

· Skill\_name

· dept\_id

· skill\_resources

· skill\_description

29. Skillset

· Skillset\_id (Primary Key)

· emp\_pnd\_id

· candidate\_id

· Skill\_one\_id

· Skill\_two\_id

· Skill\_three\_id

30. Users

· user\_id (Primary Key)

· emp\_id

· username

· password

· permission\_type

· description

· creat\_date

· last\_update\_date

31. Work\_Exp

· work\_exp\_id (Primary Key)

· emp\_files\_id

· company\_name

· POSITION

· start\_date

· end\_date

**2.2 Data Definition Language (DDL)**

**2.2.1 CREATE**

- The CREATE statement in SQL is used to create new database objects, such as tables, views, indexes, and procedures. It allows you to define the structure and characteristics of these objects before adding data or performing operations on them.

- For example, when creating a table, you specify the column names, data types, constraints, and other properties. Similarly, when creating a view, you define a virtual table that retrieves data from one or more tables based on a specified query.

- The CREATE statement is essential for setting up the initial structure of the database and plays a fundamental role in defining the schema and organization of data within the database.

- This Human Resource Management System contains 31 tables. Through these tables, the system can track employees' basic information, work status and performance evaluation in real time, providing comprehensive human resource management and decision support for enterprises.

**CREATE DATABASE**

- Create a database for use in Human Resources Management System. Before create the tables, use the CREATE TABLE Commandss “USE database\_name” and SELECT DATABASE() to ensure the tables are created in the correct database.

**CREATE TABLE Commands:**

CREATE DATABASE human\_resources\_management\_system;

USE human\_resources\_management\_system;

**Excuted Result (with Query Commands):**

(Excute successfully)

****

****

SELECT DATABASE();



**CREATE TABLES**

**CREATE TABLE Commands:**

CREATE TABLE `attendance\_record` (

`attendance\_record\_id` int NOT NULL AUTO\_INCREMENT,

`record\_date` date NULL DEFAULT NULL,

`time\_in` time NULL DEFAULT NULL,

`time\_out` time NULL DEFAULT NULL,

`OT\_Record\_id` int NULL DEFAULT NULL,

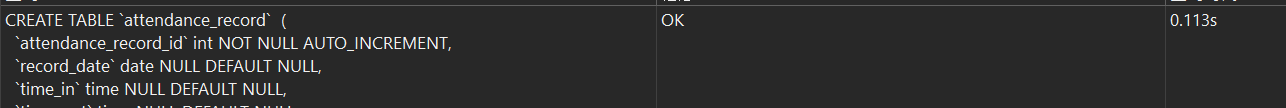
PRIMARY KEY (`attendance\_record\_id`) USING BTREE,

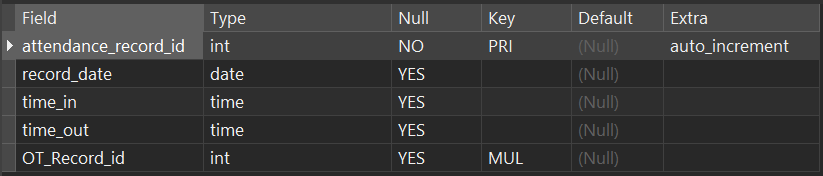
INDEX `FK\_Attendance\_Record\_OT\_Record\_id`(`OT\_Record\_id` ASC) USING BTREE

) ENGINE = InnoDB AUTO\_INCREMENT = 1 CHARACTER SET = utf8mb4 COLLATE = utf8mb4\_0900\_ai\_ci ROW\_FORMAT = Dynamic;

**Excuted Result (with Query Commands):**

(Excute successfully)



DESC attendance\_record;

SELECT \* FROM attendance\_record;

**CREATE TABLE Commands:**

CREATE TABLE `candidates` (

`Candidate\_id` int NOT NULL AUTO\_INCREMENT,

`Candidate\_name` varchar(500) CHARACTER SET utf8mb4 COLLATE utf8mb4\_0900\_ai\_ci NULL DEFAULT NULL,

`Email` varchar(500) CHARACTER SET utf8mb4 COLLATE utf8mb4\_0900\_ai\_ci NULL DEFAULT NULL,

`phone\_no` varchar(500) CHARACTER SET utf8mb4 COLLATE utf8mb4\_0900\_ai\_ci NULL DEFAULT NULL,

`Skill\_set\_id` int NULL DEFAULT NULL,

`position\_Opening\_id` int NULL DEFAULT NULL,

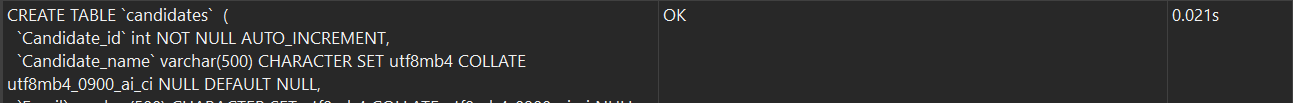
PRIMARY KEY (`Candidate\_id`) USING BTREE,

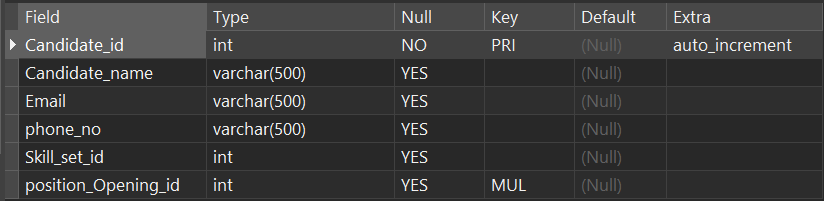
INDEX `FK\_CANDIDATES\_POSITION\_OPENING\_ID`(`position\_Opening\_id` ASC) USING BTREE

) ENGINE = InnoDB AUTO\_INCREMENT = 1 CHARACTER SET = utf8mb4 COLLATE = utf8mb4\_0900\_ai\_ci ROW\_FORMAT = DYNAMIC;

**Excuted Result (with Query Commands):**

(Excute successfully)

****

DESC candidates;

SELECT \* FROM candidates;

**CREATE TABLE Commands:**

CREATE TABLE `contracts` (

`contract\_id` int NOT NULL AUTO\_INCREMENT,

`emp\_files\_id` int NULL DEFAULT NULL,

`start\_date` date NULL DEFAULT NULL,

`end\_date` date NULL DEFAULT NULL,

`types` varchar(50) CHARACTER SET utf8mb4 COLLATE utf8mb4\_0900\_ai\_ci NULL DEFAULT NULL,

`status` varchar(50) CHARACTER SET utf8mb4 COLLATE utf8mb4\_0900\_ai\_ci NULL DEFAULT NULL,

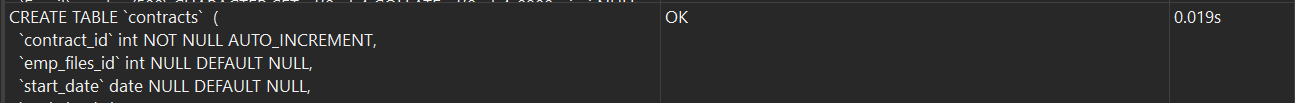
PRIMARY KEY (`contract\_id`) USING BTREE,

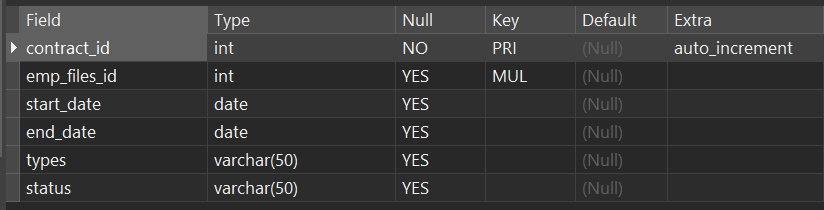
INDEX `FK\_contracts\_emp\_file\_id`(`emp\_files\_id` ASC) USING BTREE

) ENGINE = InnoDB AUTO\_INCREMENT = 1 CHARACTER SET = utf8mb4 COLLATE = utf8mb4\_0900\_ai\_ci ROW\_FORMAT = DYNAMIC;

**Excuted Result (with Query Commands):**

(Excute successfully)

****

DESC contracts;

SELECT \* FROM contracts;

**CREATE TABLE Commands:**

CREATE TABLE `departments` (

`dept\_id` int NOT NULL AUTO\_INCREMENT,

`dept\_name` varchar(500) CHARACTER SET utf8mb4 COLLATE utf8mb4\_0900\_ai\_ci NULL DEFAULT NULL,

`dept\_manager` int NULL DEFAULT NULL,

`location` varchar(500) CHARACTER SET utf8mb4 COLLATE utf8mb4\_0900\_ai\_ci NULL DEFAULT NULL,

`emp\_count` int NULL DEFAULT NULL,

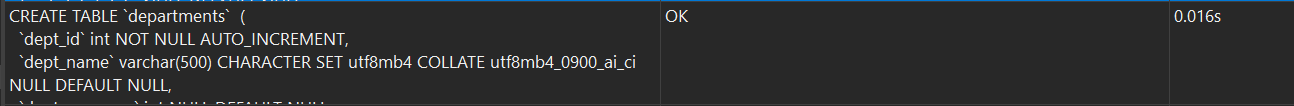
PRIMARY KEY (`dept\_id`) USING BTREE,

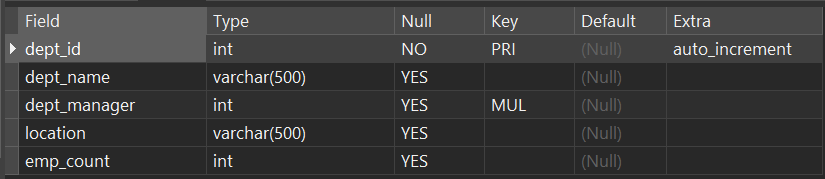
INDEX `FK\_departments\_dept\_manager`(`dept\_manager` ASC) USING BTREE

) ENGINE = InnoDB AUTO\_INCREMENT = 1 CHARACTER SET = utf8mb4 COLLATE = utf8mb4\_0900\_ai\_ci ROW\_FORMAT = DYNAMIC;

**Excuted Result (with Query Commands):**

(Excute successfully)



DESC departments;

SELECT \* FROM departments;

**CREATE TABLE Commands:**

CREATE TABLE `edu\_history` (

`edu\_history\_id` int NOT NULL AUTO\_INCREMENT,

`emp\_files\_id` int NULL DEFAULT NULL,

`school\_name` varchar(50) CHARACTER SET utf8mb4 COLLATE utf8mb4\_0900\_ai\_ci NULL DEFAULT NULL,

`major` varchar(50) CHARACTER SET utf8mb4 COLLATE utf8mb4\_0900\_ai\_ci NULL DEFAULT NULL,

`LEVEL` varchar(50) CHARACTER SET utf8mb4 COLLATE utf8mb4\_0900\_ai\_ci NULL DEFAULT NULL,

`graduate\_date` date NULL DEFAULT NULL,

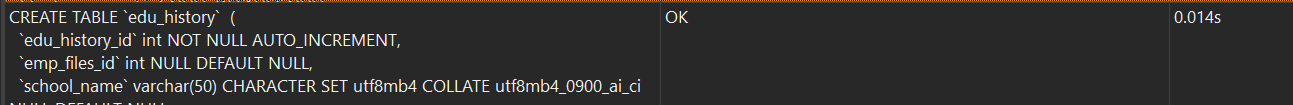
PRIMARY KEY (`edu\_history\_id`) USING BTREE,

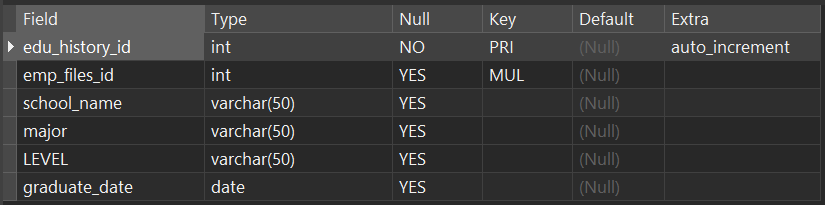
INDEX `FK\_Edu\_History\_emp\_files\_id`(`emp\_files\_id` ASC) USING BTREE

) ENGINE = InnoDB AUTO\_INCREMENT = 1 CHARACTER SET = utf8mb4 COLLATE = utf8mb4\_0900\_ai\_ci ROW\_FORMAT = DYNAMIC;

**Excuted Result (with Query Commands):**

(Excute successfully)



DESC edu\_history;

SELECT \* FROM edu\_history;

**CREATE TABLE Commands:**

CREATE TABLE `emp\_penalites` (

`emp\_penality\_id` int NOT NULL AUTO\_INCREMENT,

`emp\_files\_id` int NULL DEFAULT NULL,

`TYPE` varchar(50) CHARACTER SET utf8mb4 COLLATE utf8mb4\_0900\_ai\_ci NULL DEFAULT NULL,

`reason` varchar(50) CHARACTER SET utf8mb4 COLLATE utf8mb4\_0900\_ai\_ci NULL DEFAULT NULL,

`DATE` date NULL DEFAULT NULL,

`amount` int NULL DEFAULT NULL,

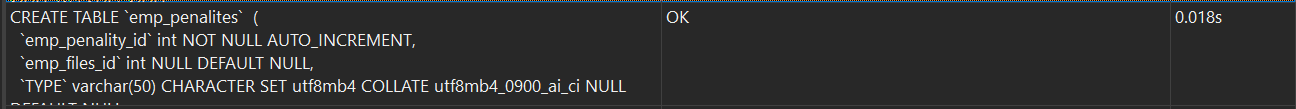
PRIMARY KEY (`emp\_penality\_id`) USING BTREE,

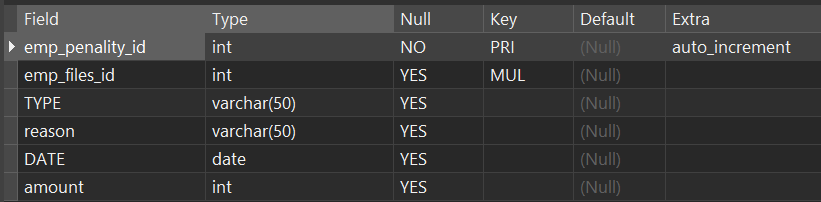
INDEX `FK\_Emp\_Penalites\_emp\_files\_id`(`emp\_files\_id` ASC) USING BTREE

) ENGINE = InnoDB AUTO\_INCREMENT = 1 CHARACTER SET = utf8mb4 COLLATE = utf8mb4\_0900\_ai\_ci ROW\_FORMAT = DYNAMIC;

**Excuted Result (with Query Commands):**

(Excute successfully)



DESC emp\_penalites;

SELECT \* FROM emp\_penalites;

**CREATE TABLE Commands:**

CREATE TABLE `employee\_attendance` (

`employee\_Attendance\_id` int NOT NULL AUTO\_INCREMENT,

`Leave\_Record\_id` int NULL DEFAULT NULL,

`Leave\_req\_id` int NULL DEFAULT NULL,

`OT\_Record\_id` int NULL DEFAULT NULL,

`attendance\_record\_id` int NULL DEFAULT NULL,

PRIMARY KEY (`employee\_Attendance\_id`) USING BTREE,

INDEX `FK\_employee\_Attendance\_Leave\_Record\_id`(`Leave\_Record\_id` ASC) USING BTREE,

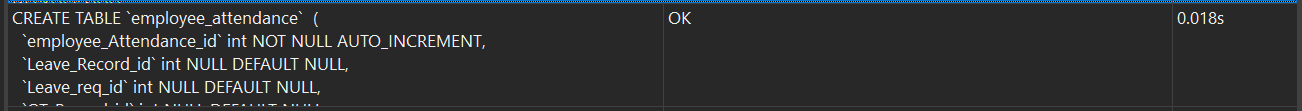
INDEX `FK\_employee\_Attendance\_Leave\_req\_id`(`Leave\_req\_id` ASC) USING BTREE,

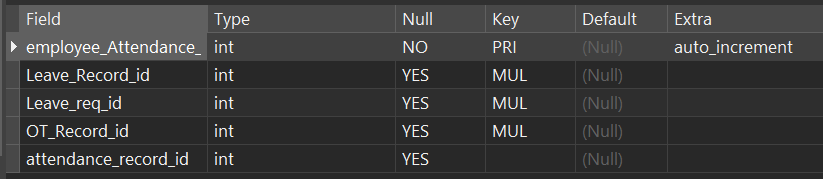
INDEX `FK\_employee\_attendance\_OT\_record\_id`(`OT\_Record\_id` ASC) USING BTREE

) ENGINE = InnoDB AUTO\_INCREMENT = 1 CHARACTER SET = utf8mb4 COLLATE = utf8mb4\_0900\_ai\_ci ROW\_FORMAT = DYNAMIC;

**Excuted Result (with Query Commands):**

(Excute successfully)



DESC employee\_attendance;

SELECT \* FROM employee\_attendance;

**CREATE TABLE Commands:**

CREATE TABLE `employee\_awards` (

`Employee\_Awards\_id` int NOT NULL AUTO\_INCREMENT,

`Employee\_Benefits\_id` int NOT NULL,

`Title` varchar(500) CHARACTER SET utf8mb4 COLLATE utf8mb4\_0900\_ai\_ci NULL DEFAULT NULL,

`Reason` varchar(500) CHARACTER SET utf8mb4 COLLATE utf8mb4\_0900\_ai\_ci NULL DEFAULT NULL,

`date` date NULL DEFAULT NULL,

`Amount` int NULL DEFAULT NULL,

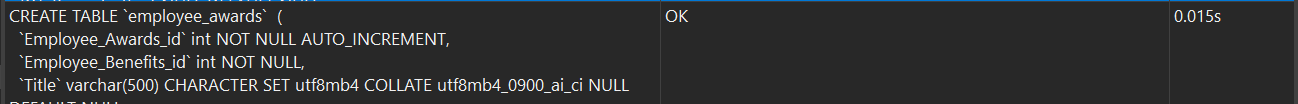
PRIMARY KEY (`Employee\_Awards\_id`) USING BTREE,

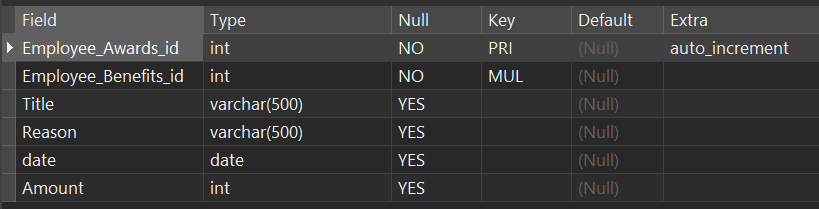
INDEX `Employee\_Benefits\_id`(`Employee\_Benefits\_id` ASC) USING BTREE

) ENGINE = InnoDB AUTO\_INCREMENT = 1 CHARACTER SET = utf8mb4 COLLATE = utf8mb4\_0900\_ai\_ci ROW\_FORMAT = DYNAMIC;

**Excuted Result (with Query Commands):**

(Excute successfully)



DESC employee\_awards;

SELECT \* FROM employee\_awards;

**CREATE TABLE Commands:**

CREATE TABLE `employee\_benefits` (

`Employee\_Benefits\_id` int NOT NULL AUTO\_INCREMENT,

`medical\_insurance` tinyint NULL DEFAULT NULL,

`sick\_leave` tinyint NULL DEFAULT NULL,

`vision\_insurance` tinyint NULL DEFAULT NULL,

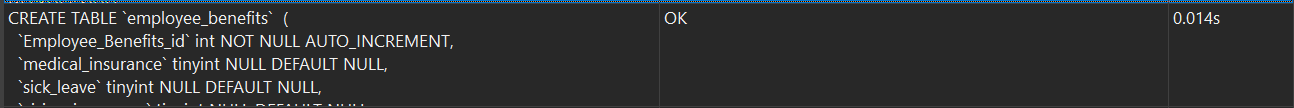
`maternity\_leave` tinyint NULL DEFAULT NULL,

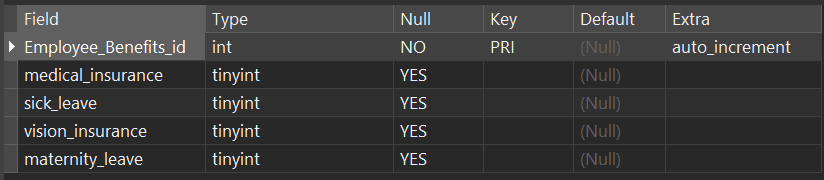
PRIMARY KEY (`Employee\_Benefits\_id`) USING BTREE

) ENGINE = InnoDB AUTO\_INCREMENT = 1 CHARACTER SET = utf8mb4 COLLATE = utf8mb4\_0900\_ai\_ci ROW\_FORMAT = DYNAMIC;

**Excuted Result (with Query Commands):**

(Excute successfully)



DESC employee\_benefits;

SELECT \* FROM employee\_benefits;

**CREATE TABLE Commands:**

CREATE TABLE `employee\_files` (

`emp\_files\_id` int NOT NULL AUTO\_INCREMENT,

`file\_name` varchar(500) CHARACTER SET utf8mb4 COLLATE utf8mb4\_0900\_ai\_ci NULL DEFAULT NULL,

`file\_size` varchar(500) CHARACTER SET utf8mb4 COLLATE utf8mb4\_0900\_ai\_ci NULL DEFAULT NULL,

`file\_path` varchar(500) CHARACTER SET utf8mb4 COLLATE utf8mb4\_0900\_ai\_ci NULL DEFAULT NULL,

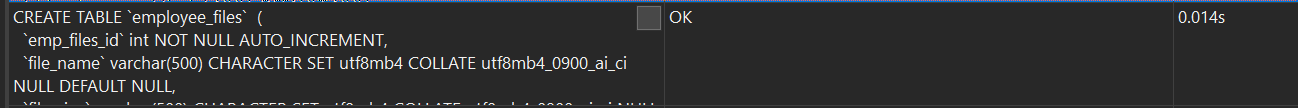
`upload\_date` date NULL DEFAULT NULL,

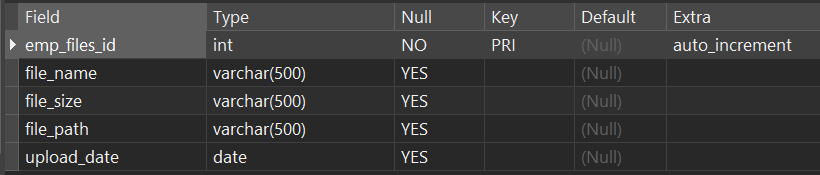
PRIMARY KEY (`emp\_files\_id`) USING BTREE

) ENGINE = InnoDB AUTO\_INCREMENT = 1 CHARACTER SET = utf8mb4 COLLATE = utf8mb4\_0900\_ai\_ci ROW\_FORMAT = DYNAMIC;

**Excuted Result (with Query Commands):**

(Excute successfully)



DESC employee\_files;

SELECT \* FROM employee\_files; 

**CREATE TABLE Commands:**

CREATE TABLE `employee\_health\_checkups` (

`Employee\_Health\_Checkups\_id` int NOT NULL AUTO\_INCREMENT,

`emp\_files\_id` int NULL DEFAULT NULL,

`Check\_date` date NULL DEFAULT NULL,

`height` varchar(500) CHARACTER SET utf8mb4 COLLATE utf8mb4\_0900\_ai\_ci NULL DEFAULT NULL,

`weight` varchar(500) CHARACTER SET utf8mb4 COLLATE utf8mb4\_0900\_ai\_ci NULL DEFAULT NULL,

`blood\_pressure` varchar(500) CHARACTER SET utf8mb4 COLLATE utf8mb4\_0900\_ai\_ci NULL DEFAULT NULL,

PRIMARY KEY (`Employee\_Health\_Checkups\_id`) USING BTREE,

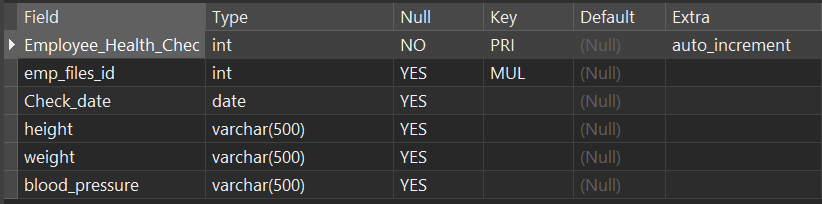
INDEX `FK\_Employee\_Health\_Checkups\_emp\_files\_id`(`emp\_files\_id` ASC) USING BTREE

) ENGINE = InnoDB AUTO\_INCREMENT = 1 CHARACTER SET = utf8mb4 COLLATE = utf8mb4\_0900\_ai\_ci ROW\_FORMAT = DYNAMIC;

**Excuted Result (with Query Commands):**

(Excute successfully)



DESC employee\_health\_checkups;

SELECT \* FROM employee\_health\_checkups;

**CREATE TABLE Commands:**

CREATE TABLE `employee\_performance\_and\_development` (

`emp\_pnd\_id` int NOT NULL AUTO\_INCREMENT,

`skillset\_id` int NULL DEFAULT NULL,

`employee\_promotion\_id` int NULL DEFAULT NULL,

`emp\_training\_id` int NULL DEFAULT NULL,

`credit\_score` int NULL DEFAULT NULL,

`credit\_score\_grader\_id` int NULL DEFAULT NULL,

PRIMARY KEY (`emp\_pnd\_id`) USING BTREE,

INDEX `skillset\_id`(`skillset\_id` ASC) USING BTREE,

INDEX `employee\_promotion\_id`(`employee\_promotion\_id` ASC) USING BTREE,

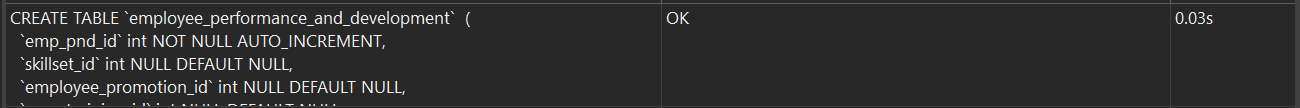
INDEX `emp\_training\_id`(`emp\_training\_id` ASC) USING BTREE,

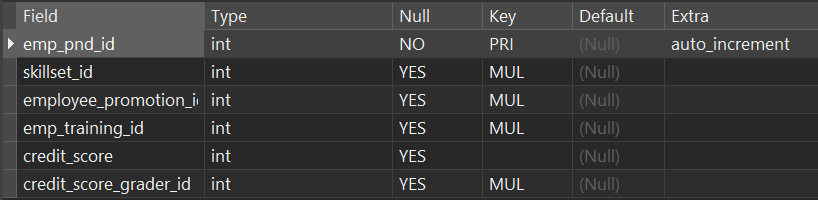
INDEX `credit\_score\_grader\_id`(`credit\_score\_grader\_id` ASC) USING BTREE

) ENGINE = InnoDB AUTO\_INCREMENT = 1 CHARACTER SET = utf8mb4 COLLATE = utf8mb4\_0900\_ai\_ci ROW\_FORMAT = DYNAMIC;

**Excuted Result (with Query Commands):**

(Excute successfully)



DESC employee\_performance\_and\_development;

SELECT \* FROM employee\_performance\_and\_development; 

**CREATE TABLE Commands:**

CREATE TABLE `employee\_promotion` (

`emp\_promo\_id` int NOT NULL AUTO\_INCREMENT,

`promo\_reason` varchar(500) CHARACTER SET utf8mb4 COLLATE utf8mb4\_0900\_ai\_ci NULL DEFAULT NULL,

`current\_position\_id` int NOT NULL,

`promoted\_position\_id` int NOT NULL,

`promoted\_date` date NULL DEFAULT NULL,

PRIMARY KEY (`emp\_promo\_id`) USING BTREE,

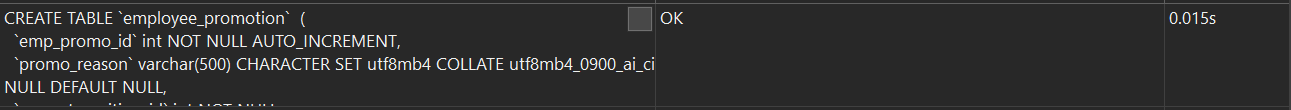
INDEX `current\_position\_id`(`current\_position\_id` ASC) USING BTREE,

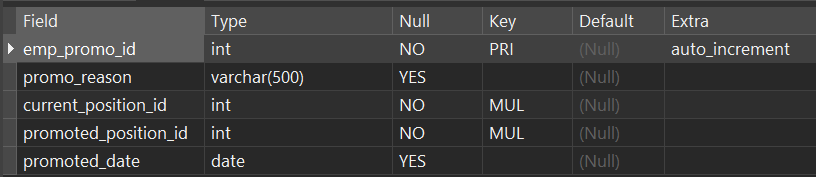
INDEX `promoted\_position\_id`(`promoted\_position\_id` ASC) USING BTREE

) ENGINE = InnoDB AUTO\_INCREMENT = 1 CHARACTER SET = utf8mb4 COLLATE = utf8mb4\_0900\_ai\_ci ROW\_FORMAT = DYNAMIC;

**Excuted Result (with Query Commands):**

(Excute successfully)



DESC employee\_promotion;

SELECT \* FROM employee\_promotion; 

**CREATE TABLE Commands:**

CREATE TABLE `employee\_training` (

`emp\_training\_id` int NOT NULL AUTO\_INCREMENT,

`training\_skill\_id` int NULL DEFAULT NULL,

`trainer\_id` int NULL DEFAULT NULL,

`training\_date` date NULL DEFAULT NULL,

`training\_duration` varchar(500) CHARACTER SET utf8mb4 COLLATE utf8mb4\_0900\_ai\_ci NULL DEFAULT NULL,

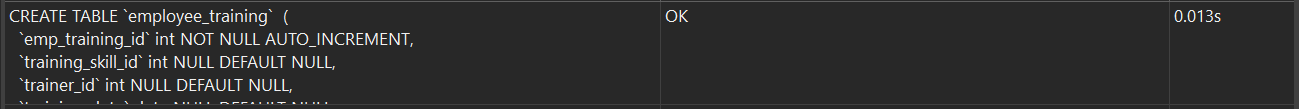
PRIMARY KEY (`emp\_training\_id`) USING BTREE,

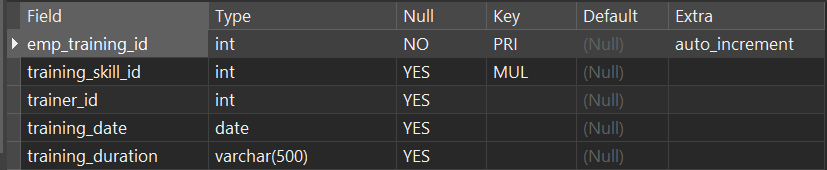
INDEX `training\_skill\_id`(`training\_skill\_id` ASC) USING BTREE

) ENGINE = InnoDB AUTO\_INCREMENT = 1 CHARACTER SET = utf8mb4 COLLATE = utf8mb4\_0900\_ai\_ci ROW\_FORMAT = DYNAMIC;

**Excuted Result (with Query Commands):**

(Excute successfully)



DESC employee\_training;

SELECT \* FROM employee\_training; 

**CREATE TABLE Commands:**

CREATE TABLE `employees` (

`emp\_id` int NOT NULL AUTO\_INCREMENT,

`name` varchar(500) CHARACTER SET utf8mb4 COLLATE utf8mb4\_0900\_ai\_ci NULL DEFAULT NULL,

`gender` varchar(10) CHARACTER SET utf8mb4 COLLATE utf8mb4\_0900\_ai\_ci NULL DEFAULT NULL,

`birth\_date` date NULL DEFAULT NULL,

`phone\_no` varchar(500) CHARACTER SET utf8mb4 COLLATE utf8mb4\_0900\_ai\_ci NULL DEFAULT NULL,

`dept\_id` int NULL DEFAULT NULL,

`position\_id` int NULL DEFAULT NULL,

PRIMARY KEY (`emp\_id`) USING BTREE,

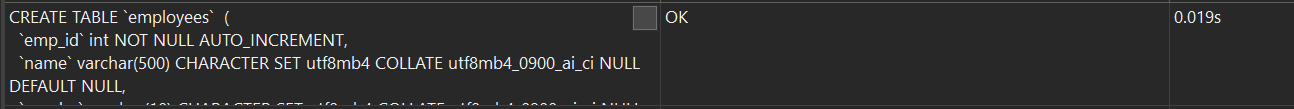
INDEX `FK\_EMPLOYEES\_DEPT\_ID`(`dept\_id` ASC) USING BTREE,

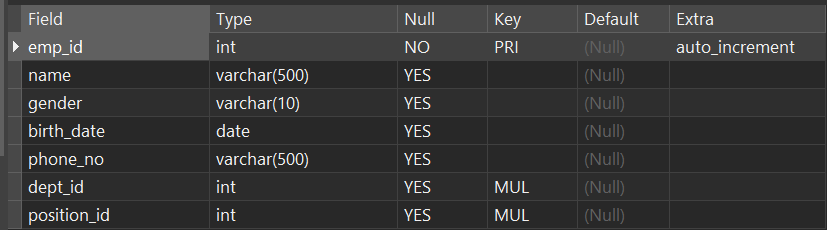
INDEX `FK\_EMPLOYEES\_POSITION\_ID`(`position\_id` ASC) USING BTREE

) ENGINE = InnoDB AUTO\_INCREMENT = 1 CHARACTER SET = utf8mb4 COLLATE = utf8mb4\_0900\_ai\_ci ROW\_FORMAT = DYNAMIC;

**Excuted Result (with Query Commands):**

(Excute successfully)



DESC employees;

SELECT \* FROM employees;

**CREATE TABLE Commands:**

CREATE TABLE `employees\_salary\_details` (

`sal\_detail\_id` int NOT NULL AUTO\_INCREMENT,

`Employee\_Benefits\_id` int NOT NULL,

`BASE\_SALARY` decimal(10, 2) NULL DEFAULT NULL,

`BONUS` decimal(10, 2) NULL DEFAULT NULL,

`ALLOWANCE` decimal(10, 2) NULL DEFAULT NULL,

`DEDUCTIONS` decimal(10, 2) NULL DEFAULT NULL,

`BENEFITS` decimal(10, 2) NULL DEFAULT NULL,

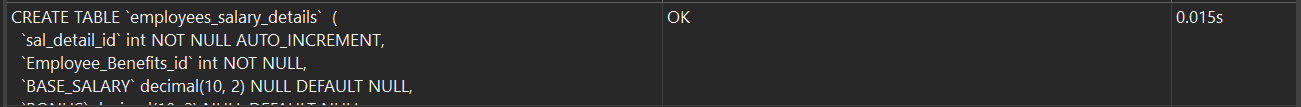
PRIMARY KEY (`sal\_detail\_id`) USING BTREE,

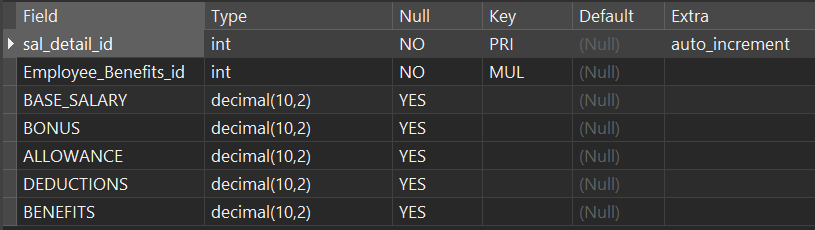
INDEX `FK\_EMPLOYEES\_SALARY\_DETAILS\_Employee\_Benefits\_id`(`Employee\_Benefits\_id` ASC) USING BTREE

) ENGINE = InnoDB AUTO\_INCREMENT = 1 CHARACTER SET = utf8mb4 COLLATE = utf8mb4\_0900\_ai\_ci ROW\_FORMAT = DYNAMIC;

**Excuted Result (with Query Commands):**

(Excute successfully)



DESC employees\_salary\_details;

SELECT \* FROM employees\_salary\_details;

**CREATE TABLE Commands:**

CREATE TABLE `interview\_list` (

`Interview\_list\_id` int NOT NULL AUTO\_INCREMENT,

`Candidate\_id` int NOT NULL,

`Interviewer\_id` int NOT NULL,

`Interview\_date` date NULL DEFAULT NULL,

`Interview\_result` varchar(500) CHARACTER SET utf8mb4 COLLATE utf8mb4\_0900\_ai\_ci NULL DEFAULT NULL,

PRIMARY KEY (`Interview\_list\_id`) USING BTREE,

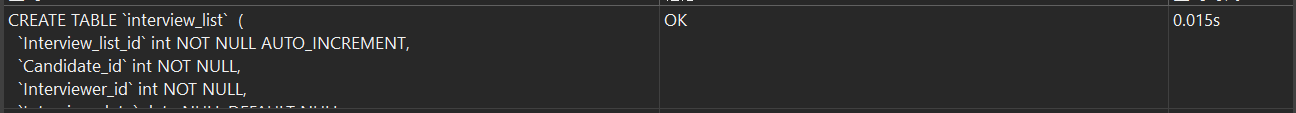
INDEX `FK\_Interview\_List\_Candidate\_id`(`Candidate\_id` ASC) USING BTREE,

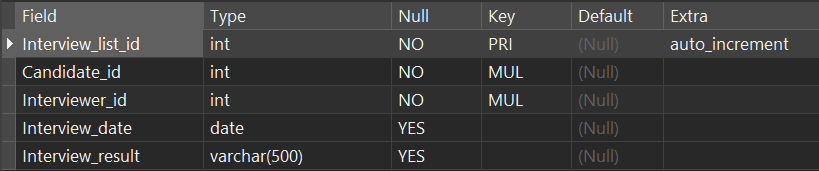
INDEX `FK\_Interview\_List\_Interviewer\_id`(`Interviewer\_id` ASC) USING BTREE

) ENGINE = InnoDB AUTO\_INCREMENT = 1 CHARACTER SET = utf8mb4 COLLATE = utf8mb4\_0900\_ai\_ci ROW\_FORMAT = DYNAMIC;

**Excuted Result (with Query Commands):**

(Excute successfully)



DESC interview\_list;

SELECT \* FROM interview\_list;

**CREATE TABLE Commands:**

CREATE TABLE `leave\_record` (

`leave\_record\_id` int NOT NULL AUTO\_INCREMENT,

`leave\_req\_id` int NULL DEFAULT NULL,

`leave\_start\_date` date NULL DEFAULT NULL,

`leave\_end\_date` date NULL DEFAULT NULL,

`leave\_approver\_id` int NOT NULL,

PRIMARY KEY (`leave\_record\_id`) USING BTREE,

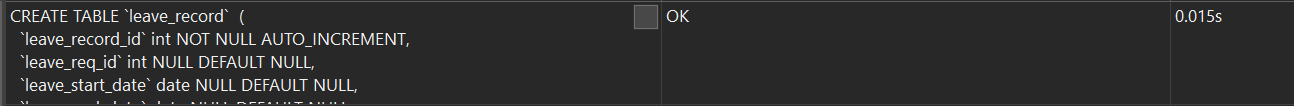
INDEX `FK\_Leave\_Record\_leave\_req\_id`(`leave\_req\_id` ASC) USING BTREE,

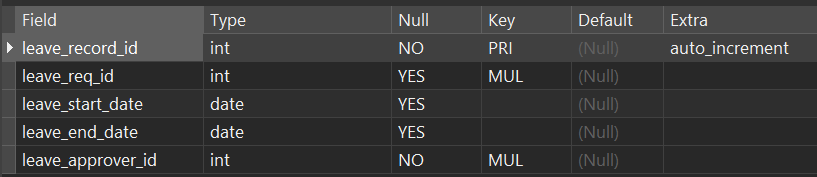
INDEX `FK\_Leave\_Record\_leave\_approver\_id`(`leave\_approver\_id` ASC) USING BTREE

) ENGINE = InnoDB AUTO\_INCREMENT = 1 CHARACTER SET = utf8mb4 COLLATE = utf8mb4\_0900\_ai\_ci ROW\_FORMAT = DYNAMIC;\

**Excuted Result (with Query Commands):**

(Excute successfully)



DESC leave\_record;

SELECT \* FROM leave\_record; 

**CREATE TABLE Commands:**

CREATE TABLE `leave\_request` (

`leave\_req\_id` int NOT NULL AUTO\_INCREMENT,

`requester\_id` int NOT NULL,

`req\_reason` varchar(50) CHARACTER SET utf8mb4 COLLATE utf8mb4\_0900\_ai\_ci NULL DEFAULT NULL,

`req\_start\_date` date NULL DEFAULT NULL,

`req\_end\_date` date NULL DEFAULT NULL,

`approve\_status` varchar(50) CHARACTER SET utf8mb4 COLLATE utf8mb4\_0900\_ai\_ci NULL DEFAULT NULL,

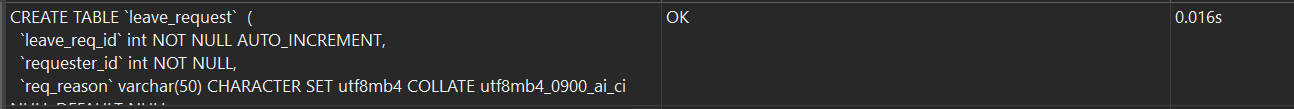
PRIMARY KEY (`leave\_req\_id`) USING BTREE,

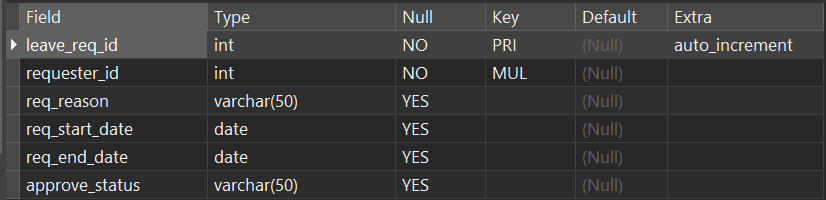
INDEX `FK\_Leave\_Request\_requester\_id`(`requester\_id` ASC) USING BTREE

) ENGINE = InnoDB AUTO\_INCREMENT = 1 CHARACTER SET = utf8mb4 COLLATE = utf8mb4\_0900\_ai\_ci ROW\_FORMAT = DYNAMIC;

**Excuted Result (with Query Commands):**

(Excute successfully)



DESC leave\_request;

SELECT \* FROM leave\_request; 

**CREATE TABLE Commands:**

CREATE TABLE `ot\_record` (

`OT\_Record\_id` int NOT NULL AUTO\_INCREMENT,

`record\_date` date NULL DEFAULT NULL,

`time\_in` time NULL DEFAULT NULL,

`time\_out` time NULL DEFAULT NULL,

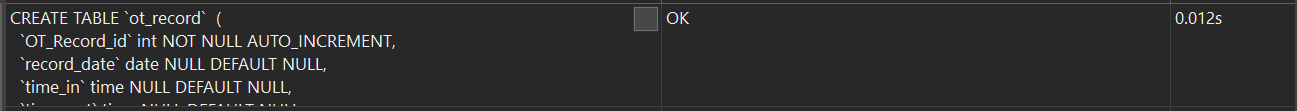
`ot\_hour` decimal(10, 2) NULL DEFAULT NULL,

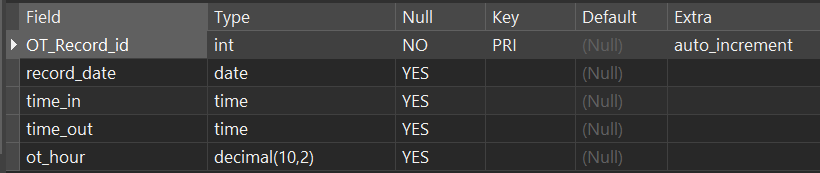
PRIMARY KEY (`OT\_Record\_id`) USING BTREE

) ENGINE = InnoDB AUTO\_INCREMENT = 1 CHARACTER SET = utf8mb4 COLLATE = utf8mb4\_0900\_ai\_ci ROW\_FORMAT = DYNAMIC;

**Excuted Result (with Query Commands):**

(Excute successfully)



DESC ot\_record;

SELECT \* FROM ot\_record; 

**CREATE TABLE Commands:**

CREATE TABLE `position\_application` (

`position\_apply\_id` int NOT NULL AUTO\_INCREMENT,

`emp\_id` int NOT NULL,

`current\_position\_id` int NOT NULL,

`position\_opening\_id` int NOT NULL,

`application\_reason` varchar(100) CHARACTER SET utf8mb4 COLLATE utf8mb4\_0900\_ai\_ci NULL DEFAULT NULL,

`skillset\_id` int NULL DEFAULT NULL,

PRIMARY KEY (`position\_apply\_id`) USING BTREE,

INDEX `emp\_id`(`emp\_id` ASC) USING BTREE,

INDEX `current\_position\_id`(`current\_position\_id` ASC) USING BTREE,

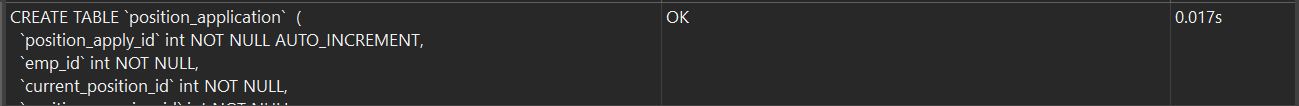
INDEX `position\_opening\_id`(`position\_opening\_id` ASC) USING BTREE,

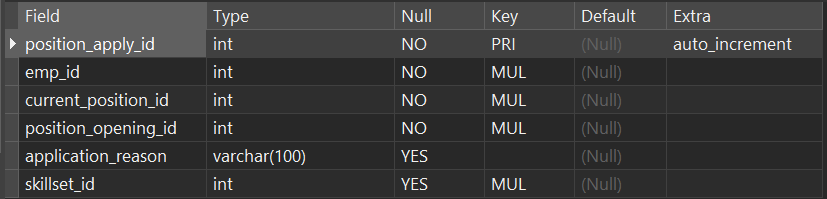
INDEX `skillset\_id`(`skillset\_id` ASC) USING BTREE

) ENGINE = InnoDB AUTO\_INCREMENT = 1 CHARACTER SET = utf8mb4 COLLATE = utf8mb4\_0900\_ai\_ci ROW\_FORMAT = DYNAMIC;

**Excuted Result (with Query Commands):**

(Excute successfully)



DESC position\_application;

SELECT \* FROM position\_application; 

**CREATE TABLE Commands:**

CREATE TABLE `position\_opening` (

`Position\_Opening\_id` int NOT NULL AUTO\_INCREMENT,

`Position\_id` int NULL DEFAULT NULL,

`Dept\_id` int NULL DEFAULT NULL,

`Experience\_Required` varchar(500) CHARACTER SET utf8mb4 COLLATE utf8mb4\_0900\_ai\_ci NULL DEFAULT NULL,

`Education\_Required` varchar(500) CHARACTER SET utf8mb4 COLLATE utf8mb4\_0900\_ai\_ci NULL DEFAULT NULL,

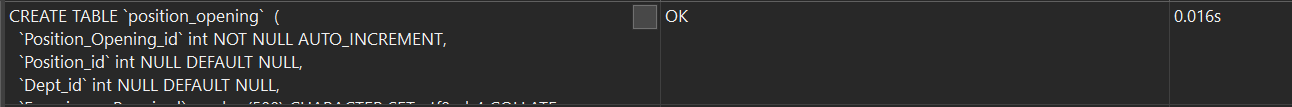
PRIMARY KEY (`Position\_Opening\_id`) USING BTREE,

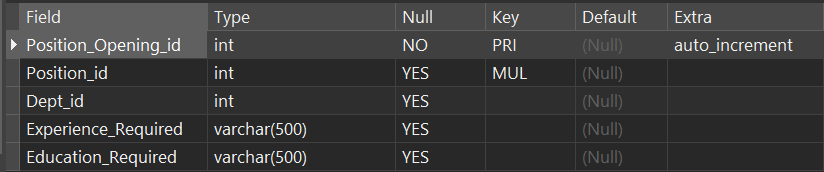
INDEX `FK\_position\_Opening\_Position\_id`(`Position\_id` ASC) USING BTREE

) ENGINE = InnoDB AUTO\_INCREMENT = 1 CHARACTER SET = utf8mb4 COLLATE = utf8mb4\_0900\_ai\_ci ROW\_FORMAT = DYNAMIC;

**Excuted Result (with Query Commands):**

(Excute successfully)



DESC position\_opening;

SELECT \* FROM position\_opening; 

**CREATE TABLE Commands:**

CREATE TABLE `positions` (

`position\_id` int NOT NULL AUTO\_INCREMENT,

`position\_name` varchar(500) CHARACTER SET utf8mb4 COLLATE utf8mb4\_0900\_ai\_ci NULL DEFAULT NULL,

`avg\_salary` decimal(10, 2) NULL DEFAULT NULL,

`dept\_id` int NULL DEFAULT NULL,

`emp\_count` int NULL DEFAULT NULL,

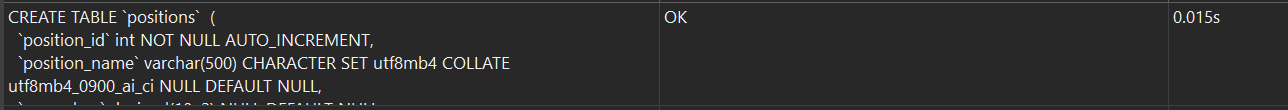
PRIMARY KEY (`position\_id`) USING BTREE,

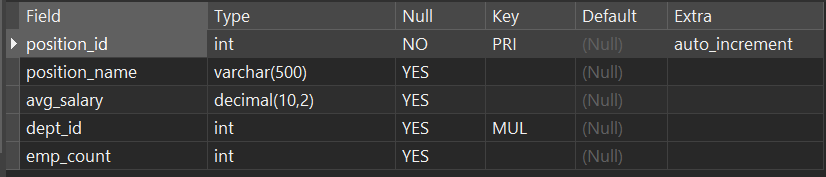
INDEX `FK\_POSITIONS\_DEPT\_ID`(`dept\_id` ASC) USING BTREE

) ENGINE = InnoDB AUTO\_INCREMENT = 1 CHARACTER SET = utf8mb4 COLLATE = utf8mb4\_0900\_ai\_ci ROW\_FORMAT = DYNAMIC;

**Excuted Result (with Query Commands):**

(Excute successfully)



DESC positions;

SELECT \* FROM positions; 

**CREATE TABLE Commands:**

CREATE TABLE `project\_participation` (

`project\_participate\_id` int NOT NULL AUTO\_INCREMENT,

`emp\_pnd\_id` int NULL DEFAULT NULL,

`project\_id` int NULL DEFAULT NULL,

`comment` varchar(50) CHARACTER SET utf8mb4 COLLATE utf8mb4\_0900\_ai\_ci NULL DEFAULT NULL,

`start\_date` date NULL DEFAULT NULL,

`end\_date` date NULL DEFAULT NULL,

PRIMARY KEY (`project\_participate\_id`) USING BTREE,

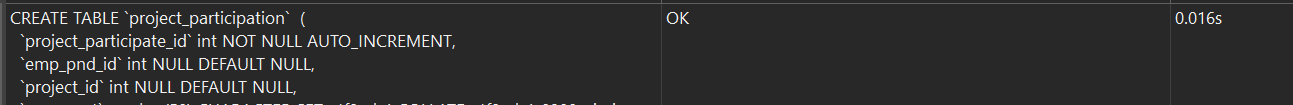
INDEX `emp\_pnd\_id`(`emp\_pnd\_id` ASC) USING BTREE,

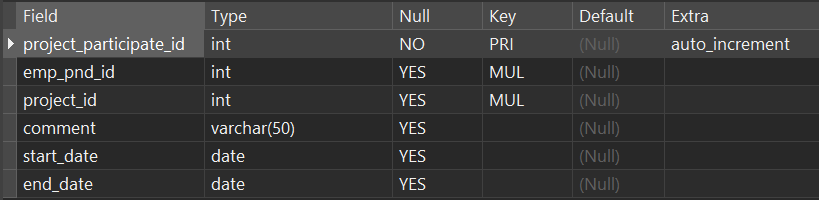
INDEX `FK\_project\_participation\_project\_id`(`project\_id` ASC) USING BTREE

) ENGINE = InnoDB AUTO\_INCREMENT = 1 CHARACTER SET = utf8mb4 COLLATE = utf8mb4\_0900\_ai\_ci ROW\_FORMAT = DYNAMIC;

**Excuted Result (with Query Commands):**

(Excute successfully)



DESC project\_participation;

SELECT \* FROM project\_participation; 

**CREATE TABLE Commands:**

CREATE TABLE `projects` (

`project\_id` int NOT NULL AUTO\_INCREMENT,

`project\_title` varchar(50) CHARACTER SET utf8mb4 COLLATE utf8mb4\_0900\_ai\_ci NULL DEFAULT NULL,

`leader\_id` int NULL DEFAULT NULL,

`start\_date` date NULL DEFAULT NULL,

`end\_date` date NULL DEFAULT NULL,

`comment` varchar(50) CHARACTER SET utf8mb4 COLLATE utf8mb4\_0900\_ai\_ci NULL DEFAULT NULL,

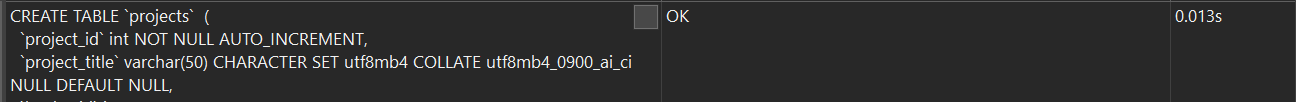
PRIMARY KEY (`project\_id`) USING BTREE,

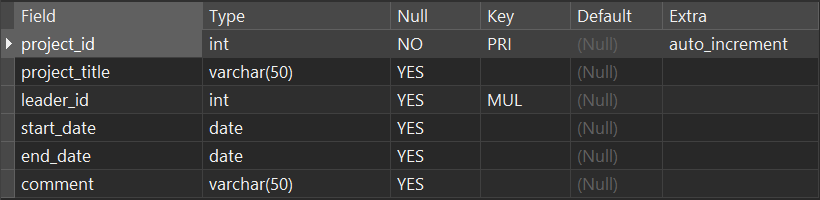
INDEX `leader\_id`(`leader\_id` ASC) USING BTREE

) ENGINE = InnoDB AUTO\_INCREMENT = 1 CHARACTER SET = utf8mb4 COLLATE = utf8mb4\_0900\_ai\_ci ROW\_FORMAT = DYNAMIC;

**Excuted Result (with Query Commands):**

(Excute successfully)



DESC projects;

SELECT \* FROM projects; 

**CREATE TABLE Commands:**

CREATE TABLE `resignations` (

`resign\_emp\_id` int NOT NULL AUTO\_INCREMENT,

`resign\_reason` varchar(500) CHARACTER SET utf8mb4 COLLATE utf8mb4\_0900\_ai\_ci NULL DEFAULT NULL,

`RESIGN\_DATE` date NULL DEFAULT NULL,

`related\_contract\_id` int NULL DEFAULT NULL,

`exit\_interviewer\_id` int NULL DEFAULT NULL,

PRIMARY KEY (`resign\_emp\_id`) USING BTREE,

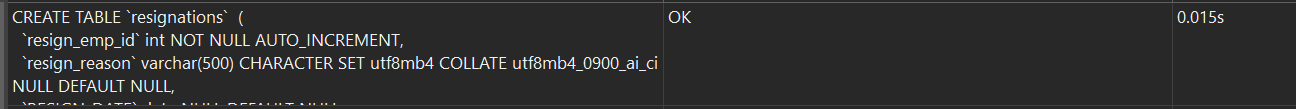
INDEX `related\_contract\_id`(`related\_contract\_id` ASC) USING BTREE,

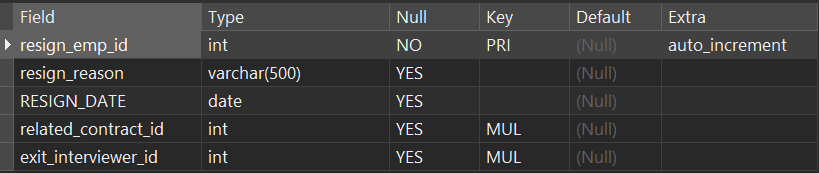
INDEX `exit\_interviewer\_id`(`exit\_interviewer\_id` ASC) USING BTREE

) ENGINE = InnoDB AUTO\_INCREMENT = 1 CHARACTER SET = utf8mb4 COLLATE = utf8mb4\_0900\_ai\_ci ROW\_FORMAT = DYNAMIC;

**Excuted Result (with Query Commands):**

(Excute successfully)



DESC resignations;

SELECT \* FROM resignations; 

**CREATE TABLE Commands:**

CREATE TABLE `salary\_history` (

`SAL\_HISTORY\_ID` int NOT NULL AUTO\_INCREMENT,

`Employee\_Benefits\_id` int NOT NULL,

`gender` varchar(10) CHARACTER SET utf8mb4 COLLATE utf8mb4\_0900\_ai\_ci NULL DEFAULT NULL,

`TOTAL\_SALARY` decimal(10, 2) NULL DEFAULT NULL,

`START\_DATE` date NULL DEFAULT NULL,

`END\_DATE` date NULL DEFAULT NULL,

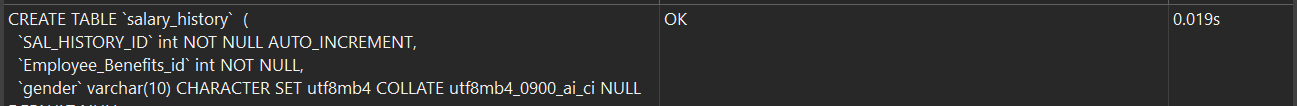
PRIMARY KEY (`SAL\_HISTORY\_ID`) USING BTREE,

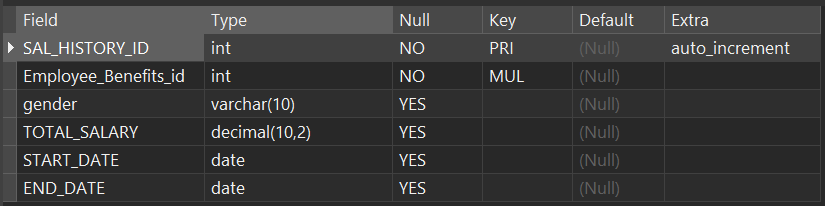
INDEX `FK\_SALARY\_HISTORY\_SALARY\_DETAILS\_Employee\_Benefits\_id`(`Employee\_Benefits\_id` ASC) USING BTREE

) ENGINE = InnoDB AUTO\_INCREMENT = 1 CHARACTER SET = utf8mb4 COLLATE = utf8mb4\_0900\_ai\_ci ROW\_FORMAT = DYNAMIC;

**Excuted Result (with Query Commands):**

(Excute successfully)



DESC salary\_history;

SELECT \* FROM salary\_history; 

**CREATE TABLE Commands:**

CREATE TABLE `skills` (

`Skill\_id` int NOT NULL AUTO\_INCREMENT,

`Skill\_name` varchar(500) CHARACTER SET utf8mb4 COLLATE utf8mb4\_0900\_ai\_ci NULL DEFAULT NULL,

`dept\_id` int NULL DEFAULT NULL,

`skill\_resources` varchar(500) CHARACTER SET utf8mb4 COLLATE utf8mb4\_0900\_ai\_ci NULL DEFAULT NULL,

`skill\_description` varchar(500) CHARACTER SET utf8mb4 COLLATE utf8mb4\_0900\_ai\_ci NULL DEFAULT NULL,

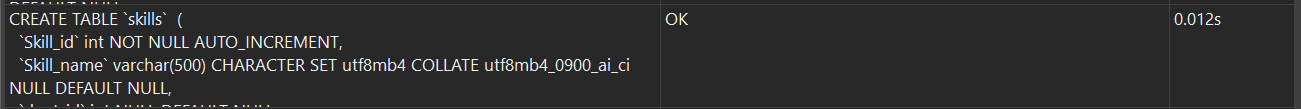
PRIMARY KEY (`Skill\_id`) USING BTREE,

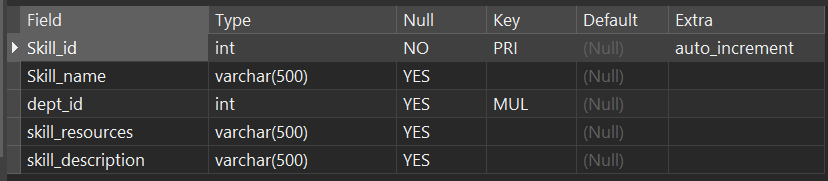
INDEX `dept\_id`(`dept\_id` ASC) USING BTREE

) ENGINE = InnoDB AUTO\_INCREMENT = 1 CHARACTER SET = utf8mb4 COLLATE = utf8mb4\_0900\_ai\_ci ROW\_FORMAT = DYNAMIC;

**Excuted Result (with Query Commands):**

(Excute successfully)



DESC skills;

SELECT \* FROM skills; 

**CREATE TABLE Commands:**

CREATE TABLE `skillset` (

`Skillset\_id` int NOT NULL AUTO\_INCREMENT,

`emp\_pnd\_id` int NULL DEFAULT NULL,

`candidate\_id` int NULL DEFAULT NULL,

`Skill\_one\_id` int NULL DEFAULT NULL,

`Skill\_two\_id` int NULL DEFAULT NULL,

`Skill\_three\_id` int NULL DEFAULT NULL,

PRIMARY KEY (`Skillset\_id`) USING BTREE,

INDEX `candidate\_id`(`candidate\_id` ASC) USING BTREE,

INDEX `Skill\_one\_id`(`Skill\_one\_id` ASC) USING BTREE,

INDEX `Skill\_two\_id`(`Skill\_two\_id` ASC) USING BTREE,

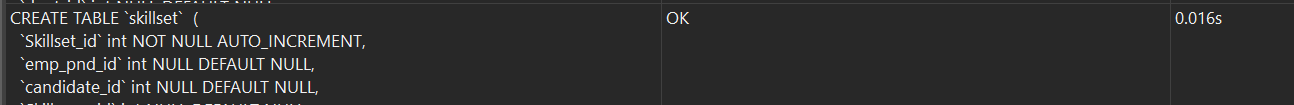
INDEX `Skill\_three\_id`(`Skill\_three\_id` ASC) USING BTREE,

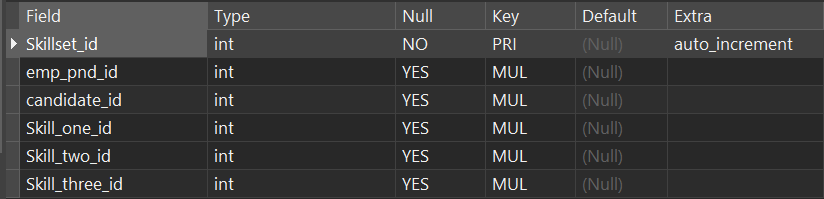
INDEX `FK\_Skillset\_emp\_pnd\_id`(`emp\_pnd\_id` ASC) USING BTREE

) ENGINE = InnoDB AUTO\_INCREMENT = 1 CHARACTER SET = utf8mb4 COLLATE = utf8mb4\_0900\_ai\_ci ROW\_FORMAT = DYNAMIC;

**Excuted Result (with Query Commands):**

(Excute successfully)



DESC skillset; 

SELECT \* FROM skillset;

**CREATE TABLE Commands:**

CREATE TABLE `users` (

`user\_id` int NOT NULL AUTO\_INCREMENT,

`emp\_id` int NOT NULL,

`username` varchar(20) CHARACTER SET utf8mb4 COLLATE utf8mb4\_0900\_ai\_ci NOT NULL,

`password` varchar(20) CHARACTER SET utf8mb4 COLLATE utf8mb4\_0900\_ai\_ci NOT NULL,

`permission\_type` varchar(20) CHARACTER SET utf8mb4 COLLATE utf8mb4\_0900\_ai\_ci NULL DEFAULT NULL,

`description` varchar(500) CHARACTER SET utf8mb4 COLLATE utf8mb4\_0900\_ai\_ci NULL DEFAULT NULL,

`creat\_date` date NULL DEFAULT NULL,

`last\_update\_date` date NULL DEFAULT NULL,

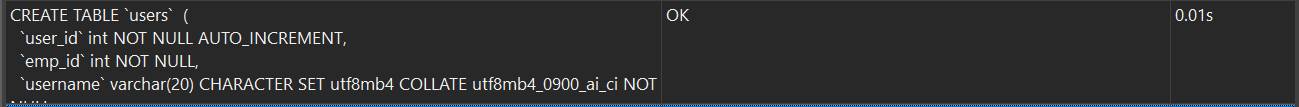
PRIMARY KEY (`user\_id`) USING BTREE,

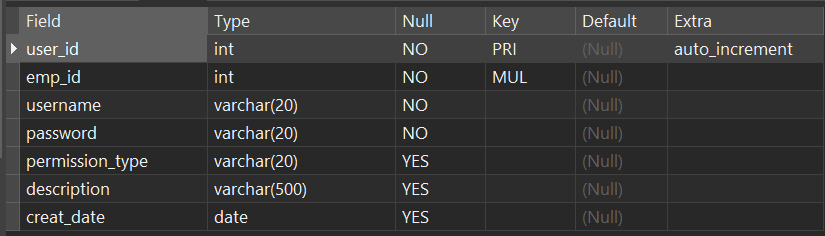
INDEX `emp\_id`(`emp\_id` ASC) USING BTREE

) ENGINE = InnoDB AUTO\_INCREMENT = 1 CHARACTER SET = utf8mb4 COLLATE = utf8mb4\_0900\_ai\_ci ROW\_FORMAT = Dynamic;

**Excuted Result (with Query Commands):**

(Excute successfully)



DESC users;

SELECT \* FROM users;

**CREATE TABLE Commands:**

CREATE TABLE `work\_exp` (

`work\_exp\_id` int NOT NULL AUTO\_INCREMENT,

`emp\_files\_id` int NULL DEFAULT NULL,

`company\_name` varchar(50) CHARACTER SET utf8mb4 COLLATE utf8mb4\_0900\_ai\_ci NULL DEFAULT NULL,

`POSITION` varchar(50) CHARACTER SET utf8mb4 COLLATE utf8mb4\_0900\_ai\_ci NULL DEFAULT NULL,

`start\_date` date NULL DEFAULT NULL,

`end\_date` date NULL DEFAULT NULL,

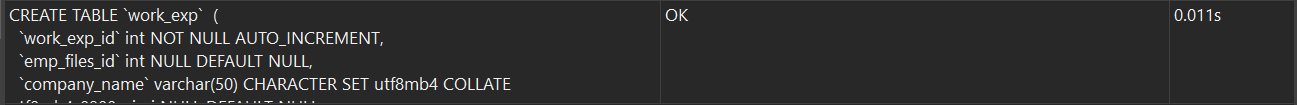
PRIMARY KEY (`work\_exp\_id`) USING BTREE,

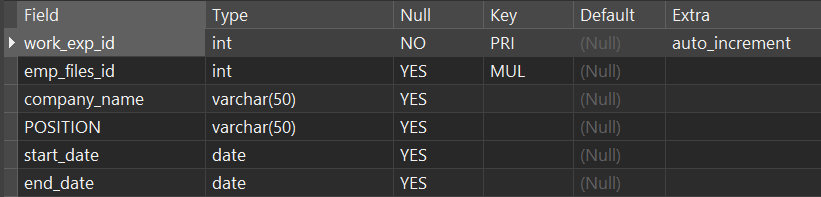
INDEX `FK\_Work\_Exp\_emp\_files\_id`(`emp\_files\_id` ASC) USING BTREE

) ENGINE = InnoDB AUTO\_INCREMENT = 1 CHARACTER SET = utf8mb4 COLLATE = utf8mb4\_0900\_ai\_ci ROW\_FORMAT = DYNAMIC;

**Query Excuted Result (with Query Commands):**

(Excute successfully)



DESC work\_exp;

SELECT \* FROM work\_exp; 

**2.2.2 ALTER TABLE**

- The ALTER statement in SQL is used to modify existing database objects, such as tables, columns, or constraints. It allows you to add, modify, or delete columns, change data types, rename tables, add or drop constraints, and more. ALTER is a powerful Commands that helps in adapting the database structure as business needs evolve over time without the need to recreate the entire database.

**ADD COLUMN**

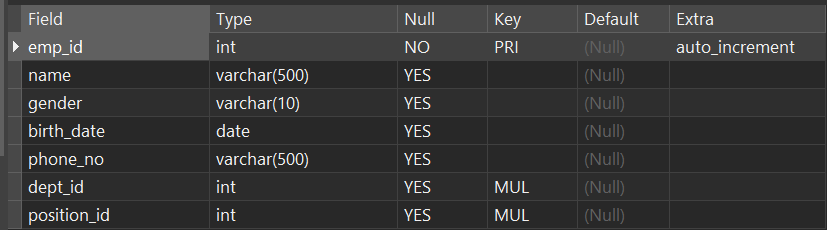
**ADD COLUMN Commands:**

ALTER TABLE employees ADD COLUMN manager\_id INT(100);

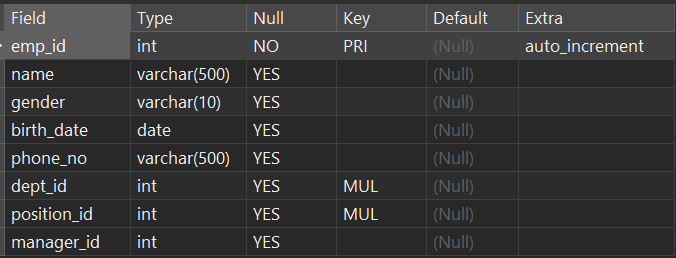
**Excuted Result (with Query Commands):**

DESC employees;

***Before:***



***After:***



**MODIFY COLUMN**

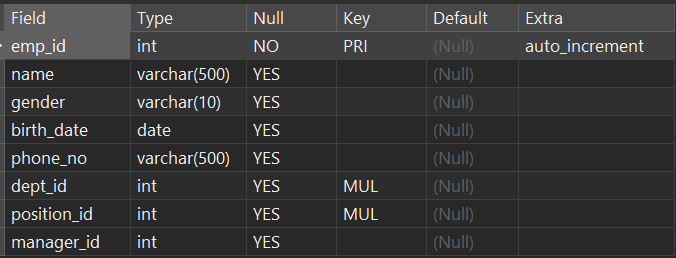
**MODIFY COLUMN Commands:**

ALTER TABLE employees MODIFY COLUMN manager\_id DATE;

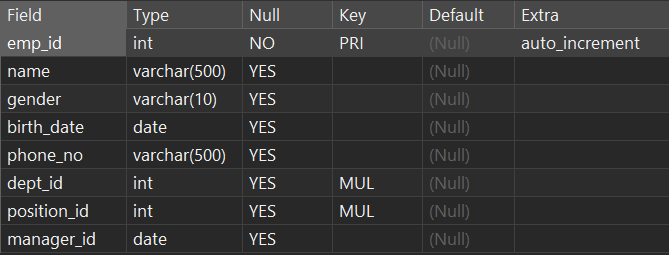
**Excuted Result (with Query Commands):**

DESC employees;

***Before:***



***After:***

****

**CHANGE COLUMN**

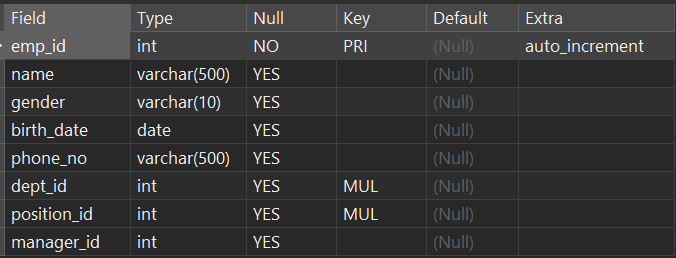
**CHANGE COLUMN Commands:**

ALTER TABLE employees CHANGE COLUMN manager\_id supervisor\_id INT(100);

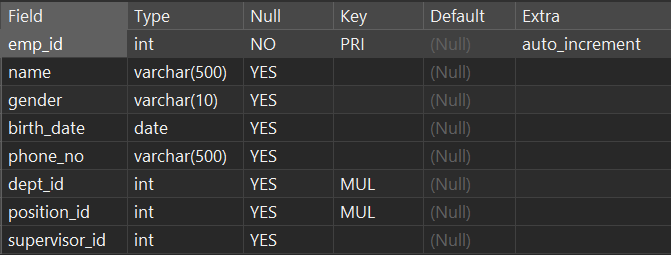
**Excuted Result (with Query Commands):**

DESC employees;

***Before:***



***After:***

****

**DROP COLUMN**

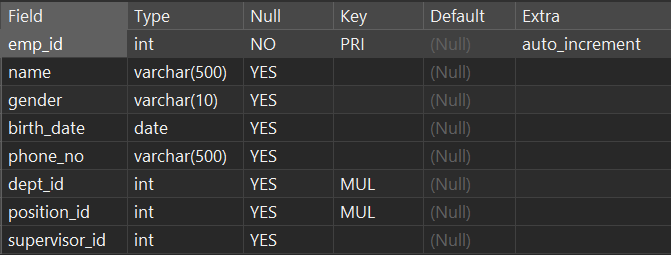
**DROP COLUMN Commands:**

ALTER TABLE employees DROP COLUMN supervisor\_id;

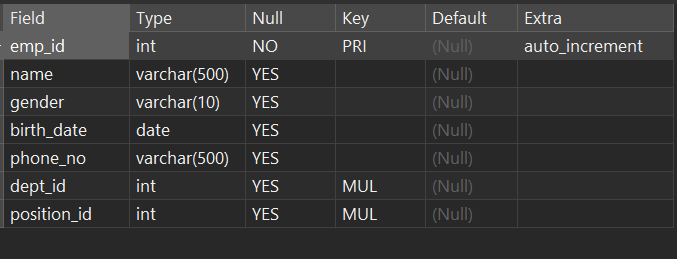
**Excuted Result (with Query Commands):**

DESC employees;

***Before:***

****

***After:***

****

**RENAME TO**

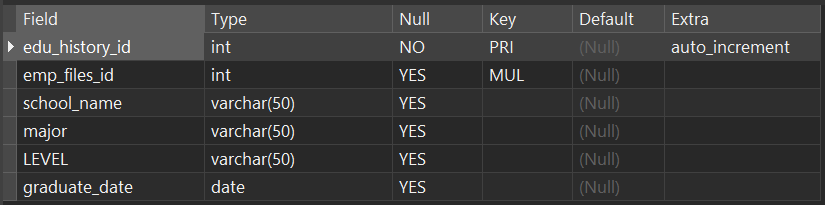
**RENAME TO Commands:**

ALTER TABLE edu\_history RENAME TO education\_history;

**Excuted Result (with Query Commands):**

DESC edu\_history;

***Before:***



***After:***

****

**ADD CONSTRAINT**

- The next step will demonstrate adding foreign keys to the created tables to achieve tight connections between them.

- There are a total of 53 foreign keys in the system, which are used to effectively connect 31 tables, ensuring data association and integrity.

**ADD CONSTARINT Commands:**

ALTER TABLE `attendance\_record` ADD CONSTRAINT `attendance\_record\_ibfk\_1` FOREIGN KEY (`attendance\_record\_id`) REFERENCES `employee\_attendance` (`employee\_Attendance\_id`) ON DELETE RESTRICT ON UPDATE RESTRICT;

**Excuted Result (with Query Commands):**

(Excute successfully)



SELECT \* FROM information\_schema.TABLE\_CONSTRAINTS

WHERE CONSTRAINT\_SCHEMA = 'human\_resource\_management\_system' AND TABLE\_NAME = 'attendance\_record' AND CONSTRAINT\_TYPE = 'FOREIGN KEY';

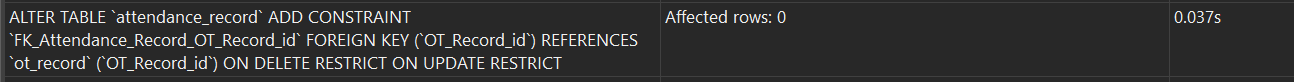


**ADD CONSTARINT Commands:**

ALTER TABLE `attendance\_record` ADD CONSTRAINT `FK\_Attendance\_Record\_OT\_Record\_id` FOREIGN KEY (`OT\_Record\_id`) REFERENCES `ot\_record` (`OT\_Record\_id`) ON DELETE RESTRICT ON UPDATE RESTRICT;

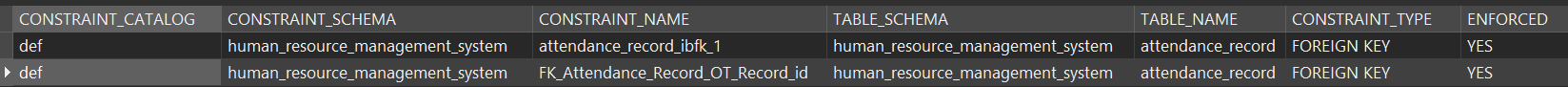
**Excuted Result (with Query Commands):**

(Excute successfully)



SELECT \* FROM information\_schema.TABLE\_CONSTRAINTS

WHERE CONSTRAINT\_SCHEMA = 'human\_resource\_management\_system' AND TABLE\_NAME = 'attendance\_record' AND CONSTRAINT\_TYPE = 'FOREIGN KEY';

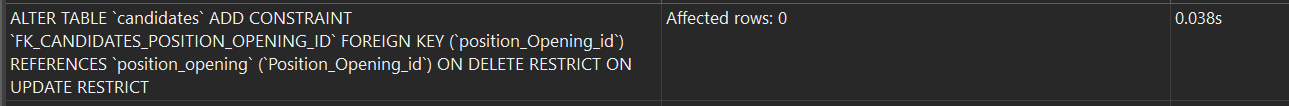


**ADD CONSTARINT Commands:**

ALTER TABLE `candidates` ADD CONSTRAINT `FK\_CANDIDATES\_POSITION\_OPENING\_ID` FOREIGN KEY (`position\_Opening\_id`) REFERENCES `position\_opening` (`Position\_Opening\_id`) ON DELETE RESTRICT ON UPDATE RESTRICT;

**Excuted Result (with Query Commands):**

(Excute successfully)



SELECT \* FROM information\_schema.TABLE\_CONSTRAINTS

WHERE CONSTRAINT\_SCHEMA = 'human\_resource\_management\_system' AND TABLE\_NAME = 'candidates' AND CONSTRAINT\_TYPE = 'FOREIGN KEY';



**ADD CONSTARINT Commands:**

ALTER TABLE `contracts` ADD CONSTRAINT `FK\_contracts\_emp\_file\_id` FOREIGN KEY (`emp\_files\_id`) REFERENCES `employee\_files` (`emp\_files\_id`) ON DELETE RESTRICT ON UPDATE RESTRICT;

**Excuted Result (with Query Commands):**

(Excute successfully)



SELECT \* FROM information\_schema.TABLE\_CONSTRAINTS WHERE CONSTRAINT\_SCHEMA = 'human\_resource\_management\_system' AND TABLE\_NAME = 'contracts' AND CONSTRAINT\_TYPE = 'FOREIGN KEY';

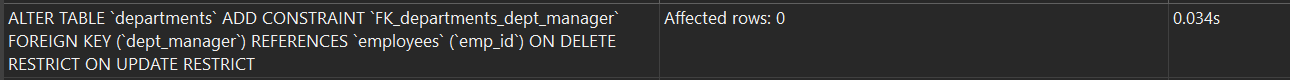


**ADD CONSTARINT Commands:**

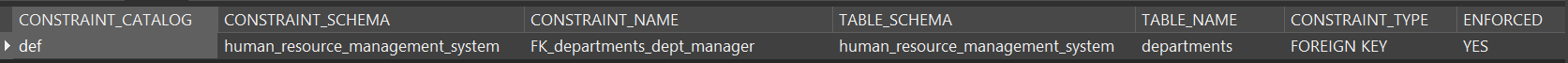
ALTER TABLE `departments` ADD CONSTRAINT `FK\_departments\_dept\_manager` FOREIGN KEY (`dept\_manager`) REFERENCES `employees` (`emp\_id`) ON DELETE RESTRICT ON UPDATE RESTRICT;

**Excuted Result (with Query Commands):**

(Excute successfully)



SELECT \* FROM information\_schema.TABLE\_CONSTRAINTS WHERE CONSTRAINT\_SCHEMA = 'human\_resource\_management\_system' AND TABLE\_NAME = 'departments' AND CONSTRAINT\_TYPE = 'FOREIGN KEY';



**ADD CONSTARINT Commands:**

ALTER TABLE `edu\_history` ADD CONSTRAINT `FK\_Edu\_History\_emp\_files\_id` FOREIGN KEY (`emp\_files\_id`) REFERENCES `employee\_files` (`emp\_files\_id`) ON DELETE RESTRICT ON UPDATE RESTRICT;

**Excuted Result (with Query Commands):**

(Excute successfully)



SELECT \* FROM information\_schema.TABLE\_CONSTRAINTS WHERE CONSTRAINT\_SCHEMA = 'human\_resource\_management\_system' AND TABLE\_NAME = 'edu\_history' AND CONSTRAINT\_TYPE = 'FOREIGN KEY';

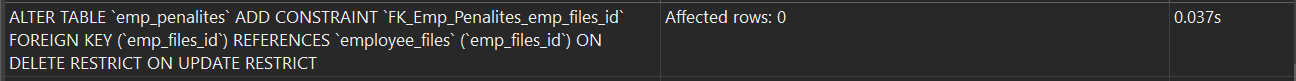


**ADD CONSTARINT Commands:**

ALTER TABLE `emp\_penalites` ADD CONSTRAINT `FK\_Emp\_Penalites\_emp\_files\_id` FOREIGN KEY (`emp\_files\_id`) REFERENCES `employee\_files` (`emp\_files\_id`) ON DELETE RESTRICT ON UPDATE RESTRICT;

**Excuted Result (with Query Commands):**

(Excute successfully)



SELECT \* FROM information\_schema.TABLE\_CONSTRAINTS WHERE CONSTRAINT\_SCHEMA = 'human\_resource\_management\_system' AND TABLE\_NAME = 'emp\_penalites' AND CONSTRAINT\_TYPE = 'FOREIGN KEY';



**ADD CONSTARINT Commands:**

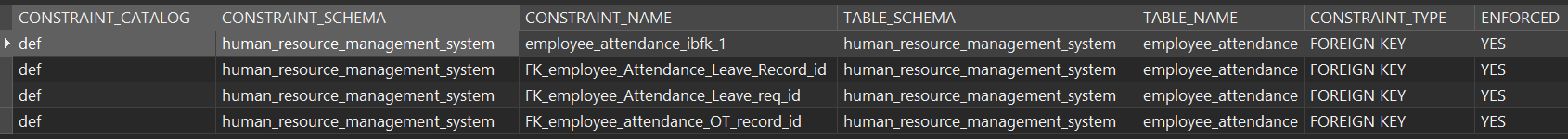
ALTER TABLE `employee\_attendance` ADD CONSTRAINT `employee\_attendance\_ibfk\_1` FOREIGN KEY (`employee\_Attendance\_id`) REFERENCES `employee\_files` (`emp\_files\_id`) ON DELETE RESTRICT ON UPDATE RESTRICT;

**Excuted Result (with Query Commands):**

(Excute successfully)



SELECT \* FROM information\_schema.TABLE\_CONSTRAINTS WHERE CONSTRAINT\_SCHEMA = 'human\_resource\_management\_system' AND TABLE\_NAME = 'employee\_attendance' AND CONSTRAINT\_TYPE = 'FOREIGN KEY';

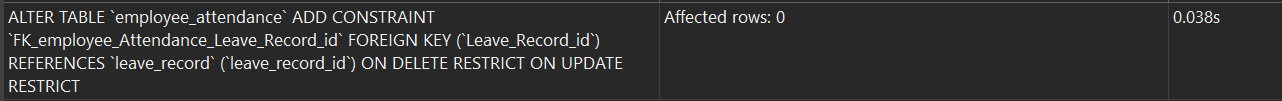


**ADD CONSTARINT Commands:**

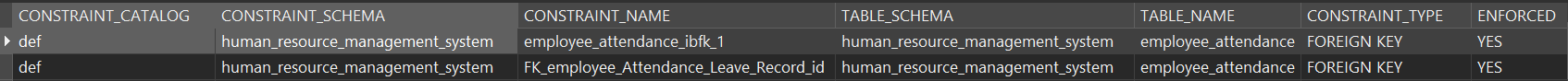
ALTER TABLE `employee\_attendance` ADD CONSTRAINT `FK\_employee\_Attendance\_Leave\_Record\_id` FOREIGN KEY (`Leave\_Record\_id`) REFERENCES `leave\_record` (`leave\_record\_id`) ON DELETE RESTRICT ON UPDATE RESTRICT;

**Excuted Result (with Query Commands):**

(Excute successfully)



SELECT \* FROM information\_schema.TABLE\_CONSTRAINTS WHERE CONSTRAINT\_SCHEMA = 'human\_resource\_management\_system' AND TABLE\_NAME = 'employee\_attendance' AND CONSTRAINT\_TYPE = 'FOREIGN KEY';

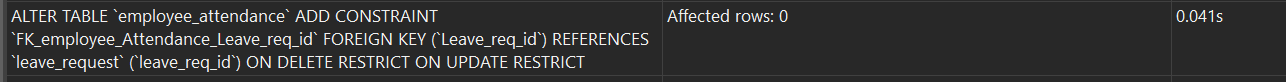


**10 ADD CONSTARINT Commands:**

ALTER TABLE `employee\_attendance` ADD CONSTRAINT `FK\_employee\_Attendance\_Leave\_req\_id` FOREIGN KEY (`Leave\_req\_id`) REFERENCES `leave\_request` (`leave\_req\_id`) ON DELETE RESTRICT ON UPDATE RESTRICT;

**Excuted Result (with Query Commands):**

(Excute successfully)



SELECT \* FROM information\_schema.TABLE\_CONSTRAINTS WHERE CONSTRAINT\_SCHEMA = 'human\_resource\_management\_system' AND TABLE\_NAME = 'employee\_attendance' AND CONSTRAINT\_TYPE = 'FOREIGN KEY';

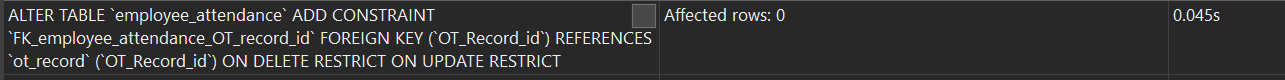


**ADD CONSTARINT Commands:**

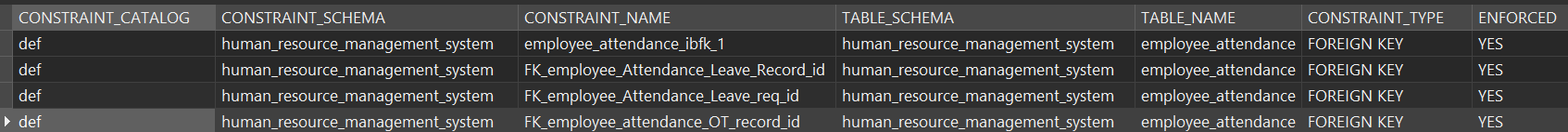
ALTER TABLE `employee\_attendance` ADD CONSTRAINT `FK\_employee\_attendance\_OT\_record\_id` FOREIGN KEY (`OT\_Record\_id`) REFERENCES `ot\_record` (`OT\_Record\_id`) ON DELETE RESTRICT ON UPDATE RESTRICT;

**Excuted Result (with Query Commands):**

(Excute successfully)



SELECT \* FROM information\_schema.TABLE\_CONSTRAINTS WHERE CONSTRAINT\_SCHEMA = 'human\_resource\_management\_system' AND TABLE\_NAME = 'employee\_attendance' AND CONSTRAINT\_TYPE = 'FOREIGN KEY';



**ADD CONSTARINT Commands:**

ALTER TABLE `employee\_awards` ADD CONSTRAINT `employee\_awards\_ibfk\_1` FOREIGN KEY (`Employee\_Benefits\_id`) REFERENCES `employee\_benefits` (`Employee\_Benefits\_id`) ON DELETE RESTRICT ON UPDATE RESTRICT;

**Excuted Result (with Query Commands):**

(Excute successfully)



SELECT \* FROM information\_schema.TABLE\_CONSTRAINTS WHERE CONSTRAINT\_SCHEMA = 'human\_resource\_management\_system' AND TABLE\_NAME = 'employee\_awards' AND CONSTRAINT\_TYPE = 'FOREIGN KEY';

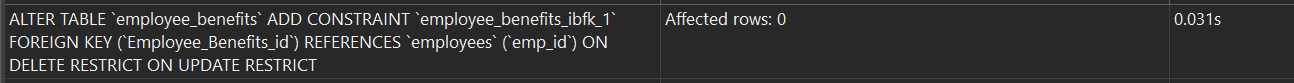


**ADD CONSTARINT Commands:**

ALTER TABLE `employee\_benefits` ADD CONSTRAINT `employee\_benefits\_ibfk\_1` FOREIGN KEY (`Employee\_Benefits\_id`) REFERENCES `employees` (`emp\_id`) ON DELETE RESTRICT ON UPDATE RESTRICT;

**Excuted Result (with Query Commands):**

(Excute successfully)



SELECT \* FROM information\_schema.TABLE\_CONSTRAINTS WHERE CONSTRAINT\_SCHEMA = 'human\_resource\_management\_system' AND TABLE\_NAME = 'employee\_benefits' AND CONSTRAINT\_TYPE = 'FOREIGN KEY';



**ADD CONSTARINT Commands:**

ALTER TABLE `employee\_files` ADD CONSTRAINT `employee\_files\_ibfk\_1` FOREIGN KEY (`emp\_files\_id`) REFERENCES `employees` (`emp\_id`) ON DELETE RESTRICT ON UPDATE RESTRICT;

**Excuted Result (with Query Commands):**

(Excute successfully)



SELECT \* FROM information\_schema.TABLE\_CONSTRAINTS WHERE CONSTRAINT\_SCHEMA = 'human\_resource\_management\_system' AND TABLE\_NAME = 'employee\_files' AND CONSTRAINT\_TYPE = 'FOREIGN KEY';



**ADD CONSTARINT Commands:**

ALTER TABLE `employee\_health\_checkups` ADD CONSTRAINT `FK\_Employee\_Health\_Checkups\_emp\_files\_id` FOREIGN KEY (`emp\_files\_id`) REFERENCES `employee\_files` (`emp\_files\_id`) ON DELETE RESTRICT ON UPDATE RESTRICT;

**Excuted Result (with Query Commands):**

(Excute successfully)



SELECT \* FROM information\_schema.TABLE\_CONSTRAINTS WHERE CONSTRAINT\_SCHEMA = 'human\_resource\_management\_system' AND TABLE\_NAME = 'employee\_health\_checkups' AND CONSTRAINT\_TYPE = 'FOREIGN KEY';



**ADD CONSTARINT Commands:**

ALTER TABLE `employee\_performance\_and\_development` ADD CONSTRAINT `employee\_performance\_and\_development\_ibfk\_1` FOREIGN KEY (`emp\_pnd\_id`) REFERENCES `employees` (`emp\_id`) ON DELETE RESTRICT ON UPDATE RESTRICT;

**Excuted Result (with Query Commands):**

(Excute successfully)



SELECT \* FROM information\_schema.TABLE\_CONSTRAINTS WHERE CONSTRAINT\_SCHEMA = 'human\_resource\_management\_system' AND TABLE\_NAME = 'employee\_performance\_and\_development' AND CONSTRAINT\_TYPE = 'FOREIGN KEY';



**ADD CONSTARINT Commands:**

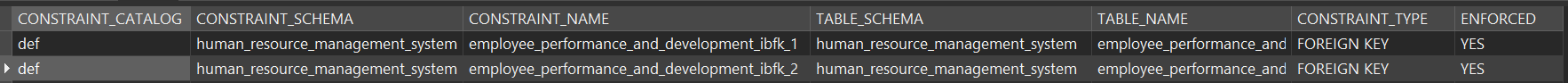
ALTER TABLE `employee\_performance\_and\_development` ADD CONSTRAINT `employee\_performance\_and\_development\_ibfk\_2` FOREIGN KEY (`skillset\_id`) REFERENCES `skillset` (`Skillset\_id`) ON DELETE RESTRICT ON UPDATE RESTRICT;

**Excuted Result (with Query Commands):**

(Excute successfully)



SELECT \* FROM information\_schema.TABLE\_CONSTRAINTS WHERE CONSTRAINT\_SCHEMA = 'human\_resource\_management\_system' AND TABLE\_NAME = 'employee\_performance\_and\_development' AND CONSTRAINT\_TYPE = 'FOREIGN KEY';

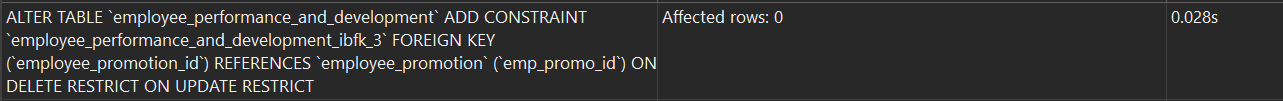


**ADD CONSTARINT Commands:**

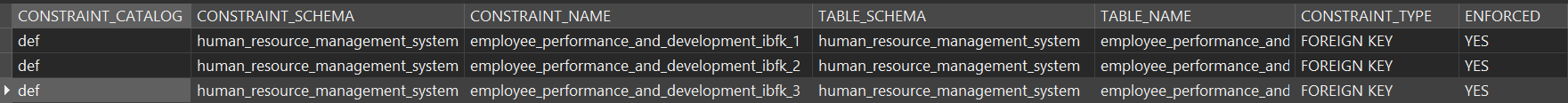
ALTER TABLE `employee\_performance\_and\_development` ADD CONSTRAINT `employee\_performance\_and\_development\_ibfk\_3` FOREIGN KEY (`employee\_promotion\_id`) REFERENCES `employee\_promotion` (`emp\_promo\_id`) ON DELETE RESTRICT ON UPDATE RESTRICT;

**Excuted Result (with Query Commands):**

(Excute successfully)



SELECT \* FROM information\_schema.TABLE\_CONSTRAINTS WHERE CONSTRAINT\_SCHEMA = 'human\_resource\_management\_system' AND TABLE\_NAME = 'employee\_performance\_and\_development' AND CONSTRAINT\_TYPE = 'FOREIGN KEY';

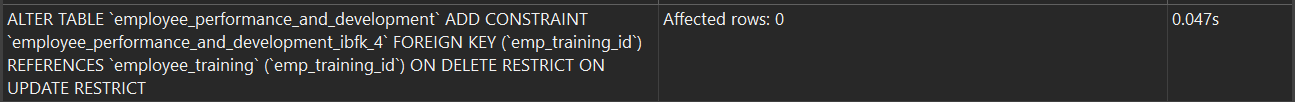


**ADD CONSTARINT Commands:**

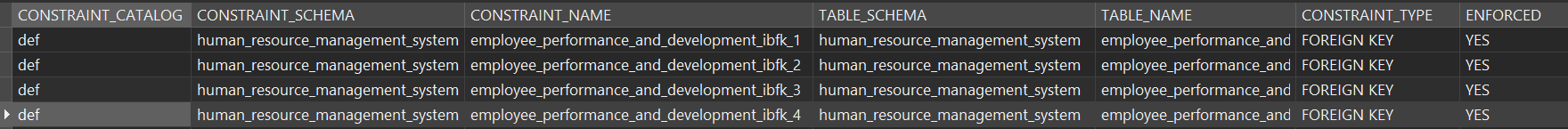
ALTER TABLE `employee\_performance\_and\_development` ADD CONSTRAINT `employee\_performance\_and\_development\_ibfk\_4` FOREIGN KEY (`emp\_training\_id`) REFERENCES `employee\_training` (`emp\_training\_id`) ON DELETE RESTRICT ON UPDATE RESTRICT;

**Excuted Result (with Query Commands):**

(Excute successfully)



SELECT \* FROM information\_schema.TABLE\_CONSTRAINTS WHERE CONSTRAINT\_SCHEMA = 'human\_resource\_management\_system' AND TABLE\_NAME = 'employee\_performance\_and\_development' AND CONSTRAINT\_TYPE = 'FOREIGN KEY';

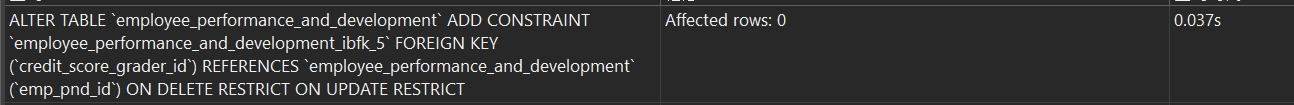


**20 ADD CONSTARINT Commands:**

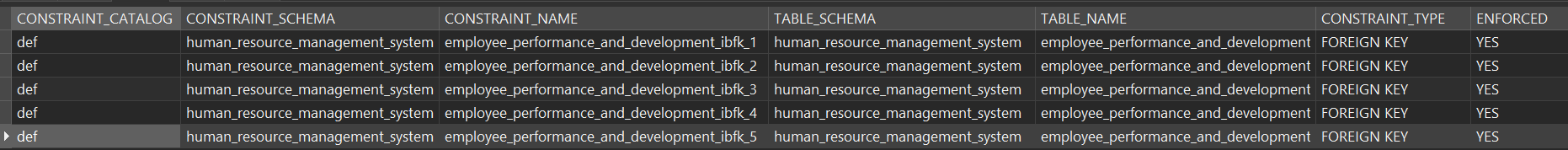
ALTER TABLE `employee\_performance\_and\_development` ADD CONSTRAINT `employee\_performance\_and\_development\_ibfk\_5` FOREIGN KEY (`credit\_score\_grader\_id`) REFERENCES `employee\_performance\_and\_development` (`emp\_pnd\_id`) ON DELETE RESTRICT ON UPDATE RESTRICT;

**Excuted Result (with Query Commands):**

(Excute successfully)



SELECT \* FROM information\_schema.TABLE\_CONSTRAINTS WHERE CONSTRAINT\_SCHEMA = 'human\_resource\_management\_system' AND TABLE\_NAME = 'employee\_performance\_and\_development' AND CONSTRAINT\_TYPE = 'FOREIGN KEY';

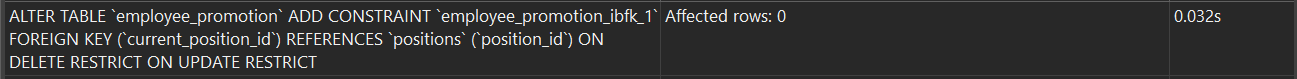


**ADD CONSTARINT Commands:**

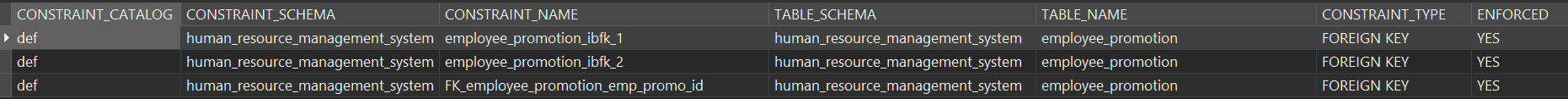
ALTER TABLE `employee\_promotion` ADD CONSTRAINT `employee\_promotion\_ibfk\_1` FOREIGN KEY (`current\_position\_id`) REFERENCES `positions` (`position\_id`) ON DELETE RESTRICT ON UPDATE RESTRICT;

**Excuted Result (with Query Commands):**

(Excute successfully)



SELECT \* FROM information\_schema.TABLE\_CONSTRAINTS WHERE CONSTRAINT\_SCHEMA = 'human\_resource\_management\_system' AND TABLE\_NAME = 'employee\_promotion' AND CONSTRAINT\_TYPE = 'FOREIGN KEY';

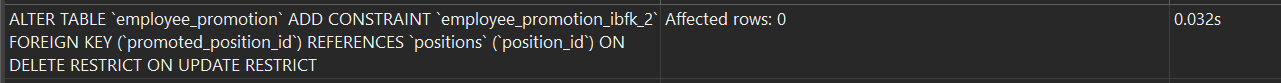


**ADD CONSTARINT Commands:**

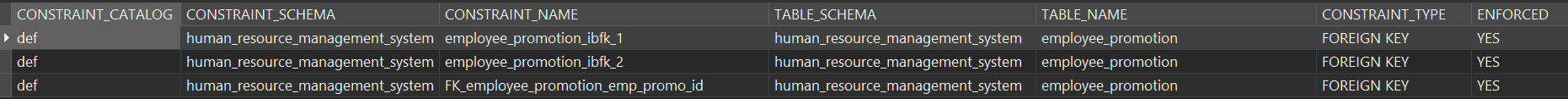
ALTER TABLE `employee\_promotion` ADD CONSTRAINT `employee\_promotion\_ibfk\_2` FOREIGN KEY (`promoted\_position\_id`) REFERENCES `positions` (`position\_id`) ON DELETE RESTRICT ON UPDATE RESTRICT;

**Excuted Result (with Query Commands):**

(Excute successfully)



SELECT \* FROM information\_schema.TABLE\_CONSTRAINTS WHERE CONSTRAINT\_SCHEMA = 'human\_resource\_management\_system' AND TABLE\_NAME = 'employee\_promotion' AND CONSTRAINT\_TYPE = 'FOREIGN KEY';

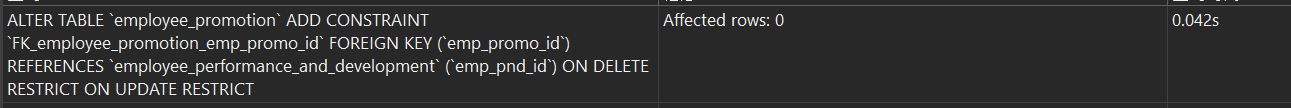


**ADD CONSTARINT Commands:**

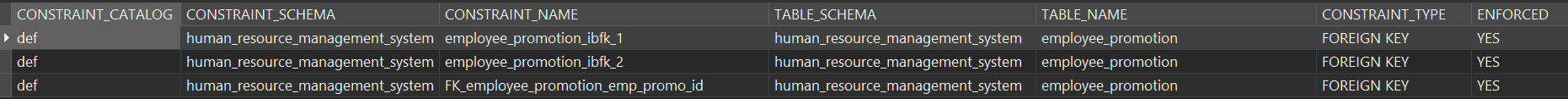
ALTER TABLE `employee\_promotion` ADD CONSTRAINT `FK\_employee\_promotion\_emp\_promo\_id` FOREIGN KEY (`emp\_promo\_id`) REFERENCES `employee\_performance\_and\_development` (`emp\_pnd\_id`) ON DELETE RESTRICT ON UPDATE RESTRICT;

**Excuted Result (with Query Commands):**

(Excute successfully)



SELECT \* FROM information\_schema.TABLE\_CONSTRAINTS WHERE CONSTRAINT\_SCHEMA = 'human\_resource\_management\_system' AND TABLE\_NAME = 'employee\_promotion' AND CONSTRAINT\_TYPE = 'FOREIGN KEY';



**ADD CONSTARINT Commands:**

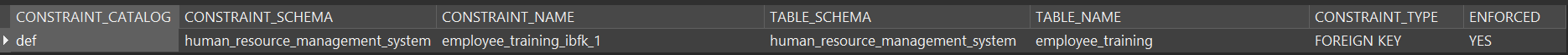
ALTER TABLE `employee\_training` ADD CONSTRAINT `employee\_training\_ibfk\_1` FOREIGN KEY (`training\_skill\_id`) REFERENCES `skills` (`Skill\_id`) ON DELETE RESTRICT ON UPDATE RESTRICT;

**Excuted Result (with Query Commands):**

(Excute successfully)



SELECT \* FROM information\_schema.TABLE\_CONSTRAINTS WHERE CONSTRAINT\_SCHEMA = 'human\_resource\_management\_system' AND TABLE\_NAME = 'employee\_training' AND CONSTRAINT\_TYPE = 'FOREIGN KEY';

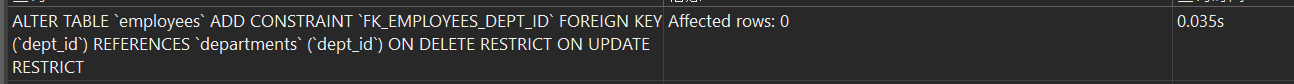


**ADD CONSTARINT Commands:**

ALTER TABLE `employees` ADD CONSTRAINT `FK\_EMPLOYEES\_DEPT\_ID` FOREIGN KEY (`dept\_id`) REFERENCES `departments` (`dept\_id`) ON DELETE RESTRICT ON UPDATE RESTRICT;

**Excuted Result (with Query Commands):**

(Excute successfully)



SELECT \* FROM information\_schema.TABLE\_CONSTRAINTS WHERE CONSTRAINT\_SCHEMA = 'human\_resource\_management\_system' AND TABLE\_NAME = 'employees' AND CONSTRAINT\_TYPE = 'FOREIGN KEY';



**ADD CONSTARINT Commands:**

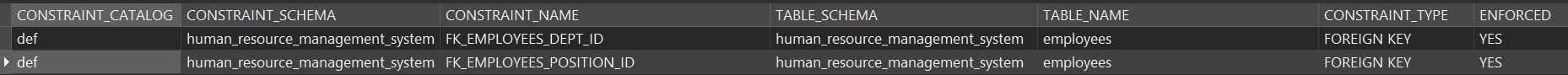
ALTER TABLE `employees` ADD CONSTRAINT `FK\_EMPLOYEES\_POSITION\_ID` FOREIGN KEY (`position\_id`) REFERENCES `positions` (`position\_id`) ON DELETE RESTRICT ON UPDATE RESTRICT;

**Excuted Result (with Query Commands):**

(Excute successfully)



SELECT \* FROM information\_schema.TABLE\_CONSTRAINTS WHERE CONSTRAINT\_SCHEMA = 'human\_resource\_management\_system' AND TABLE\_NAME = 'employees' AND CONSTRAINT\_TYPE = 'FOREIGN KEY';



**ADD CONSTARINT Commands:**

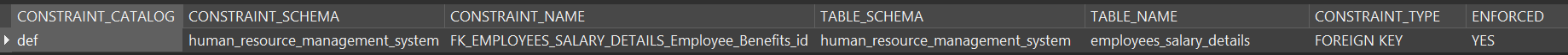
ALTER TABLE `employees\_salary\_details` ADD CONSTRAINT `FK\_EMPLOYEES\_SALARY\_DETAILS\_Employee\_Benefits\_id` FOREIGN KEY (`Employee\_Benefits\_id`) REFERENCES `employee\_benefits` (`Employee\_Benefits\_id`) ON DELETE RESTRICT ON UPDATE RESTRICT;

**Excuted Result (with Query Commands):**

(Excute successfully)



SELECT \* FROM information\_schema.TABLE\_CONSTRAINTS WHERE CONSTRAINT\_SCHEMA = 'human\_resource\_management\_system' AND TABLE\_NAME = 'employees\_salary\_details' AND CONSTRAINT\_TYPE = 'FOREIGN KEY';

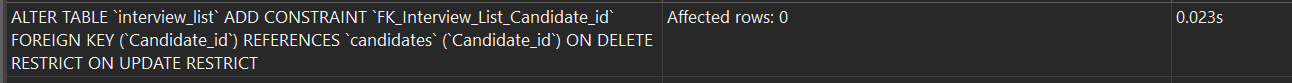


**ADD CONSTARINT Commands:**

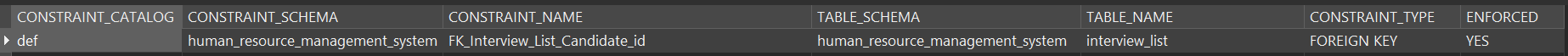
ALTER TABLE `interview\_list` ADD CONSTRAINT `FK\_Interview\_List\_Candidate\_id` FOREIGN KEY (`Candidate\_id`) REFERENCES `candidates` (`Candidate\_id`) ON DELETE RESTRICT ON UPDATE RESTRICT;

**Excuted Result (with Query Commands):**

(Excute successfully)



SELECT \* FROM information\_schema.TABLE\_CONSTRAINTS WHERE CONSTRAINT\_SCHEMA = 'human\_resource\_management\_system' AND TABLE\_NAME = 'interview\_list' AND CONSTRAINT\_TYPE = 'FOREIGN KEY';



**ADD CONSTARINT Commands:**

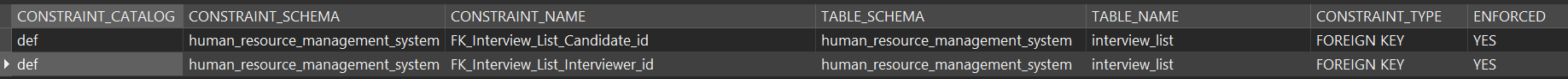
ALTER TABLE `interview\_list` ADD CONSTRAINT `FK\_Interview\_List\_Interviewer\_id` FOREIGN KEY (`Interviewer\_id`) REFERENCES `employees` (`emp\_id`) ON DELETE RESTRICT ON UPDATE RESTRICT;

**Excuted Result (with Query Commands):**

(Excute successfully)



SELECT \* FROM information\_schema.TABLE\_CONSTRAINTS WHERE CONSTRAINT\_SCHEMA = 'human\_resource\_management\_system' AND TABLE\_NAME = 'interview\_list' AND CONSTRAINT\_TYPE = 'FOREIGN KEY';

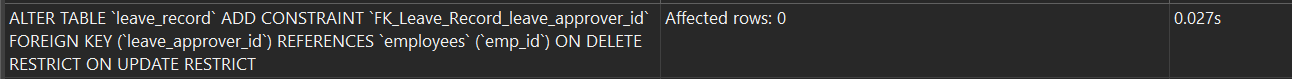


**30 ADD CONSTARINT Commands:**

ALTER TABLE `leave\_record` ADD CONSTRAINT `FK\_Leave\_Record\_leave\_approver\_id` FOREIGN KEY (`leave\_approver\_id`) REFERENCES `employees` (`emp\_id`) ON DELETE RESTRICT ON UPDATE RESTRICT;

**Excuted Result (with Query Commands):**

(Excute successfully)



SELECT \* FROM information\_schema.TABLE\_CONSTRAINTS WHERE CONSTRAINT\_SCHEMA = 'human\_resource\_management\_system' AND TABLE\_NAME = 'leave\_record' AND CONSTRAINT\_TYPE = 'FOREIGN KEY';

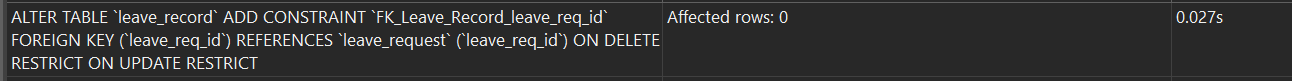


**ADD CONSTARINT Commands:**

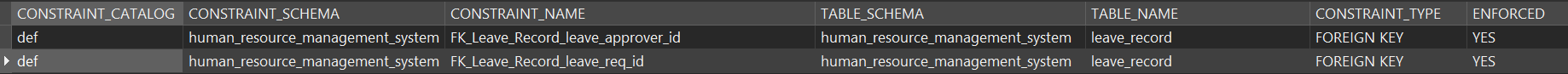
ALTER TABLE `leave\_record` ADD CONSTRAINT `FK\_Leave\_Record\_leave\_req\_id` FOREIGN KEY (`leave\_req\_id`) REFERENCES `leave\_request` (`leave\_req\_id`) ON DELETE RESTRICT ON UPDATE RESTRICT;

**Excuted Result (with Query Commands):**

(Excute successfully)



SELECT \* FROM information\_schema.TABLE\_CONSTRAINTS WHERE CONSTRAINT\_SCHEMA = 'human\_resource\_management\_system' AND TABLE\_NAME = 'leave\_record' AND CONSTRAINT\_TYPE = 'FOREIGN KEY';

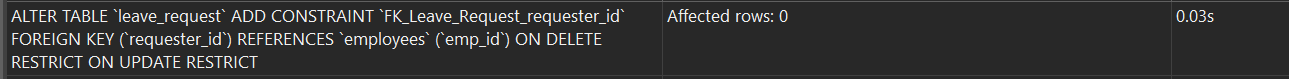


**ADD CONSTARINT Commands:**

ALTER TABLE `leave\_request` ADD CONSTRAINT `FK\_Leave\_Request\_requester\_id` FOREIGN KEY (`requester\_id`) REFERENCES `employees` (`emp\_id`) ON DELETE RESTRICT ON UPDATE RESTRICT;

**Excuted Result (with Query Commands):**

(Excute successfully)



SELECT \* FROM information\_schema.TABLE\_CONSTRAINTS WHERE CONSTRAINT\_SCHEMA = 'human\_resource\_management\_system' AND TABLE\_NAME = 'leave\_request' AND CONSTRAINT\_TYPE = 'FOREIGN KEY';

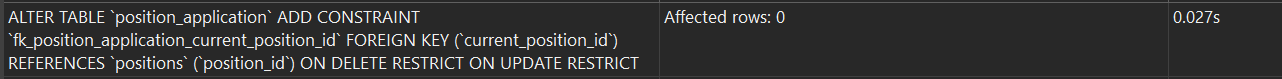


**ADD CONSTARINT Commands:**

ALTER TABLE `position\_application` ADD CONSTRAINT `fk\_position\_application\_current\_position\_id` FOREIGN KEY (`current\_position\_id`) REFERENCES `positions` (`position\_id`) ON DELETE RESTRICT ON UPDATE RESTRICT;

**Excuted Result (with Query Commands):**

(Excute successfully)



SELECT \* FROM information\_schema.TABLE\_CONSTRAINTS WHERE CONSTRAINT\_SCHEMA = 'human\_resource\_management\_system' AND TABLE\_NAME = 'position\_application' AND CONSTRAINT\_TYPE = 'FOREIGN KEY';



**ADD CONSTARINT Commands:**

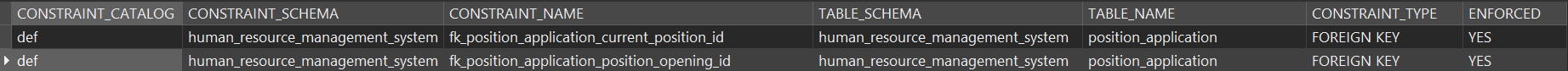
ALTER TABLE `position\_application` ADD CONSTRAINT `fk\_position\_application\_position\_opening\_id` FOREIGN KEY (`position\_opening\_id`) REFERENCES `position\_opening` (`Position\_Opening\_id`) ON DELETE RESTRICT ON UPDATE RESTRICT;

**Excuted Result (with Query Commands):**

(Excute successfully)



SELECT \* FROM information\_schema.TABLE\_CONSTRAINTS WHERE CONSTRAINT\_SCHEMA = 'human\_resource\_management\_system' AND TABLE\_NAME = 'position\_application' AND CONSTRAINT\_TYPE = 'FOREIGN KEY';

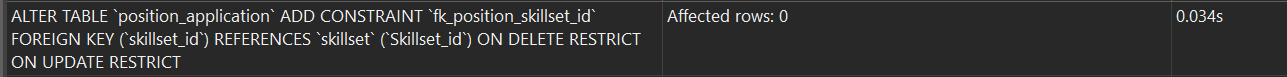
****

**ADD CONSTARINT Commands:**

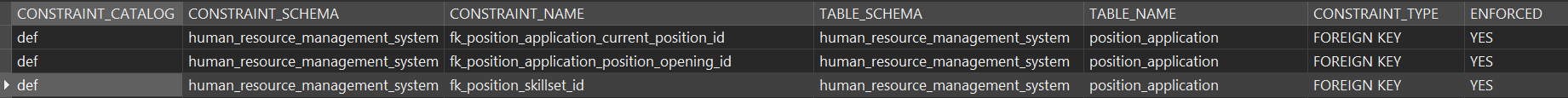
ALTER TABLE `position\_application` ADD CONSTRAINT `fk\_position\_skillset\_id` FOREIGN KEY (`skillset\_id`) REFERENCES `skillset` (`Skillset\_id`) ON DELETE RESTRICT ON UPDATE RESTRICT;

**Excuted Result (with Query Commands):**

(Excute successfully)



SELECT \* FROM information\_schema.TABLE\_CONSTRAINTS WHERE CONSTRAINT\_SCHEMA = 'human\_resource\_management\_system' AND TABLE\_NAME = 'position\_application' AND CONSTRAINT\_TYPE = 'FOREIGN KEY';

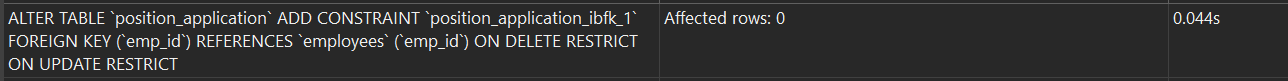


**ADD CONSTARINT Commands:**

ALTER TABLE `position\_application` ADD CONSTRAINT `position\_application\_ibfk\_1` FOREIGN KEY (`emp\_id`) REFERENCES `employees` (`emp\_id`) ON DELETE RESTRICT ON UPDATE RESTRICT;

**Excuted Result (with Query Commands):**

(Excute successfully)



SELECT \* FROM information\_schema.TABLE\_CONSTRAINTS WHERE CONSTRAINT\_SCHEMA = 'human\_resource\_management\_system' AND TABLE\_NAME = 'position\_application' AND CONSTRAINT\_TYPE = 'FOREIGN KEY';

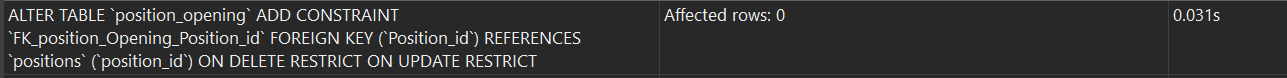


**ADD CONSTARINT Commands:**

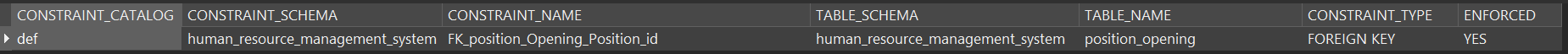
ALTER TABLE `position\_opening` ADD CONSTRAINT `FK\_position\_Opening\_Position\_id` FOREIGN KEY (`Position\_id`) REFERENCES `positions` (`position\_id`) ON DELETE RESTRICT ON UPDATE RESTRICT;

**Excuted Result (with Query Commands):**

(Excute successfully)



SELECT \* FROM information\_schema.TABLE\_CONSTRAINTS WHERE CONSTRAINT\_SCHEMA = 'human\_resource\_management\_system' AND TABLE\_NAME = 'position\_opening' AND CONSTRAINT\_TYPE = 'FOREIGN KEY';

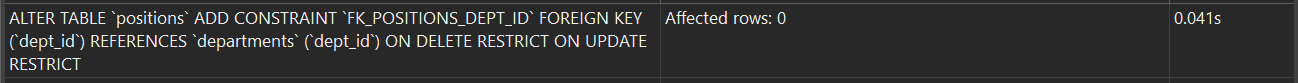


**ADD CONSTARINT Commands:**

ALTER TABLE `positions` ADD CONSTRAINT `FK\_POSITIONS\_DEPT\_ID` FOREIGN KEY (`dept\_id`) REFERENCES `departments` (`dept\_id`) ON DELETE RESTRICT ON UPDATE RESTRICT;

**Excuted Result (with Query Commands):**

(Excute successfully)



SELECT \* FROM information\_schema.TABLE\_CONSTRAINTS WHERE CONSTRAINT\_SCHEMA = 'human\_resource\_management\_system' AND TABLE\_NAME = 'positions' AND CONSTRAINT\_TYPE = 'FOREIGN KEY';

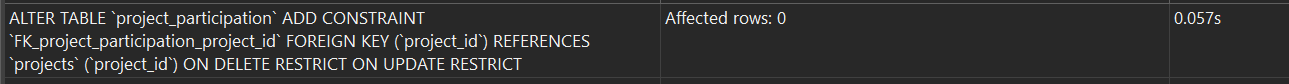


**ADD CONSTARINT Commands:**

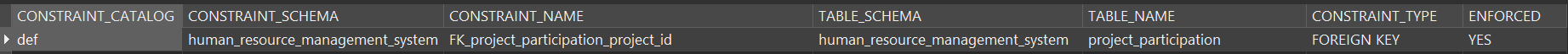
ALTER TABLE `project\_participation` ADD CONSTRAINT `FK\_project\_participation\_project\_id` FOREIGN KEY (`project\_id`) REFERENCES `projects` (`project\_id`) ON DELETE RESTRICT ON UPDATE RESTRICT;

**Excuted Result (with Query Commands):**

(Excute successfully)



SELECT \* FROM information\_schema.TABLE\_CONSTRAINTS WHERE CONSTRAINT\_SCHEMA = 'human\_resource\_management\_system' AND TABLE\_NAME = 'project\_participation' AND CONSTRAINT\_TYPE = 'FOREIGN KEY';

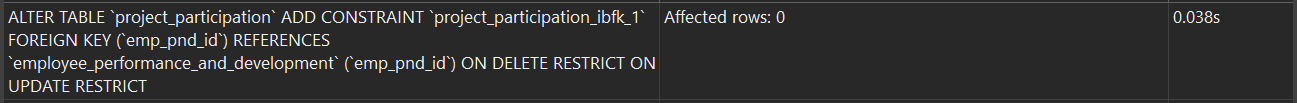


**40 ADD CONSTARINT Commands:**

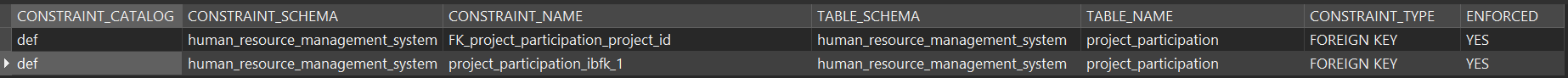
ALTER TABLE `project\_participation` ADD CONSTRAINT `project\_participation\_ibfk\_1` FOREIGN KEY (`emp\_pnd\_id`) REFERENCES `employee\_performance\_and\_development` (`emp\_pnd\_id`) ON DELETE RESTRICT ON UPDATE RESTRICT;

**Excuted Result (with Query Commands):**

(Excute successfully)



SELECT \* FROM information\_schema.TABLE\_CONSTRAINTS WHERE CONSTRAINT\_SCHEMA = 'human\_resource\_management\_system' AND TABLE\_NAME = 'project\_participation' AND CONSTRAINT\_TYPE = 'FOREIGN KEY';

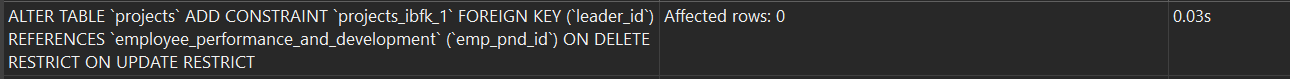


**ADD CONSTARINT Commands:**

ALTER TABLE `projects` ADD CONSTRAINT `projects\_ibfk\_1` FOREIGN KEY (`leader\_id`) REFERENCES `employee\_performance\_and\_development` (`emp\_pnd\_id`) ON DELETE RESTRICT ON UPDATE RESTRICT;

**Excuted Result (with Query Commands):**

(Excute successfully)



SELECT \* FROM information\_schema.TABLE\_CONSTRAINTS WHERE CONSTRAINT\_SCHEMA = 'human\_resource\_management\_system' AND TABLE\_NAME = 'projects' AND CONSTRAINT\_TYPE = 'FOREIGN KEY';



**ADD CONSTARINT Commands:**

ALTER TABLE `resignations` ADD CONSTRAINT `resignations\_ibfk\_1` FOREIGN KEY (`resign\_emp\_id`) REFERENCES `employee\_files` (`emp\_files\_id`) ON DELETE RESTRICT ON UPDATE RESTRICT;

**Excuted Result (with Query Commands):**

(Excute successfully)



SELECT \* FROM information\_schema.TABLE\_CONSTRAINTS WHERE CONSTRAINT\_SCHEMA = 'human\_resource\_management\_system' AND TABLE\_NAME = 'resignations' AND CONSTRAINT\_TYPE = 'FOREIGN KEY';



**ADD CONSTARINT Commands:**

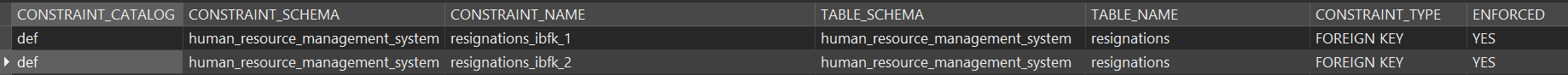
ALTER TABLE `resignations` ADD CONSTRAINT `resignations\_ibfk\_2` FOREIGN KEY (`related\_contract\_id`) REFERENCES `contracts` (`contract\_id`) ON DELETE RESTRICT ON UPDATE RESTRICT;

**Excuted Result (with Query Commands):**

(Excute successfully)



SELECT \* FROM information\_schema.TABLE\_CONSTRAINTS WHERE CONSTRAINT\_SCHEMA = 'human\_resource\_management\_system' AND TABLE\_NAME = 'resignations' AND CONSTRAINT\_TYPE = 'FOREIGN KEY';



**ADD CONSTARINT Commands:**

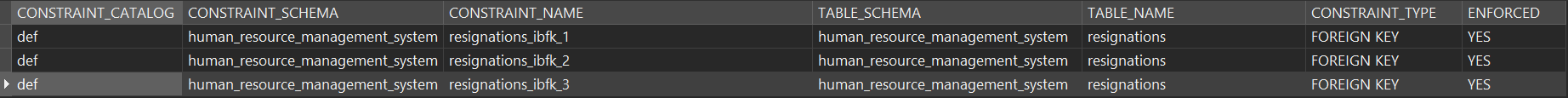
ALTER TABLE `resignations` ADD CONSTRAINT `resignations\_ibfk\_3` FOREIGN KEY (`exit\_interviewer\_id`) REFERENCES `employees` (`emp\_id`) ON DELETE RESTRICT ON UPDATE RESTRICT;

**Excuted Result (with Query Commands):**

(Excute successfully)



SELECT \* FROM information\_schema.TABLE\_CONSTRAINTS WHERE CONSTRAINT\_SCHEMA = 'human\_resource\_management\_system' AND TABLE\_NAME = 'resignations' AND CONSTRAINT\_TYPE = 'FOREIGN KEY';

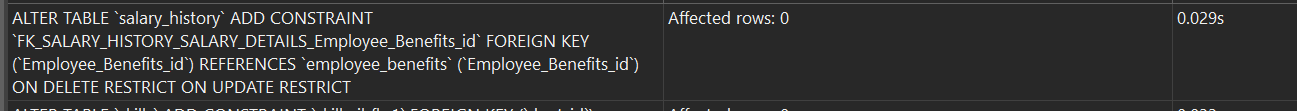


**ADD CONSTARINT Commands:**

ALTER TABLE `salary\_history` ADD CONSTRAINT `FK\_SALARY\_HISTORY\_SALARY\_DETAILS\_Employee\_Benefits\_id` FOREIGN KEY (`Employee\_Benefits\_id`) REFERENCES `employee\_benefits` (`Employee\_Benefits\_id`) ON DELETE RESTRICT ON UPDATE RESTRICT;

**Excuted Result (with Query Commands):**

(Excute successfully)



SELECT \* FROM information\_schema.TABLE\_CONSTRAINTS WHERE CONSTRAINT\_SCHEMA = 'human\_resource\_management\_system' AND TABLE\_NAME = 'salary\_history' AND CONSTRAINT\_TYPE = 'FOREIGN KEY';



**ADD CONSTARINT Commands:**

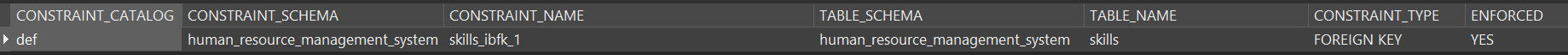
ALTER TABLE `skills` ADD CONSTRAINT `skills\_ibfk\_1` FOREIGN KEY (`dept\_id`) REFERENCES `departments` (`dept\_id`) ON DELETE RESTRICT ON UPDATE RESTRICT;

**Excuted Result (with Query Commands):**

(Excute successfully)



SELECT \* FROM information\_schema.TABLE\_CONSTRAINTS WHERE CONSTRAINT\_SCHEMA = 'human\_resource\_management\_system' AND TABLE\_NAME = 'skills' AND CONSTRAINT\_TYPE = 'FOREIGN KEY';

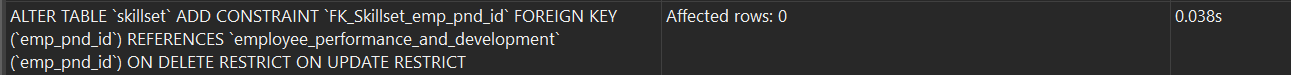


**ADD CONSTARINT Commands:**

ALTER TABLE `skillset` ADD CONSTRAINT `FK\_Skillset\_emp\_pnd\_id` FOREIGN KEY (`emp\_pnd\_id`) REFERENCES `employee\_performance\_and\_development` (`emp\_pnd\_id`) ON DELETE RESTRICT ON UPDATE RESTRICT;

**Excuted Result (with Query Commands):**

(Excute successfully)



SELECT \* FROM information\_schema.TABLE\_CONSTRAINTS WHERE CONSTRAINT\_SCHEMA = 'human\_resource\_management\_system' AND TABLE\_NAME = 'skillset' AND CONSTRAINT\_TYPE = 'FOREIGN KEY';



**ADD CONSTARINT Commands:**

ALTER TABLE `skillset` ADD CONSTRAINT `skillset\_ibfk\_1` FOREIGN KEY (`candidate\_id`) REFERENCES `candidates` (`Candidate\_id`) ON DELETE RESTRICT ON UPDATE RESTRICT;

**Excuted Result (with Query Commands):**

(Excute successfully)



SELECT \* FROM information\_schema.TABLE\_CONSTRAINTS WHERE CONSTRAINT\_SCHEMA = 'human\_resource\_management\_system' AND TABLE\_NAME = 'skillset' AND CONSTRAINT\_TYPE = 'FOREIGN KEY';



**ADD CONSTARINT Commands:**

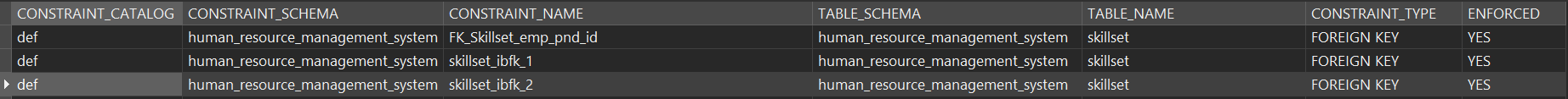
ALTER TABLE `skillset` ADD CONSTRAINT `skillset\_ibfk\_2` FOREIGN KEY (`Skill\_one\_id`) REFERENCES `skills` (`Skill\_id`) ON DELETE RESTRICT ON UPDATE RESTRICT;

**Excuted Result (with Query Commands):**

(Excute successfully)



SELECT \* FROM information\_schema.TABLE\_CONSTRAINTS WHERE CONSTRAINT\_SCHEMA = 'human\_resource\_management\_system' AND TABLE\_NAME = 'skillset' AND CONSTRAINT\_TYPE = 'FOREIGN KEY';



**50 ADD CONSTARINT Commands:**

ALTER TABLE `skillset` ADD CONSTRAINT `skillset\_ibfk\_3` FOREIGN KEY (`Skill\_two\_id`) REFERENCES `skills` (`Skill\_id`) ON DELETE RESTRICT ON UPDATE RESTRICT;

**Excuted Result (with Query Commands):**

(Excute successfully)



SELECT \* FROM information\_schema.TABLE\_CONSTRAINTS WHERE CONSTRAINT\_SCHEMA = 'human\_resource\_management\_system' AND TABLE\_NAME = 'skillset' AND CONSTRAINT\_TYPE = 'FOREIGN KEY';



**ADD CONSTARINT Commands:**

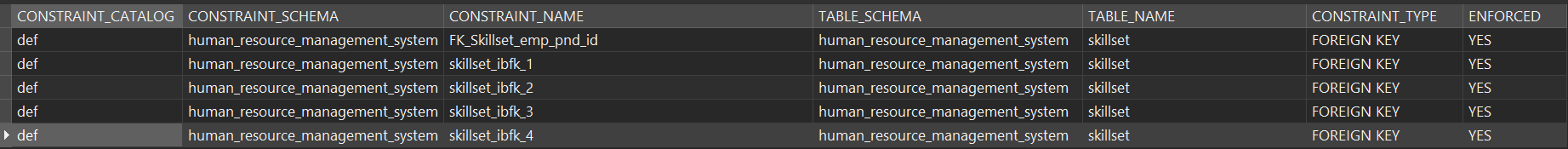
ALTER TABLE `skillset` ADD CONSTRAINT `skillset\_ibfk\_4` FOREIGN KEY (`Skill\_three\_id`) REFERENCES `skills` (`Skill\_id`) ON DELETE RESTRICT ON UPDATE RESTRICT;

**Excuted Result (with Query Commands):**

(Excute successfully)



SELECT \* FROM information\_schema.TABLE\_CONSTRAINTS WHERE CONSTRAINT\_SCHEMA = 'human\_resource\_management\_system' AND TABLE\_NAME = 'skillset' AND CONSTRAINT\_TYPE = 'FOREIGN KEY';



**第52个..ADD CONSTARINT Commands:**

ALTER TABLE `users` ADD CONSTRAINT `users\_ibfk\_1` FOREIGN KEY (`emp\_id`) REFERENCES `employees` (`emp\_id`) ON DELETE RESTRICT ON UPDATE RESTRICT;

**Excuted Result (with Query Commands):**

(Excute successfully)



SELECT \* FROM information\_schema.TABLE\_CONSTRAINTS

WHERE CONSTRAINT\_SCHEMA = 'human\_resource\_management\_system' AND TABLE\_NAME = 'users' AND CONSTRAINT\_TYPE = 'FOREIGN KEY';

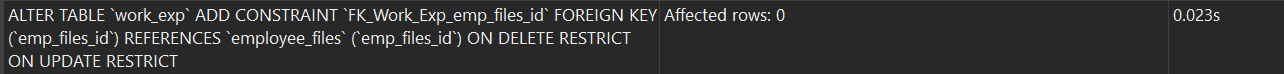


**53 ADD CONSTARINT Commands:**

ALTER TABLE `work\_exp` ADD CONSTRAINT `FK\_Work\_Exp\_emp\_files\_id` FOREIGN KEY (`emp\_files\_id`) REFERENCES `employee\_files` (`emp\_files\_id`) ON DELETE RESTRICT ON UPDATE RESTRICT;

**Excuted Result (with Query Commands):**

(Excute successfully)



SELECT \* FROM information\_schema.TABLE\_CONSTRAINTS WHERE CONSTRAINT\_SCHEMA = 'human\_resource\_management\_system' AND TABLE\_NAME = 'work\_exp' AND CONSTRAINT\_TYPE = 'FOREIGN KEY';



**DROP CONSTRAINT**

**DROP CONSTRAINT Commands:**

ALTER TABLE users DROP CONSTRAINT users\_ibfk\_1;

**Excuted Result (with Query Commands):**

(Excute successfully)

****

SELECT \* FROM information\_schema.TABLE\_CONSTRAINTS

WHERE CONSTRAINT\_SCHEMA = 'human\_resource\_management\_system' AND TABLE\_NAME = 'users' AND CONSTRAINT\_TYPE = 'FOREIGN KEY';

***Before:***



***After:***



**2.2.3 RENAME**

**RENAME TABLE**

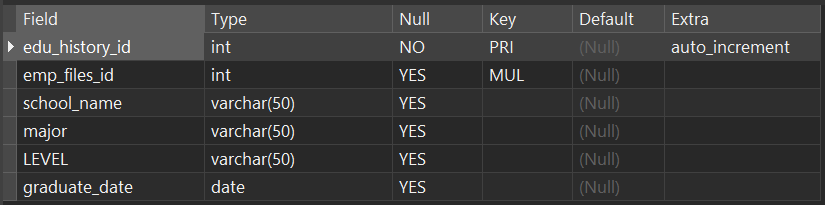
**Commands:**

RENAME TABLE edu\_history TO education\_history;

**Excuted Result (with Query Commands):**

DESC edu\_history;

***Before:***



***After:***

****

**RENAME COLUMN**

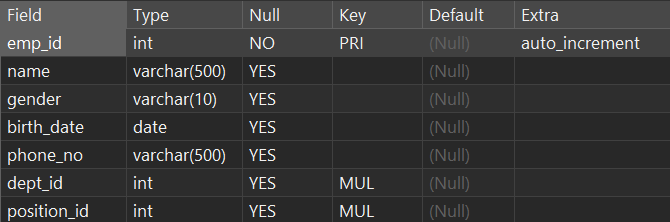
**Commands:**

ALTER TABLE employees RENAME COLUMN emp\_id TO employee\_id;

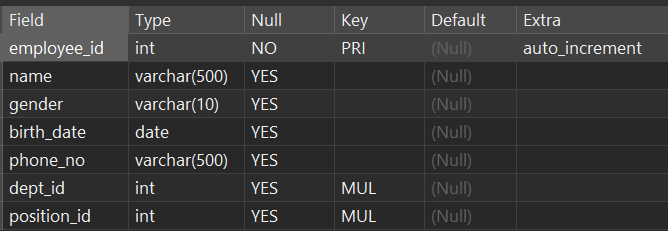
**Excuted Result (with Query Commands):**

DESC employees;

***Before:***

******

***After:***



**2.3 Data Manipulation Language (DML)**

**2.3.1 INSERT INTO**

- Through the INSERT INTO statement, we can add new rows to an existing table in order to store related data in the database. This function is very useful to ensure that all kinds of information in the human resource management system are accurately recorded and saved.

- The following commands will be used to insert relevant data into the existing tables to ensure that the basic information of employees, attendance records, performance evaluation and other information can be included in the system in a complete and timely manner.

**INSERT INTO Commands:**

-- INSERT DATA INTO TABLE positions

INSERT INTO positions (position\_name, avg\_salary, dept\_id, emp\_count) VALUES

('Sales Manager', 70000.00, NULL, 1),

('Sales Representative', 45000.00, NULL, 8),

('Sales Associate', 38000.00, NULL, 10),

('Sales Support Specialist', 42000.00, NULL, 4),

('IT Manager', 65000.00, NULL, 1),

('Software Engineer', 58000.00, NULL, 7),

('Systems Analyst', 52000.00, NULL, 5),

('IT Support Specialist', 42000.00, NULL, 6),

('HR Manager', 60000.00, NULL, 1),

('HR Assistant', 45000.00, NULL, 3),

('HR Specialist', 48000.00, NULL, 5),

('HR System Administrator', 40000.00, NULL, 4),

('Marketing Manager', 60000.00, NULL, 5),

('Marketing Specialist', 45000.00, NULL, 8),

('Social Media Manager', 50000.00, NULL, 3),

('Marketing Analyst', 55000.00, NULL, 6),

('Finance Manager', 62000.00, NULL, 1),

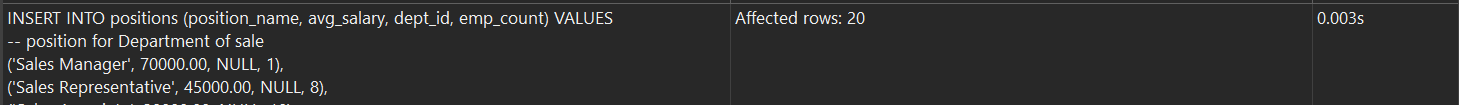
('Financial Analyst', 55000.00, NULL, 3),

('Accountant', 48000.00, NULL, 6),

('Auditor', 53000.00, NULL, 2);

**Excuted Result (with Query Commands, and data will be updated in the UPDATE section later):**

(Excute successfully)

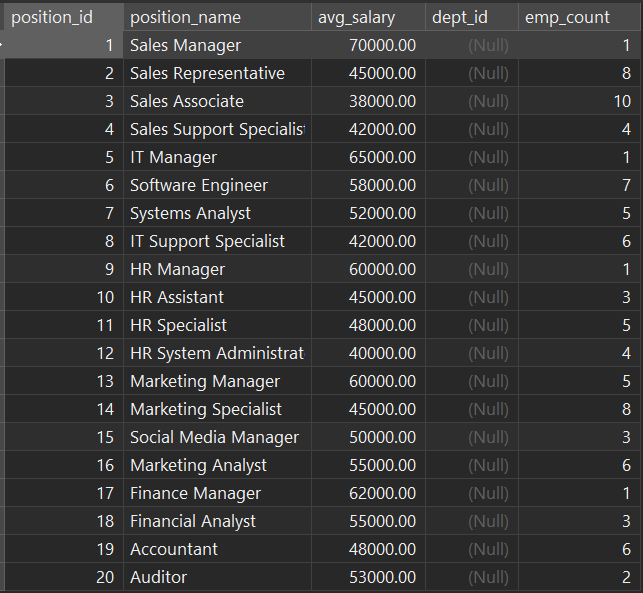
****

SELECT \* FROM positions;

***Before:***



***After:***

****

**INSERT INTO Commands:**

-- INSERT DATA INTO TABLE departments

INSERT INTO departments(dept\_name, dept\_manager, location, emp\_count) VALUES

("Department of sale", NULL, "floor1", 10),

("Department of information technology", NULL, "floor2", 9),

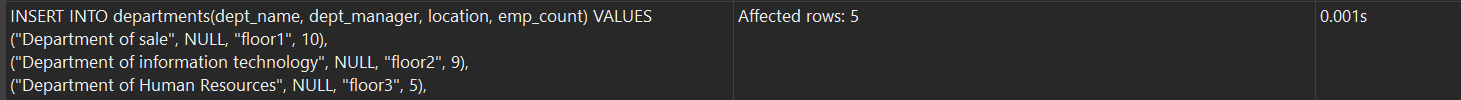
("Department of Human Resources", NULL, "floor3", 5),

("Department of Marketing", NULL, "floor4", 5),

("Department of Finance", NULL, "floor5", 4);

**Excuted Result (with Query Commands, and data will be updated in the UPDATE section later):**

(Excute successfully)

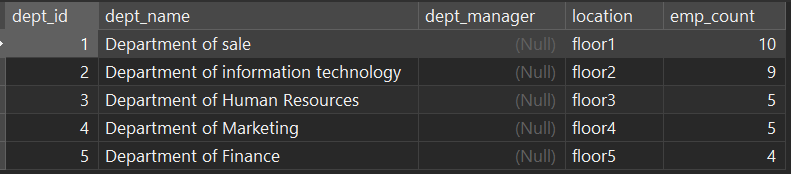
****

SELECT \* FROM departments;

***Before:***

******

***After:***

****

**INSERT INTO Commands:**

INSERT INTO employees (name, gender, birth\_date, phone\_no, dept\_id, position\_id) VALUES

('John Doe', 'Male', '1990-05-15', '011-12345678', 3, 9),

('Jane Smith', 'Female', '1985-08-22', '012-98765432', 1, 1),

('Michael Johnson', 'Male', '1988-11-30', '013-55555555', 2, 5),

('Emily Brown', 'Female', '1992-03-25', '014-77777777', 4, 13),

('David Lee', 'Male', '1987-09-18', '015-88888888', 5, 17),

('Sarah Wang', 'Female', '1991-07-10', '016-99999999', 1, 3),

('Robert Chen', 'Male', '1986-12-05', '017-66666666', 2, 7),

('Jessica Liu', 'Female', '1989-04-12', '018-11111111', 3, 12),

('William Zhang', 'Male', '1993-02-08', '019-22222222', 4, 16),

('Linda Wu', 'Female', '1994-01-01', '010-44444444', 5, 20),

('Alex Johnson', 'Male', '1992-08-11', '011-12345678', 1, 2),

('Ella Williams', 'Female', '1990-05-22', '011-98765432', 2, 6),

('Noah Smith', 'Male', '1988-11-19', '011-56789012', 1, 3),

('Ava Davis', 'Female', '1995-03-25', '011-24681357', 3, 10),

('Liam Taylor', 'Male', '1994-09-18', '011-11223344', 2, 7),

('Mia Brown', 'Female', '1991-07-11', '011-55443322', 1, 4),

('Jameson Lee', 'Male', '1989-12-29', '011-98761234', 3, 11),

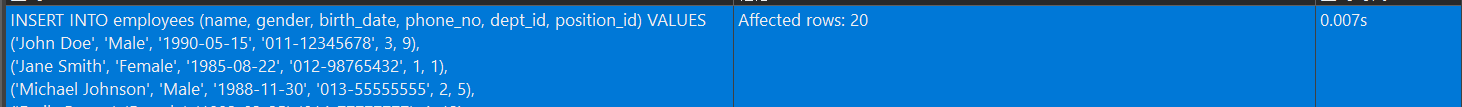
('Aria Wilson', 'Female', '1993-04-06', '011-66778899', 2, 8),

('Ethan Miller', 'Male', '1996-06-30', '011-13579135', 1, 2),

('Scarlett Martin', 'Female', '1987-02-14', '011-11235813', 3, 12);

**Excuted Result (with Query Commands):**

(Excute successfully)



SELECT \* FROM employees;

***Before:***



***After:***

****

**INSERT INTO Commands:**

-- INSERT DATA INTO TABLE employee\_files

INSERT INTO employee\_files (file\_name, file\_size, file\_path, upload\_date) VALUES

('Resume\_John\_Doe.pdf', '2.3 MB', '/uploads/resumes/', '2023-05-10'),

('Profile\_Picture\_Jane\_Smith.jpg', '300 KB', '/uploads/pictures/', '2023-05-11'),

('Performance\_Review\_James\_Brown.docx', '1.8 MB', '/uploads/reviews/', '2023-05-12'),

('Contract\_Annexure\_Alice\_Johnson.pdf', '800 KB', '/uploads/contracts/', '2023-05-13'),

('Training\_Certificate\_Robert\_Lee.pdf', '1.5 MB', '/uploads/certificates/', '2023-05-14'),

('Promotion\_Letter\_Susan\_Wilson.pdf', '1.2 MB', '/uploads/letters/', '2023-05-15'),

('Offer\_Letter\_Michael\_Chen.pdf', '900 KB', '/uploads/offers/', '2023-05-16'),

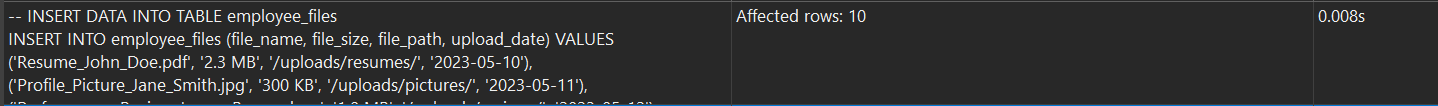
('Resignation\_Letter\_Linda\_Wang.pdf', '700 KB', '/uploads/resignations/', '2023-05-17'),

('Leave\_Application\_Kevin\_Lin.pdf', '600 KB', '/uploads/leave/', '2023-05-18'),

('Training\_Plan\_Mary\_Lee.docx', '2.0 MB', '/uploads/training/', '2023-05-19');

**Excuted Result (with Query Commands):**

(Excute successfully)

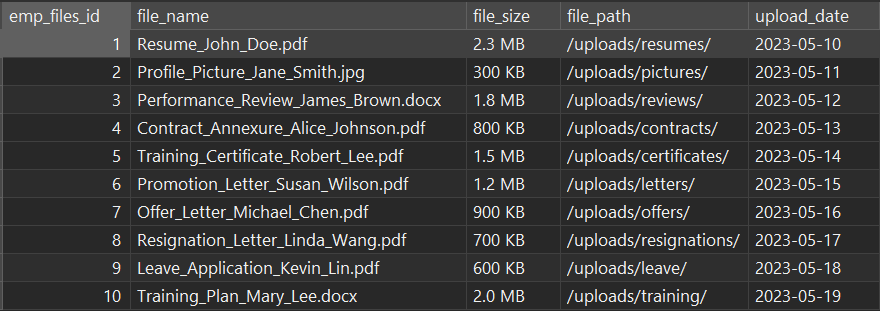


SELECT \* FROM employee\_files;

***Before:***



***After:***

****

**INSERT INTO Commands:**

-- INSERT DATA INTO TABLE work\_exp

INSERT INTO work\_exp (emp\_files\_id, company\_name, POSITION, start\_date, end\_date) VALUES

(1, 'ABC Corporation', 'Software Engineer', '2020-01-05', '2021-02-28'),

(2, 'XYZ Tech', 'Web Developer', '2019-03-15', '2020-12-31'),

(3, 'Acme Solutions', 'Project Manager', '2018-06-10', '2019-12-15'),

(4, 'Tech Innovators', 'Data Analyst', '2022-03-01', '2022-08-31'),

(5, 'Global Industries', 'Marketing Specialist', '2021-05-20', '2022-03-10'),

(6, 'Software Solutions', 'Software Developer', '2020-08-10', '2021-11-30'),

(7, 'Innovative Tech', 'Systems Administrator', '2019-02-15', '2020-05-31'),

(8, 'Tech Solutions', 'Network Engineer', '2022-01-01', '2022-09-30'),

(9, 'Data Systems', 'Database Administrator', '2017-09-12', '2019-01-15'),

(10, 'Digital Marketing Co.', 'Marketing Manager', '2021-03-05', '2022-06-30');

**Excuted Result (with Query Commands):**

(Excute successfully)



SELECT \* FROM work\_exp;

***Before:***

******

***After:***

****

**INSERT INTO Commands:**

-- INSERT DATA INTO TABLE employee\_health\_checkups

INSERT INTO employee\_health\_checkups (emp\_files\_id, Check\_date, height, weight, blood\_pressure) VALUES

(1, '2022-05-15', '170 cm', '65 kg', '120/80 mmHg'),

(2, '2021-12-30', '165 cm', '55 kg', '110/70 mmHg'),

(3, '2023-02-20', '175 cm', '75 kg', '130/85 mmHg'),

(4, '2022-08-10', '160 cm', '50 kg', '115/75 mmHg'),

(5, '2021-06-25', '180 cm', '85 kg', '140/90 mmHg'),

(6, '2023-01-05', '172 cm', '70 kg', '125/80 mmHg'),

(7, '2022-03-12', '168 cm', '60 kg', '118/78 mmHg'),

(8, '2021-11-18', '178 cm', '80 kg', '135/85 mmHg'),

(9, '2022-01-30', '162 cm', '53 kg', '112/72 mmHg'),

(10, '2023-04-05', '185 cm', '90 kg', '145/95 mmHg');

**Excuted Result (with Query Commands):**

(Excute successfully)

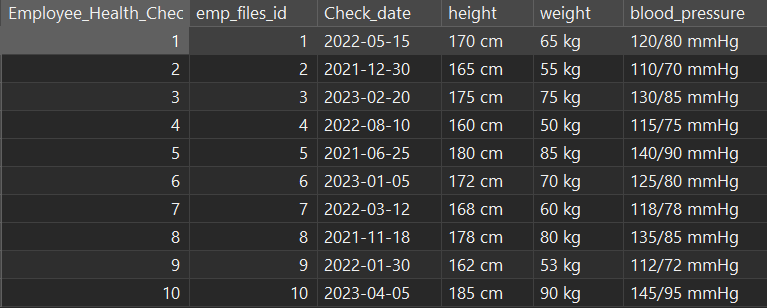


SELECT \* FROM employee\_health\_checkups;

***Before:***

******

***After:***

****

**INSERT INTO Commands:**

-- INSERT DATA INTO TABLE emp\_penalites

INSERT INTO emp\_penalites (emp\_files\_id, TYPE, reason, DATE, amount) VALUES

(1, 'Fine', 'Late to work', '2022-05-15', 100),

(2, 'Deduction', 'Violation of company policy', '2021-12-30', 200),

(3, 'Warning', 'Unprofessional behavior', '2023-02-20', NULL),

(4, 'Fine', 'Unauthorized absence', '2022-08-10', 150),

(5, 'Deduction', 'Misuse of company resources', '2021-06-25', 300),

(6, 'Warning', 'Failure to meet performance targets', '2023-01-05', NULL),

(7, 'Fine', 'Violation of safety protocols', '2022-03-12', 120),

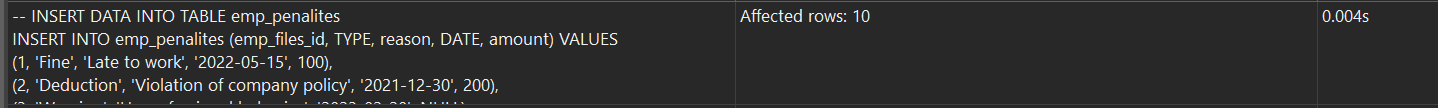
(8, 'Deduction', 'Breach of confidentiality', '2021-11-18', 250),

(9, 'Warning', 'Poor time management', '2022-01-30', NULL),

(10, 'Fine', 'Insubordination', '2023-04-05', 180);

**Excuted Result (with Query Commands):**

(Excute successfully)

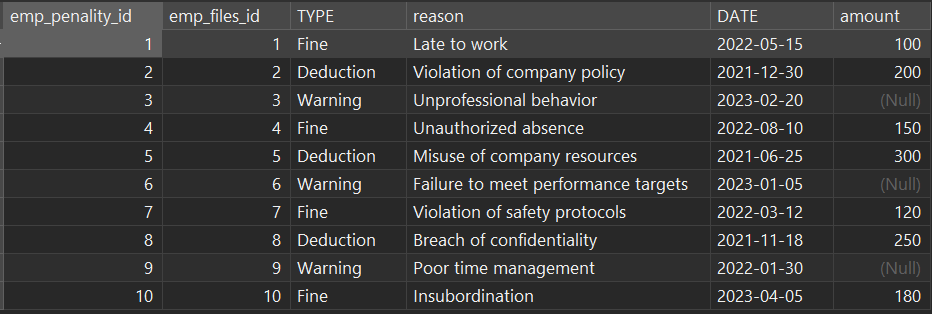


SELECT \* FROM emp\_penalites;

***Before:***

******

***After:***

****

**INSERT INTO Commands:**

-- INSERT DATA INTO TABLE edu\_history

INSERT INTO edu\_history (emp\_files\_id, school\_name, major, LEVEL, graduate\_date) VALUES

(1, 'ABC University', 'Computer Science', 'Bachelor', '2021-06-30'),

(2, 'XYZ College', 'Business Administration', 'Master', '2020-12-15'),

(3, '123 School of Engineering', 'Mechanical Engineering', 'Bachelor', '2022-05-20'),

(4, 'DEF Institute', 'Finance', 'Bachelor', '2019-09-10'),

(5, 'LMN College', 'Marketing', 'Master', '2023-03-25'),

(6, '456 University', 'Electrical Engineering', 'Bachelor', '2022-01-05'),

(7, 'GHI College', 'Human Resources', 'Master', '2021-07-12'),

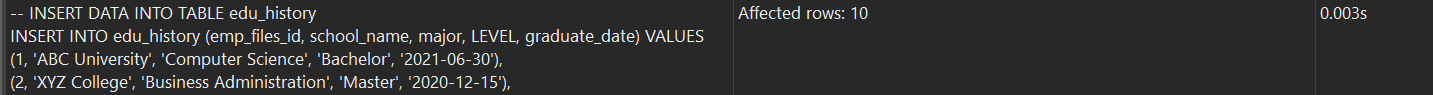
(8, 'UVW School of Medicine', 'Medicine', 'Doctorate', '2020-11-18'),

(9, 'PQR Institute of Technology', 'Information Technology', 'Bachelor', '2023-01-30'),

(10, 'JKL College', 'Psychology', 'Master', '2022-04-05');

**Excuted Result (with Query Commands):**

(Excute successfully)

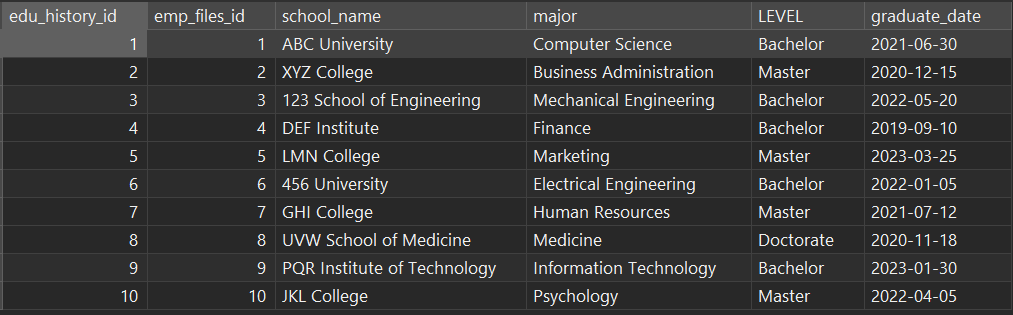


SELECT \* FROM edu\_history;

***Before:***

******

***After:***

****

**INSERT INTO Commands:**

-- INSERT DATA INTO TABLE contracts

INSERT INTO contracts (emp\_files\_id, start\_date, end\_date, types, status) VALUES

(1, '2021-06-01', '2023-05-31', 'Full-time', 'Active'),

(2, '2020-12-15', '2022-12-14', 'Part-time', 'Active'),

(3, '2022-03-10', '2024-03-09', 'Contractor', 'Active'),

(4, '2019-09-01', '2022-08-31', 'Full-time', 'Inactive'),

(5, '2023-01-15', '2023-07-14', 'Internship', 'Active'),

(6, '2022-08-20', '2023-08-19', 'Part-time', 'Active'),

(7, '2021-07-01', '2023-06-30', 'Full-time', 'Active'),

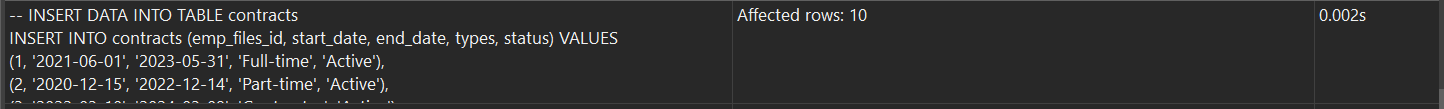
(8, '2020-11-01', '2021-10-31', 'Contractor', 'Inactive'),

(9, '2023-02-05', '2024-02-04', 'Full-time', 'Active'),

(10, '2022-04-10', '2023-04-09', 'Internship', 'Active');

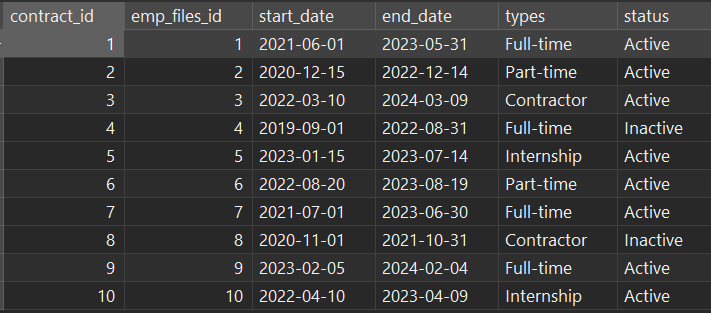
**Excuted Result (with Query Commands):**

(Excute successfully)



SELECT \* FROM contracts;

***Before: ***

***After:* **

**INSERT INTO Commands:**

-- INSERT DATA INTO TABLE resignations

INSERT INTO resignations (resign\_emp\_id, resign\_reason, RESIGN\_DATE, related\_contract\_id, exit\_interviewer\_id) VALUES

(1, 'Personal reasons', '2023-05-15', 1, 5),

(2, 'Found a new opportunity', '2023-06-30', 2, 8),

(3, 'Relocating to another city', '2023-07-20', 3, 6),

(4, 'Health issues', '2023-08-10', 4, 3),

(5, 'Career change', '2023-09-05', 5, 7),

(6, 'Retirement', '2023-10-15', 6, 2),

(7, 'Family reasons', '2023-11-25', 7, 4),

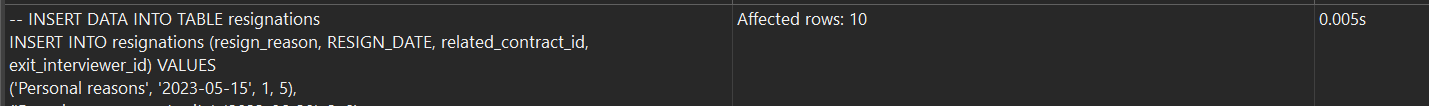
(8, 'Pursuing higher education', '2023-12-20', 8, 9),

(9, 'Contract completion', '2023-12-31', 9, 1),

(10, 'Unsatisfactory work environment', '2024-01-15', 10, 7);

**Excuted Result (with Query Commands):**

(Excute successfully)

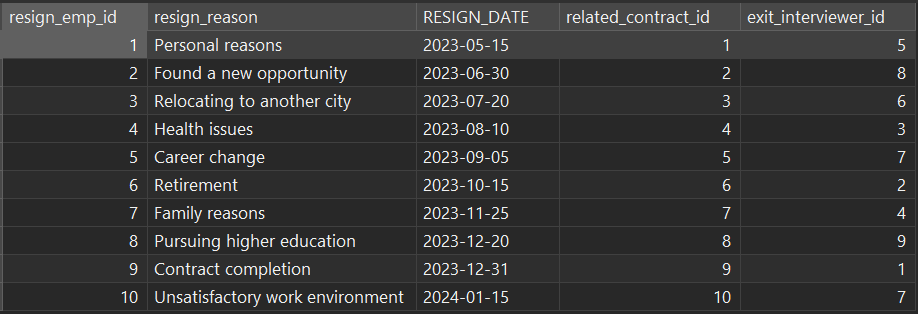


SELECT \* FROM resignations;

***Before:***

******

***After:***

****

**INSERT INTO Commands:**

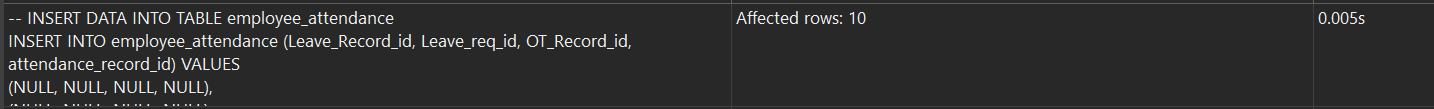
-- INSERT DATA INTO TABLE employee\_attendance

INSERT INTO employee\_attendance (employee\_attendance\_id) VALUES

(1), (2), (3), (4), (5), (6), (7), (8), (9), (10);

**Excuted Result (data will be updated in the UPDATE section later):**

(Excute successfully)

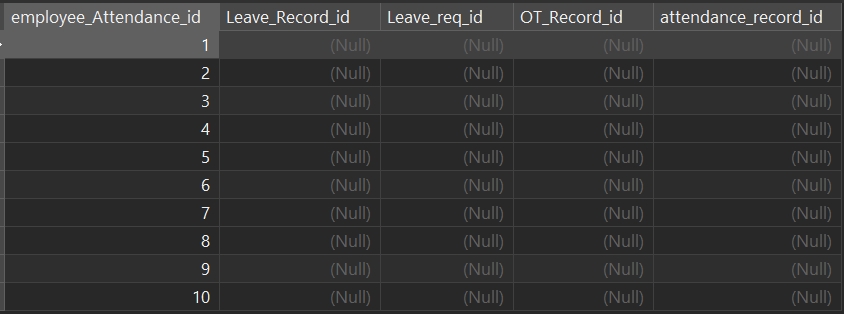


SELECT \* FROM employee\_attendance;

***Before:***

******

***After:***

****

**INSERT INTO Commands:**

-- INSERT DATA INTO TABLE ot\_record

INSERT INTO ot\_record (record\_date, time\_in, time\_out, ot\_hour) VALUES

('2023-05-01', '08:30:00', '17:30:00', 9.00),

('2023-05-02', '09:00:00', '18:00:00', 8.00),

('2023-05-03', '08:00:00', '17:00:00', 0.00),

('2023-05-04', '10:00:00', '19:00:00', 9.00),

('2023-05-05', '09:30:00', '18:30:00', 9.00),

('2023-05-06', '08:45:00', '18:45:00', 10.00),

('2023-05-07', '09:15:00', '19:15:00', 10.00),

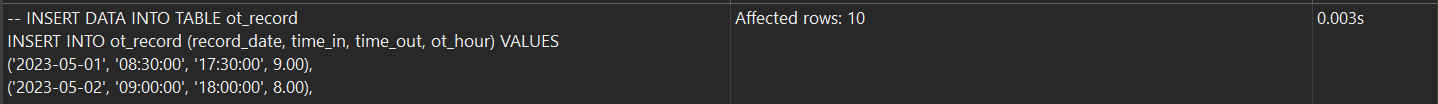
('2023-05-08', '08:30:00', '17:30:00', 9.00),

('2023-05-09', '09:00:00', '18:00:00', 8.00),

('2023-05-10', '08:00:00', '17:00:00', 0.00);

**Excuted Result (with Query Commands):**

(Excute successfully)

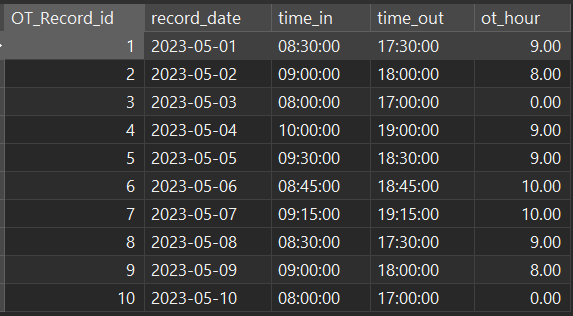


SELECT \* FROM ot\_record;

***Before:***

******

***After:***

****

**INSERT INTO Commands:**

-- INSERT DATA INTO TABLE attendance\_record

INSERT INTO attendance\_record (record\_date, time\_in, time\_out, OT\_Record\_id) VALUES

('2023-05-01', '08:30:00', '17:30:00', 1),

('2023-05-02', '09:00:00', '18:00:00', 2),

('2023-05-03', '08:00:00', '17:00:00', 3),

('2023-05-04', '10:00:00', '19:00:00', 4),

('2023-05-05', '09:30:00', '18:30:00', 5),

('2023-05-06', '08:45:00', '18:45:00', 6),

('2023-05-07', '09:15:00', '19:15:00', 7),

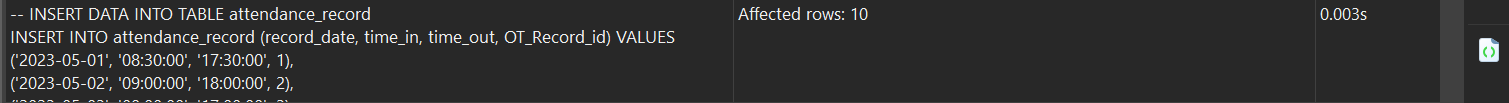
('2023-05-08', '08:30:00', '17:30:00', 8),

('2023-05-09', '09:00:00', '18:00:00', 9),

('2023-05-10', '08:00:00', '17:00:00', 10);

**Excuted Result (with Query Commands):**

(Excute successfully)

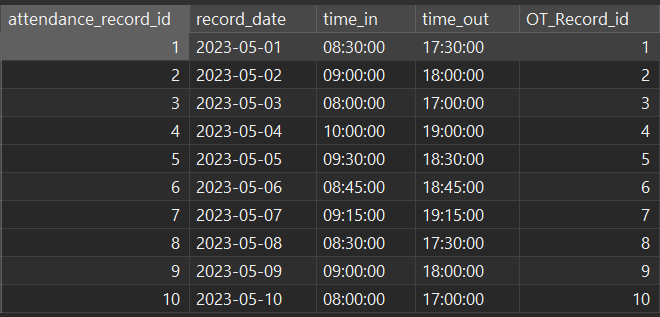


SELECT \* FROM attendance\_record;

***Before:***

******

***After:***

****

**INSERT INTO Commands:**

-- INSERT DATA INTO TABLE leave\_request

INSERT INTO leave\_request (requester\_id, req\_reason, req\_start\_date, req\_end\_date, approve\_status) VALUES

(1, 'Vacation', '2023-05-15', '2023-05-18', 'Pending'),

(2, 'Sick Leave', '2023-05-20', '2023-05-22', 'Approved'),

(3, 'Family Emergency', '2023-05-10', '2023-05-11', 'Approved'),

(4, 'Personal Leave', '2023-05-25', '2023-05-25', 'Pending'),

(5, 'Vacation', '2023-05-12', '2023-05-14', 'Approved'),

(6, 'Sick Leave', '2023-05-30', '2023-05-31', 'Pending'),

(7, 'Vacation', '2023-05-23', '2023-05-25', 'Approved'),

(8, 'Personal Leave', '2023-05-05', '2023-05-06', 'Approved'),

(9, 'Sick Leave', '2023-05-02', '2023-05-03', 'Approved'),

(10, 'Family Emergency', '2023-05-28', '2023-05-29', 'Pending');

**Excuted Result (with Query Commands):**

(Excute successfully)

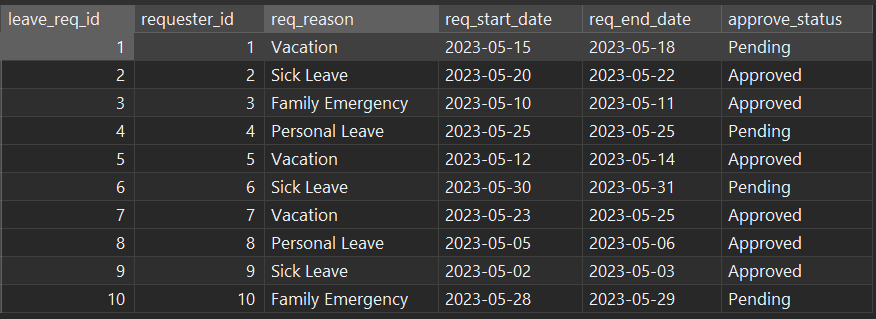


SELECT \* FROM leave\_request;

***Before:***

******

***After:***

****

**INSERT INTO Commands:**

-- INSERT DATA INTO TABLE leave\_record

INSERT INTO leave\_record (leave\_req\_id, leave\_start\_date, leave\_end\_date, leave\_approver\_id) VALUES

(1, '2023-05-15', '2023-05-18', 1),

(2, '2023-05-20', '2023-05-22', 2),

(3, '2023-05-10', '2023-05-11', 3),

(4, '2023-05-25', '2023-05-25', 4),

(5, '2023-05-12', '2023-05-14', 5),

(6, '2023-05-30', '2023-05-31', 6),

(7, '2023-05-23', '2023-05-25', 7),

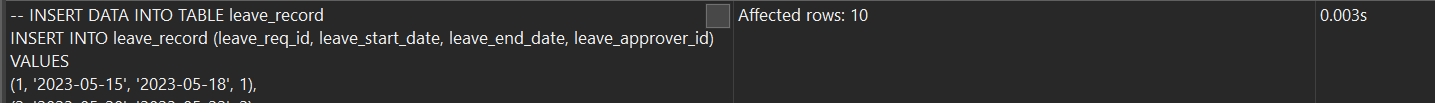
(8, '2023-05-05', '2023-05-06', 8),

(9, '2023-05-02', '2023-05-03', 9),

(10, '2023-05-28', '2023-05-29', 10);

**Excuted Result (with Query Commands):**

(Excute successfully)

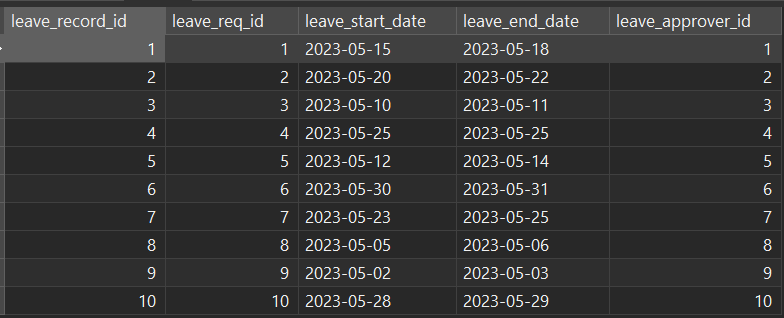


SELECT \* FROM leave\_record;

***Before:***

******

***After:***

****

**INSERT INTO Commands:**

-- INSERT DATA INTO TABLE position\_opening

INSERT INTO position\_opening (Position\_id, Dept\_id, Experience\_Required, Education\_Required) VALUES

(1, 1, '2-4 years of relevant experience', 'Bachelor\'s degree in Computer Science'),

(2, 2, '5+ years of experience in marketing', 'Bachelor\'s degree in Marketing or related field'),

(3, 3, '3-5 years of experience in finance', 'Bachelor\'s degree in Finance or Accounting'),

(4, 4, '2-3 years of experience in customer service', 'High school diploma or equivalent'),

(5, 5, '4-6 years of experience in sales', 'Bachelor\'s degree in Business or a related field'),

(6, 1, '3+ years of experience in project management', 'Bachelor\'s degree in Project Management or a related field'),

(7, 2, '5-7 years of experience in human resources', 'Bachelor\'s degree in Human Resources or a related field'),

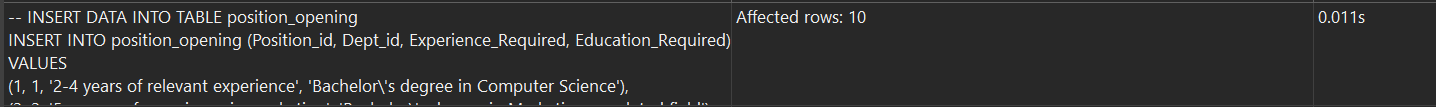
(8, 3, '2-4 years of experience in software development', 'Bachelor\'s degree in Computer Science or Software Engineering'),

(9, 4, '1-2 years of experience in data analysis', 'Bachelor\'s degree in Data Science or a related field'),

(10, 5, '6+ years of experience in leadership', 'Bachelor\'s degree in Business Administration or a related field');

**Excuted Result (with Query Commands):**

(Excute successfully)



SELECT \* FROM position\_opening;

***Before:***

******

***After:***

****

**INSERT INTO Commands:**

-- INSERT DATA INTO TABLE employee\_benefits

INSERT INTO employee\_benefits (medical\_insurance, sick\_leave, vision\_insurance, maternity\_leave) VALUES

(1, 1, 0, 1),

(0, 1, 1, 0),

(1, 1, 1, 1),

(1, 0, 0, 1),

(1, 1, 0, 0),

(0, 1, 1, 1),

(1, 0, 1, 1),

(1, 1, 1, 0),

(0, 0, 1, 1),

(1, 1, 0, 1);

**Excuted Result (1=true/have, 0=false/don’t have):**

(Excute successfully)

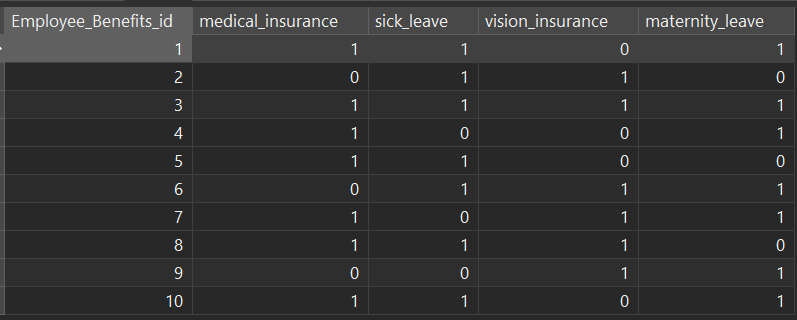


SELECT \* FROM employee\_benefits;

***Before:***

******

***After:***

****

**INSERT INTO Commands:**

-- INSERT DATA INTO TABLE employee\_awards

INSERT INTO employee\_awards (Employee\_Benefits\_id, Title, Reason, date, Amount) VALUES

(1, 'Outstanding Performance Award', 'Exceeded sales targets for Q1', '2023-04-15', 1000),

(2, 'Team Collaboration Award', 'Contributed significantly to a successful project', '2023-03-22', 800),

(3, 'Innovation Excellence Award', 'Introduced a new cost-saving process', '2023-04-05', 1200),

(4, 'Leadership Achievement Award', 'Successfully led a cross-functional team', '2023-02-10', 1500),

(5, 'Customer Service Award', 'Received outstanding customer feedback', '2023-05-02', 600),

(1, 'Employee of the Month', 'Consistently high performance and dedication', '2023-04-30', 500),

(3, 'Quality Improvement Award', 'Improved product quality and customer satisfaction', '2023-03-18', 1000),

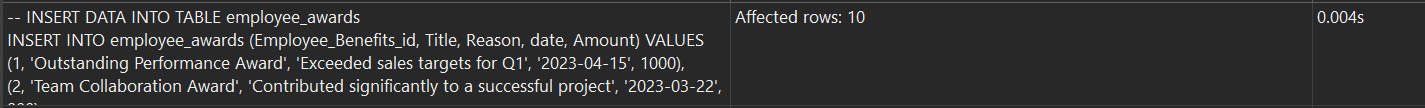
(2, 'Excellence in Problem-Solving', 'Resolved a critical issue with efficiency', '2023-05-08', 700),

(4, 'Inspirational Leadership Award', 'Motivated team members to achieve their best', '2023-02-28', 900),

(5, 'Safety Champion Award', 'Promoted a culture of safety in the workplace', '2023-04-20', 800);

**Excuted Result (with Query Commands):**

(Excute successfully)

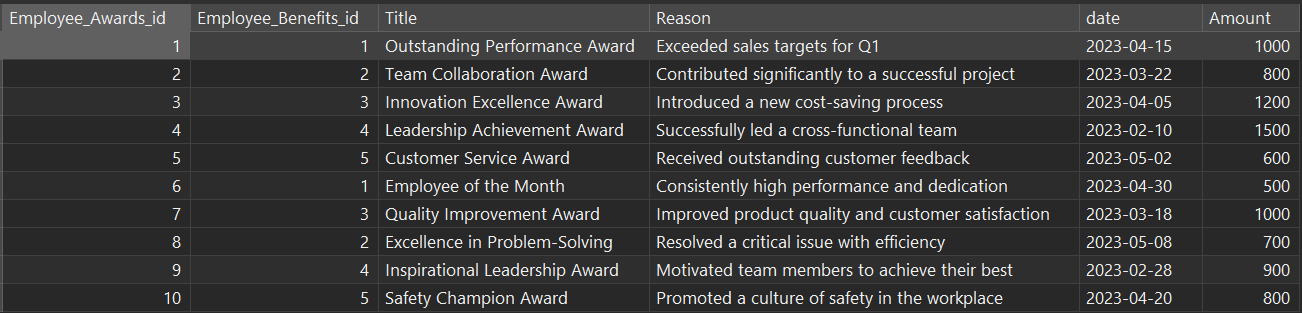


SELECT \* FROM employee\_awards;

***Before:***

******

***After:***

****

**INSERT INTO Commands:**

-- INSERT DATA INTO TABLE employees\_salary\_details

INSERT INTO employees\_salary\_details (Employee\_Benefits\_id, BASE\_SALARY, BONUS, ALLOWANCE, DEDUCTIONS, BENEFITS) VALUES

(1, 5000.00, 1000.00, 500.00, 200.00, 300.00),

(2, 4800.00, 800.00, 450.00, 180.00, 280.00),

(3, 5200.00, 1200.00, 600.00, 250.00, 320.00),

(4, 5500.00, 1500.00, 700.00, 280.00, 350.00),

(5, 4700.00, 750.00, 400.00, 150.00, 240.00),

(1, 4900.00, 900.00, 480.00, 190.00, 290.00),

(3, 5100.00, 1100.00, 580.00, 230.00, 310.00),

(2, 4600.00, 700.00, 420.00, 160.00, 260.00),

(4, 5300.00, 1400.00, 650.00, 270.00, 330.00),

(5, 4800.00, 800.00, 450.00, 180.00, 280.00);

**Excuted Result (with Query Commands):**

(Excute successfully)

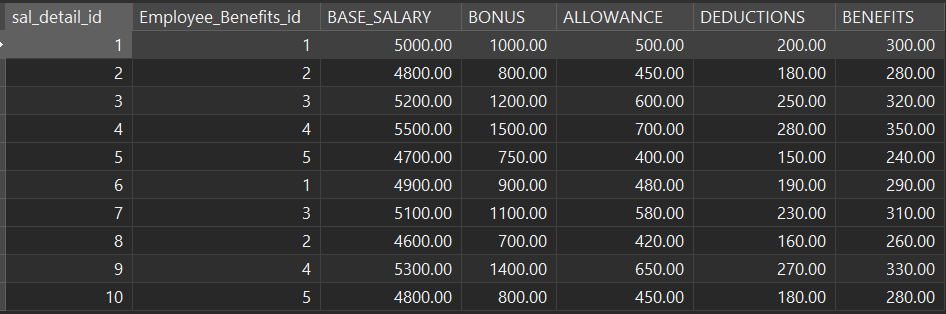


SELECT \* FROM employees\_salary\_details;

***Before:***

******

***After:***

****

**INSERT INTO Commands:**

-- INSERT DATA INTO TABLE salary\_history

INSERT INTO salary\_history (Employee\_Benefits\_id, gender, TOTAL\_SALARY, START\_DATE, END\_DATE) VALUES

(1, 'Male', 5500.00, '2023-01-01', '2023-01-31'),

(2, 'Female', 5200.00, '2023-01-01', '2023-01-31'),

(3, 'Male', 5800.00, '2023-01-01', '2023-01-31'),

(4, 'Female', 6100.00, '2023-01-01', '2023-01-31'),

(5, 'Male', 5300.00, '2023-01-01', '2023-01-31'),

(1, 'Female', 5400.00, '2023-02-01', '2023-02-28'),

(3, 'Male', 6000.00, '2023-02-01', '2023-02-28'),

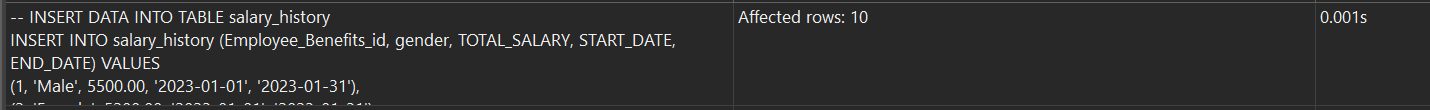
(2, 'Male', 5000.00, '2023-02-01', '2023-02-28'),

(4, 'Female', 6200.00, '2023-02-01', '2023-02-28'),

(5, 'Male', 5500.00, '2023-02-01', '2023-02-28');

**Excuted Result (with Query Commands):**

(Excute successfully)

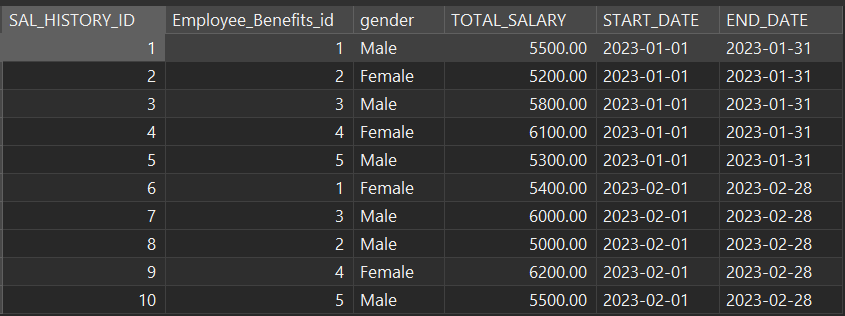


SELECT \* FROM salary\_history;

***Before:***

******

***After:***

****

**INSERT INTO Commands:**

-- INSERT DATA INTO TABLE skills

INSERT INTO skills (Skill\_name, dept\_id, skill\_resources, skill\_description) VALUES

-- skills that matched Department of Sale

('Sales Techniques', 1, 'Sales training materials', 'Techniques to improve sales performance'),

('Negotiation Skills', 1, 'Negotiation workshops', 'Skills for effective negotiation'),

('Customer Relationship Management', 1, 'CRM software', 'Managing customer relationships effectively'),

('Presentation Skills', 1, 'Presentation workshops', 'Improving presentation and communication skills'),

-- skills that matched Department of Information Technology

('Java', 2, 'Online Java tutorials', 'Java programming language'),

('Python', 2, 'Python IDEs', 'Python programming language'),

('Web Development', 2, 'Web development resources', 'Skills for building web applications'),

('Database Management', 2, 'Database management tools', 'Managing and optimizing databases'),

-- skills that matched Department of Human Resources

('Recruitment and Selection', 3, 'Recruitment guidelines', 'Best practices for hiring and selection'),

('Performance Management', 3, 'Performance appraisal tools', 'Managing employee performance'),

('Training and Development', 3, 'Training modules', 'Developing employee skills through training'),

('Employee Relations', 3, 'Employee relation resources', 'Managing employee relations and conflicts'),

-- skills that matched Department of Marketing

('Digital Marketing', 4, 'Online marketing platforms', 'Skills for digital marketing campaigns'),

('Market Research', 4, 'Market research tools', 'Conducting market research and analysis'),

('Brand Management', 4, 'Brand management guides', 'Managing and building brand identity'),

('Social Media Marketing', 4, 'Social media marketing resources', 'Utilizing social media for marketing'),

-- skills that matched Department of Finance

('Financial Analysis', 5, 'Financial analysis software', 'Analyzing financial data and reports'),

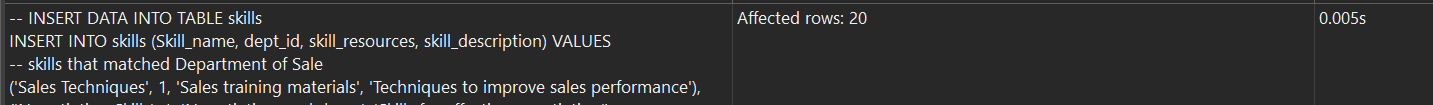
('Budgeting and Forecasting', 5, 'Budgeting tools', 'Creating budgets and financial forecasts'),

('Taxation', 5, 'Taxation guidelines', 'Understanding and complying with taxation laws'),

('Risk Management', 5, 'Risk management strategies', 'Identifying and managing financial risks');

**Excuted Result (with Query Commands):**

(Excute successfully)

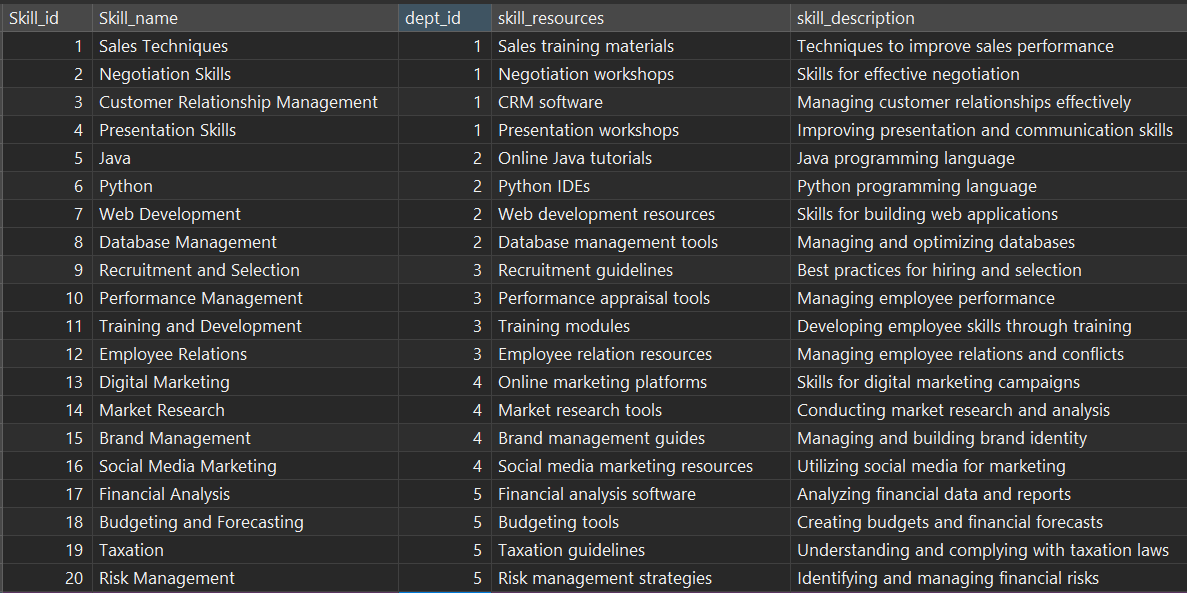


SELECT \* FROM skills;

***Before:***

******

***After:***

****

**INSERT INTO Commands:**

-- INSERT DATA INTO TABLE candidates

INSERT INTO `candidates` (`Candidate\_name`, `Email`, `phone\_no`, `Skill\_set\_id`, `position\_Opening\_id`) VALUES

('Emily Turner', 'emily.turner@example.com', '020-11111111', 12, 6),

('Jacob Parker', 'jacob.parker@example.com', '021-22222222', 13, 8),

('Olivia Mitchell', 'olivia.mitchell@example.com', '022-33333333', 14, 2),

('Michael Scott', 'michael.scott@example.com', '023-44444444', 15, 4),

('Sophia Young', 'sophia.young@example.com', '024-55555555', 16, 9),

('William Allen', 'william.allen@example.com', '025-66666666', 19, 5),

('Emma Hill', 'emma.hill@example.com', '026-77777777', 20, 10),

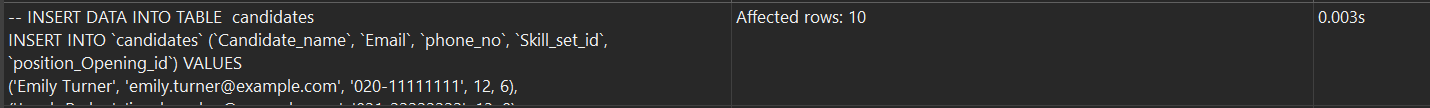
('Noah Turner', 'noah.turner@example.com', '027-88888888', 11, 3),

('Ava Reed', 'ava.reed@example.com', '028-99999999', 6, 7),

('James Evans', 'james.evans@example.com', '029-00000000', 8, 2);

**Excuted Result (with Query Commands):**

(Excute successfully)

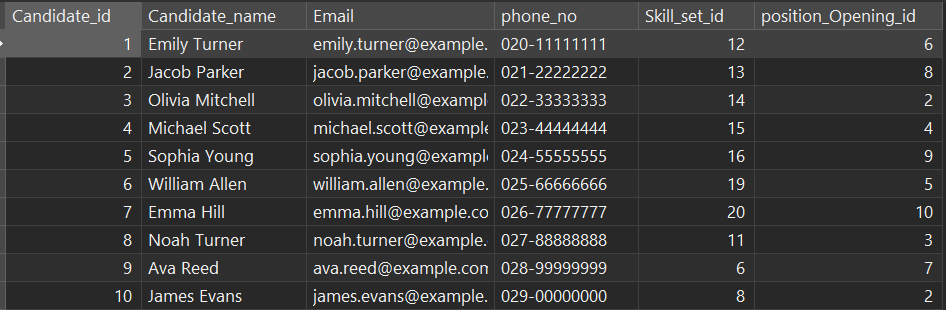


SELECT \* FROM candidates;

***Before:***

******

***After:***

****

**INSERT INTO Commands:**

-- INSERT DATA INTO TABLE skillset

INSERT INTO skillset (emp\_pnd\_id, candidate\_id, Skill\_one\_id, Skill\_two\_id, Skill\_three\_id) VALUES

(NULL, NULL, 2, 3, 6),

(NULL, NULL, 8, 9, 11),

(NULL, NULL, 14, 15, 17),

(NULL, NULL, 19, 20, 4),

(NULL, NULL, 10, 12, 18),

(NULL, 1, 3, 5, 7),

(NULL, 2, 9, 13, 16),

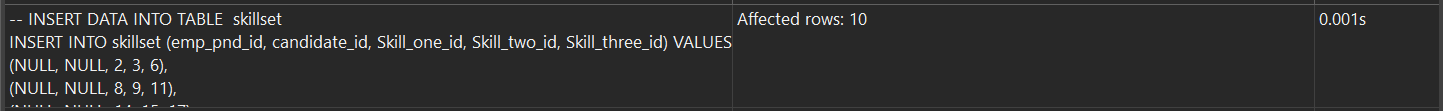
(NULL, 3, 17, 6, 12),

(NULL, 4, 8, 10, 15),

(NULL, 5, 2, 4, 20);

**Excuted Result (data will be updated at UPDATE section later):**

(Excute successfully)

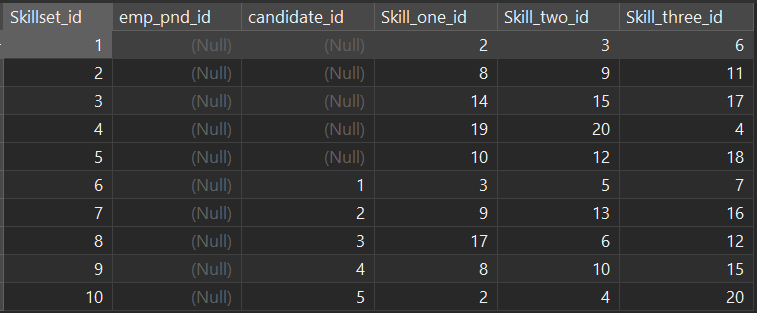


SELECT \* FROM skillset;

***Before:***

******

***After:***

****

**INSERT INTO Commands:**

-- INSERT DATA INTO TABLE position\_application

INSERT INTO `position\_application` (`emp\_id`, `current\_position\_id`, `position\_opening\_id`, `application\_reason`, `skillset\_id`) VALUES

(1, 3, 8, 'I am interested in this new position', 7),

(2, 2, 5, 'I believe I have the required skills', 5),

(3, 4, 9, 'I want to take on new challenges', 10),

(4, 1, 4, 'I have relevant experience in this area', 2),

(5, 3, 7, 'I see this as an opportunity for growth', 8),

(6, 2, 5, 'I am passionate about this role', 6),

(7, 4, 10, 'I am excited about the company culture', 3),

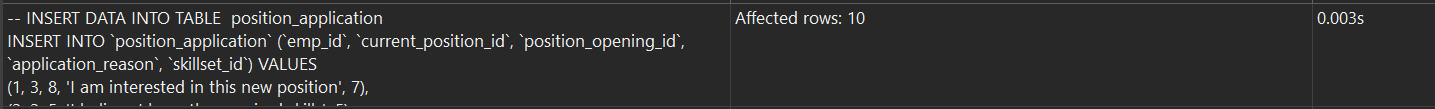
(8, 1, 3, 'I am looking for a new opportunity', 1),

(9, 3, 6, 'I want to contribute to the team', 9),

(10, 2, 2, 'I am ready to take on more responsibility', 4);

**Excuted Result (with Query Commands):**

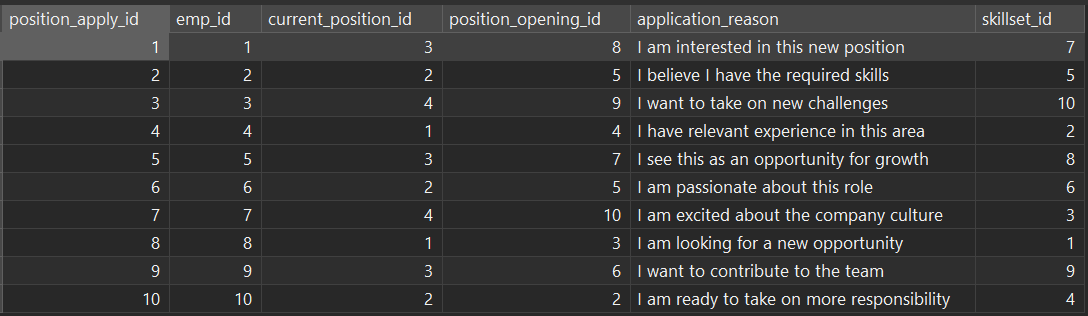
(Excute successfully)



SELECT \* FROM position\_application;

***Before:  
***

***After:***

****

**INSERT INTO Commands:**

-- INSERT DATA INTO TABLE interview\_list

INSERT INTO interview\_list (Candidate\_id, Interviewer\_id, Interview\_date, Interview\_result) VALUES

(1, 5, '2023-05-01', 'Impressed with candidate\'s technical skills and problem-solving ability.'),

(2, 3, '2023-05-02', 'Candidate demonstrated excellent communication and teamwork.'),

(3, 1, '2023-05-03', 'Candidate has relevant experience and a strong work ethic.'),

(4, 4, '2023-05-04', 'Interviewer recommended the candidate for further consideration.'),

(5, 2, '2023-05-05', 'Candidate showed a proactive attitude and willingness to learn.'),

(6, 6, '2023-05-06', 'Interviewer was impressed with the candidate\'s leadership potential.'),

(7, 7, '2023-05-07', 'Candidate displayed a deep understanding of the industry.'),

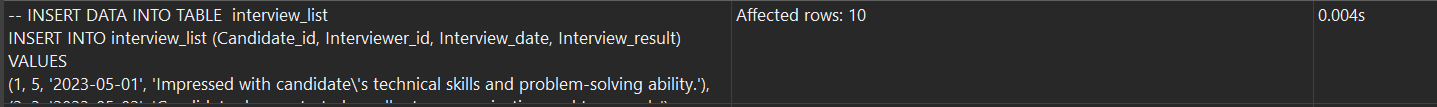
(8, 9, '2023-05-08', 'Interviewer noted the candidate\'s attention to detail and accuracy.'),

(9, 12, '2023-05-09', 'Candidate demonstrated strong problem-solving skills.'),

(10, 8, '2023-05-10', 'Interviewer was pleased with the candidate\'s passion for the role.');

**Excuted Result (with Query Commands):**

(Excute successfully)



SELECT \* FROM interview\_list;

***Before:***

******

***After:***

****

**INSERT INTO Commands:**

-- INSERT DATA INTO TABLE employee\_training

INSERT INTO `employee\_training` (`training\_skill\_id`, `trainer\_id`, `training\_date`, `training\_duration`) VALUES

(1, 4, '2023-05-01', '3 days'),

(2, 4, '2023-05-02', '2 days'),

(3, 4, '2023-05-03', '1 day'),

(4, 4, '2023-05-04', '4 days'),

(5, 8, '2023-05-05', '5 days'),

(1, 4, '2023-05-06', '2 days'),

(2, 8, '2023-05-07', '3 days'),

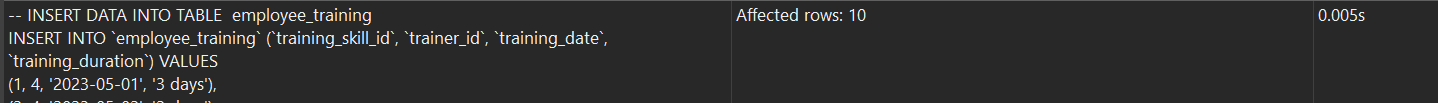
(3, 8, '2023-05-08', '4 days'),

(4, 8, '2023-05-09', '1 day'),

(5, 8, '2023-05-10', '2 days');

**Excuted Result (with Query Commands):**

(Excute successfully)

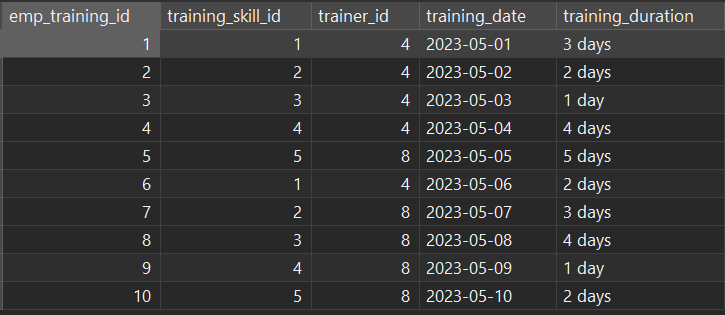


SELECT \* FROM employee\_training;

***Before:***

******

***After:***

****

**INSERT INTO Commands:**

-- INSERT DATA INTO TABLE employee\_performance\_and\_development

INSERT INTO `employee\_performance\_and\_development` (`skillset\_id`, `employee\_promotion\_id`, `emp\_training\_id`, `credit\_score`, `credit\_score\_grader\_id`) VALUES

(1, NULL, 1, 85, NULL),

(2, NULL, 2, 78, NULL),

(3, NULL, 3, 92, NULL),

(4, NULL, 4, 79, NULL),

(5, NULL, 5, 88, NULL),

(1, NULL, 6, 95, NULL),

(2, NULL, 7, 82, NULL),

(3, NULL, 8, 90, NULL),

(4, NULL, 9, 75, NULL),

(5, NULL, 10, 87, NULL);

**Excuted Result (data will updated at UPDATE section later):**

(Excute successfully)

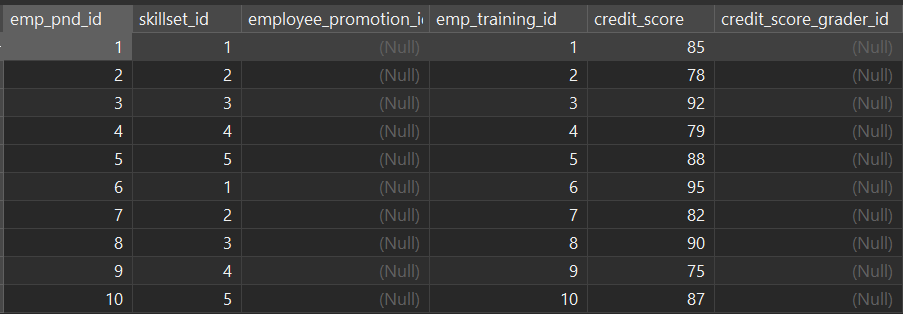


SELECT \* FROM employee\_performance\_and\_development;

***Before:***

******

***After:***

****

**INSERT INTO Commands:**

-- INSERT DATA INTO TABLE employee\_promotion

INSERT INTO employee\_promotion (promo\_reason, current\_position\_id, promoted\_position\_id, promoted\_date) VALUES

('Excellent performance and leadership skills', 3, 8, '2023-03-15'),

('Completed advanced training program', 5, 10, '2023-02-28'),

('Consistent high-quality work', 2, 6, '2023-04-10'),

('Outstanding teamwork and problem-solving', 4, 9, '2023-03-05'),

('Demonstrated exceptional customer service', 6, 12, '2023-04-20'),

('Achieved remarkable sales performance', 7, 13, '2023-03-22'),

('Strong dedication and commitment', 1, 4, '2023-02-12'),

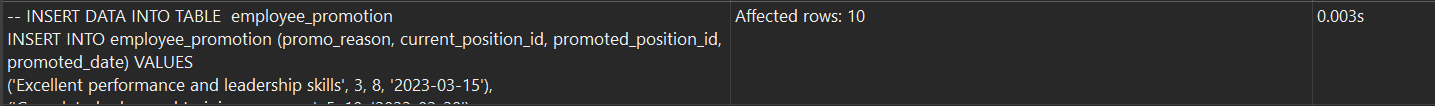
('Leadership skills and team management', 9, 15, '2023-03-18'),

('Innovative ideas and project management', 8, 14, '2023-02-25'),

('Exemplary communication and problem-solving', 10, 16, '2023-04-05');

**Excuted Result (with Query Commands):**

(Excute successfully)

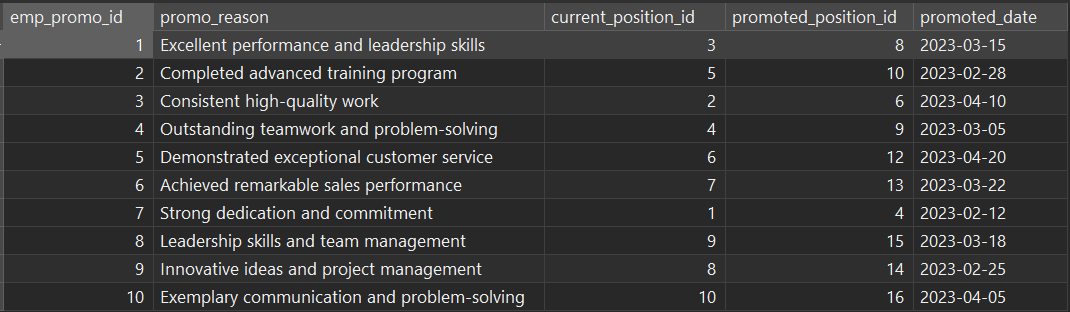


SELECT \* FROM employee\_promotion;

***Before:***

******

***After:***

****

**INSERT INTO Commands:**

-- INSERT DATA INTO TABLE projects

INSERT INTO projects (project\_title, leader\_id, start\_date, end\_date, comment) VALUES

('Sales Performance Analysis', 2, '2023-06-10', '2023-08-20', 'Analyzing sales data for improvement'),

('IT System Upgrade', 3, '2023-07-15', '2023-09-25', 'Enhancing technology infrastructure'),

('Employee Training Initiative', 1, '2023-05-05', '2023-07-15', 'Enhancing employee skills'),

('Marketing Campaign Launch', 4, '2023-06-25', '2023-09-05', 'Launching new marketing campaign'),

('Financial Data Management', 5, '2023-07-01', '2023-09-30', 'Optimizing financial data handling'),

('Sales Team Performance Evaluation', 2, '2023-08-05', '2023-10-15', 'Assessing sales team effectiveness'),

('IT Security Enhancement', 3, '2023-06-15', '2023-08-30', 'Improving IT security measures'),

('Talent Acquisition Strategy', 1, '2023-07-20', '2023-10-10', 'Developing talent acquisition plan'),

('Product Marketing Research', 4, '2023-05-20', '2023-07-30', 'Conducting market research for products'),

('Financial Forecasting Project', 5, '2023-08-15', '2023-11-05', 'Forecasting financial trends');

**Excuted Result (with Query Commands):**

(Excute successfully)

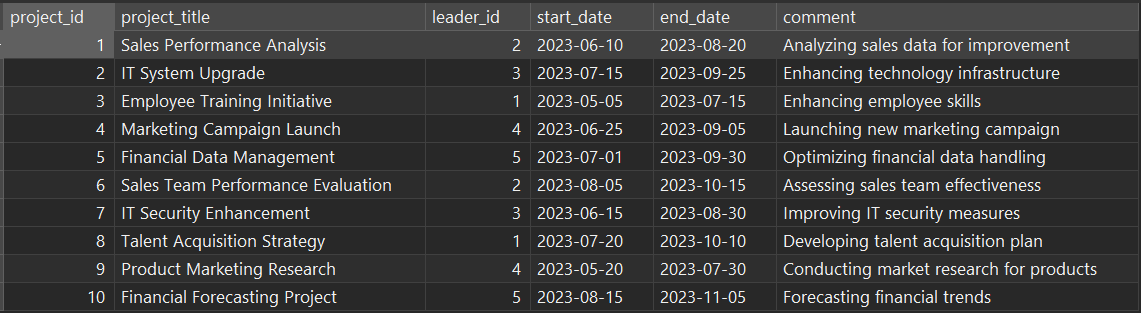


SELECT \* FROM projects;

***Before:***

******

***After:***

****

**INSERT INTO Commands:**

-- INSERT DATA INTO TABLE project\_participation

INSERT INTO project\_participation (emp\_pnd\_id, project\_id, comment, start\_date, end\_date) VALUES

(1, 1, 'Assisting in sales analysis', '2023-06-10', '2023-08-20'),

(2, 3, 'Working on IT system upgrade', '2023-07-15', '2023-09-25'),

(3, 2, 'Participating in employee training', '2023-05-05', '2023-07-15'),

(4, 4, 'Contributing to marketing campaign', '2023-06-25', '2023-09-05'),

(5, 5, 'Supporting financial data management', '2023-07-01', '2023-09-30'),

(1, 2, 'Evaluating sales team performance', '2023-08-05', '2023-10-15'),

(2, 3, 'Assisting in IT security enhancement', '2023-06-15', '2023-08-30'),

(3, 1, 'Participating in talent acquisition strategy', '2023-07-20', '2023-10-10'),

(4, 4, 'Conducting product marketing research', '2023-05-20', '2023-07-30'),

(5, 5, 'Contributing to financial forecasting', '2023-08-15', '2023-11-05');

**Excuted Result (with Query Commands):**

(Excute successfully)

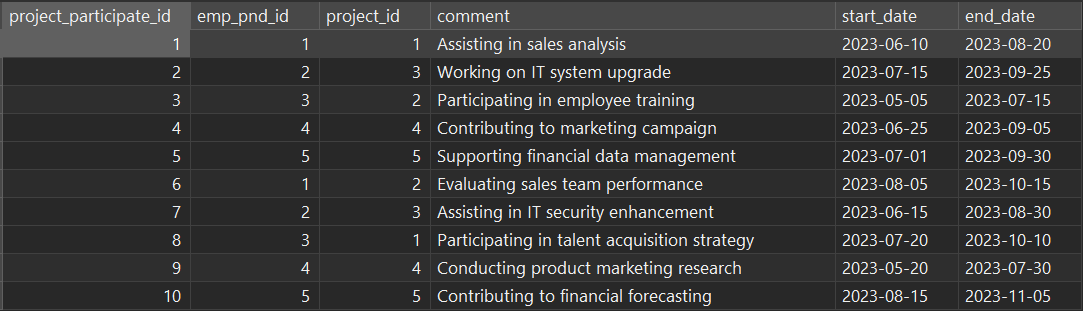


SELECT \* FROM project\_participation;

***Before:***

******

***After:***

****

**INSERT INTO Commands:**

-- INSERT DATA INTO TABLE users

INSERT INTO users (emp\_id, username, password, permission\_type, description, creat\_date, last\_update\_date) VALUES

(1, 'john\_doe\_9257', '19900515', 'general', 'Users with this permission can only view and modify their basic personal information.', '2023-05-01', '2023-05-01'),

(2, 'jane\_smith\_9257', '19850822', 'general', 'Users with this permission can only view and modify their basic personal information.', '2023-05-02', '2023-05-02'),

(3, 'michael\_johnson\_9257', '19881130', 'general', 'Users with this permission can only view and modify their basic personal information.', '2023-05-03', '2023-05-03'),

(4, 'emily\_brown\_9257', '19920325', 'general', 'Users with this permission can only view and modify their basic personal information.', '2023-05-04', '2023-05-04'),

(5, 'david\_lee\_9257', '19870918', 'general', 'Users with this permission can only view and modify their basic personal information.', '2023-05-05', '2023-05-05'),

(6, 'sarah\_wang\_9257', '19910710', 'general', 'Users with this permission can only view and modify their basic personal information.', '2023-05-06', '2023-05-06'),

(7, 'robert\_chen\_9257', '19861205', 'general', 'Users with this permission can only view and modify their basic personal information.', '2023-05-07', '2023-05-07'),

(8, 'jessica\_liu\_9257', '19890412', 'administrator', 'Users with this permission can add, delete, modify and query all tables and data.', '2023-05-08', '2023-05-08'),

(9, 'william\_zhang\_9257', '19930208', 'general', 'Users with this permission can only view and modify their basic personal information.', '2023-05-09', '2023-05-09'),

(10, 'linda\_wu\_9257', '19940101', 'general', 'Users with this permission can only view and modify their basic personal information.', '2023-05-10', '2023-05-10'),

(11, 'alex\_johnson\_9257', '19920811', 'general', 'Users with this permission can only view and modify their basic personal information.', '2023-05-11', '2023-05-11'),

(12, 'ella\_williams\_9257', '19900522', 'general', 'Users with this permission can only view and modify their basic personal information.', '2023-05-12', '2023-05-12'),

(13, 'noah\_smith\_9257', '19881119', 'general', 'Users with this permission can only view and modify their basic personal information.', '2023-05-13', '2023-05-13'),

(14, 'ava\_davis\_9257', '19950325', 'general', 'Users with this permission can only view and modify their basic personal information.', '2023-05-14', '2023-05-14'),

(15, 'liam\_taylor\_9257', '19940918', 'general', 'Users with this permission can only view and modify their basic personal information.', '2023-05-15', '2023-05-15'),

(16, 'mia\_brown\_9257', '19910711', 'general', 'Users with this permission can only view and modify their basic personal information.', '2023-05-16', '2023-05-16'),

(17, 'jameson\_lee\_9257', '19891229', 'general', 'Users with this permission can only view and modify their basic personal information.', '2023-05-17', '2023-05-17'),

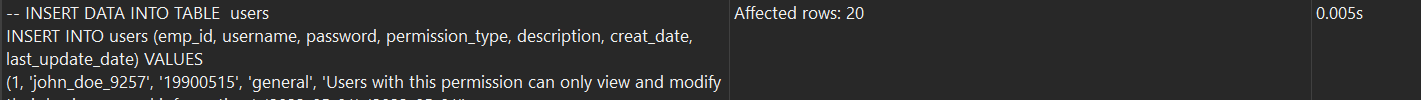
(18, 'aria\_wilson\_9257', '19930406', 'general', 'Users with this permission can only view and modify their basic personal information.', '2023-05-18', '2023-05-18'),

(19, 'ethan\_miller\_9257', '19960630', 'general', 'Users with this permission can only view and modify their basic personal information.', '2023-05-19', '2023-05-19'),

(20, 'scarlett\_martin\_9257', '19870214', 'administrator', 'Users with this permission can add, delete, modify and query all tables and data.', '2023-05-20', '2023-05-20');

**Excuted Result (with Query Commands):**

(Excute successfully)

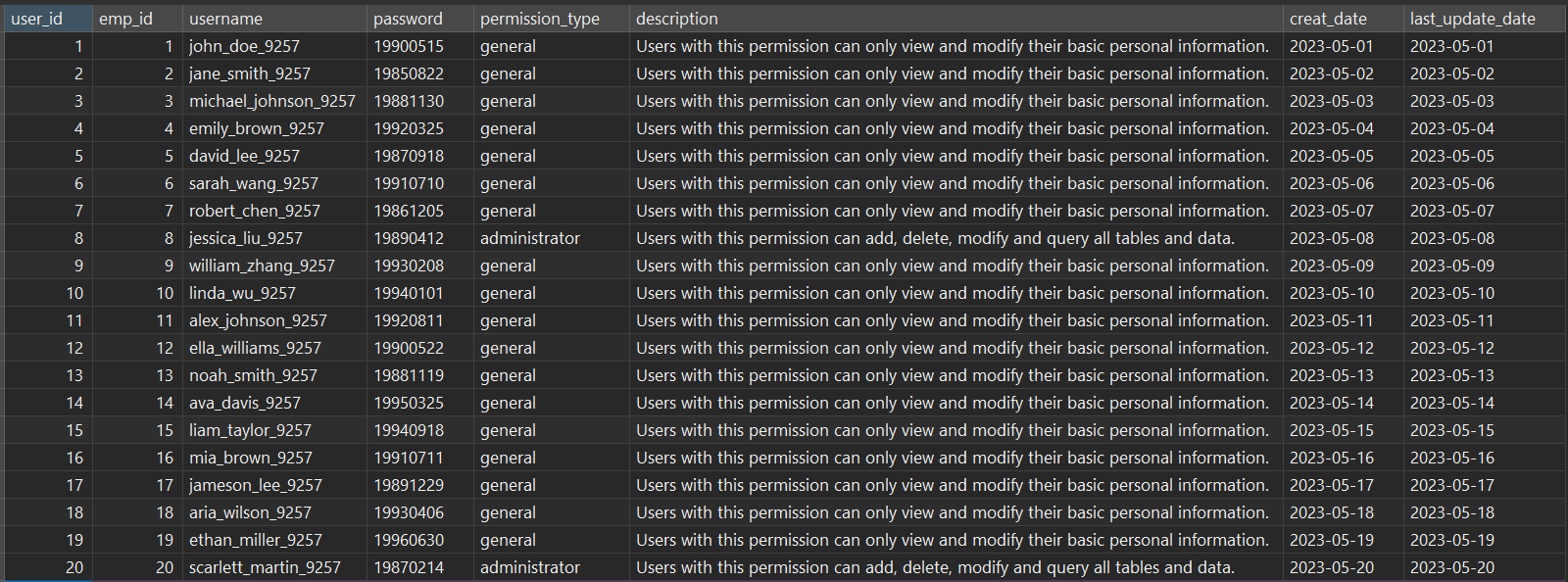


SELECT \* FROM users;

***Before:***

******

***After:***

****

**2.3.2 UPDATE**

- In MySQL, UPDATE is a command used to modify existing records in the database. Through the UPDATE statement, we can update the value of a specific column or multiple columns in the table, and modify the data according to the specified conditions.

- The following statement will be used to update the previously inserted data to ensure its accuracy and completeness, so as to ensure that the data in the human resource management system is always up-to-date and valid.

**UPDATE TABLE positions**

**UPDATE Commands:**

-- positions

UPDATE positions SET dept\_id = 1 WHERE position\_id BETWEEN 1 AND 4;

UPDATE positions SET dept\_id = 2 WHERE position\_id BETWEEN 5 AND 8;

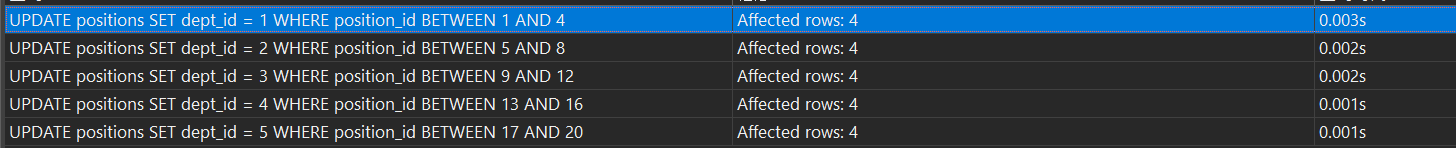
UPDATE positions SET dept\_id = 3 WHERE position\_id BETWEEN 9 AND 12;

UPDATE positions SET dept\_id = 4 WHERE position\_id BETWEEN 13 AND 16;

UPDATE positions SET dept\_id = 5 WHERE position\_id BETWEEN 17 AND 20;

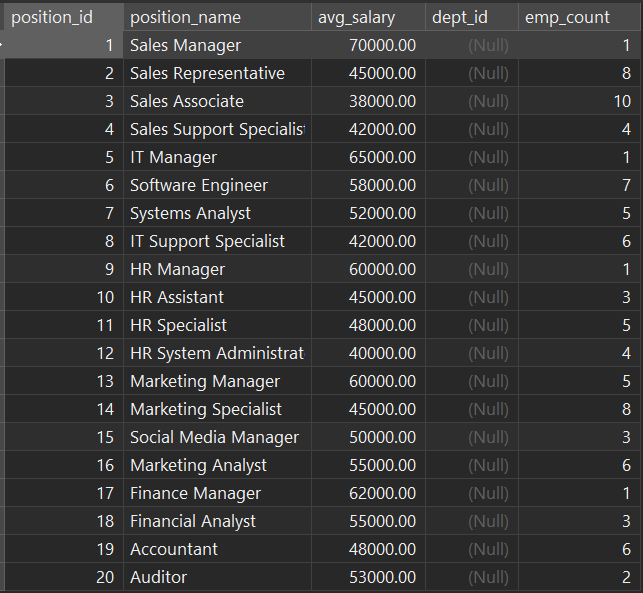
**Excuted Result (with Query Commands):**

(Excute successfully)

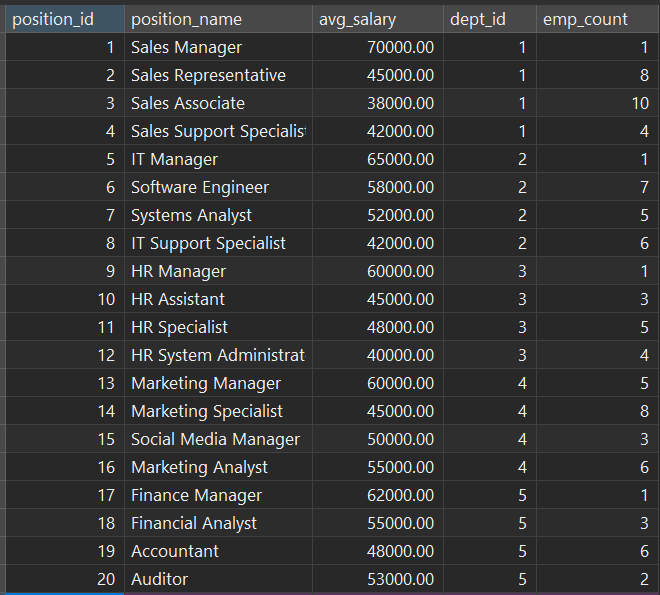
****

SELECT \* FROM positions;

***Before:***

****

***After:***

****

**UPDATE TABLE departments**

**UPDATE Commands:**

-- departments

UPDATE departments SET dept\_manager = 2 WHERE dept\_id = 1;

UPDATE departments SET dept\_manager = 3 WHERE dept\_id = 2;

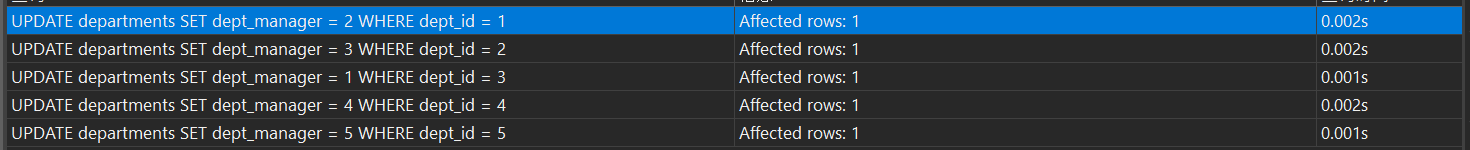
UPDATE departments SET dept\_manager = 1 WHERE dept\_id = 3;

UPDATE departments SET dept\_manager = 4 WHERE dept\_id = 4;

UPDATE departments SET dept\_manager = 5 WHERE dept\_id = 5;

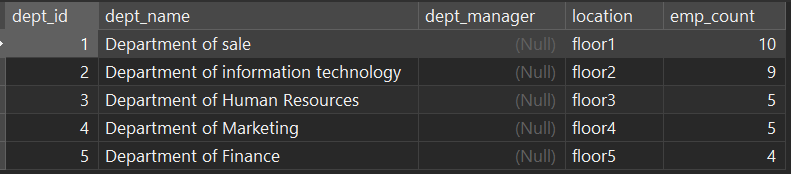
**Excuted Result (with Query Commands):**

(Excute successfully)

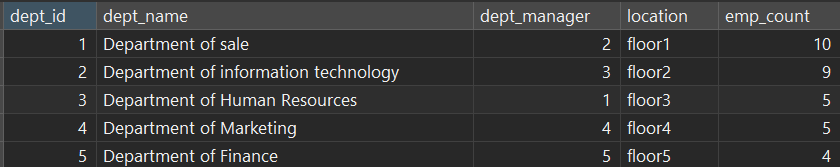


SELECT \* FROM departments;

***Before:***

****

***After:***



**UPDATE TABLE employee\_attendance**

**UPDATE Commands:**

-- employee\_attendance

UPDATE `employee\_attendance` SET

`Leave\_Record\_id` = 10,

`Leave\_req\_id` = 8,

`OT\_Record\_id` = 6,

`attendance\_record\_id` = 4

WHERE `employee\_Attendance\_id` = 1;

UPDATE `employee\_attendance` SET

`Leave\_Record\_id` = 9,

`Leave\_req\_id` = 7,

`OT\_Record\_id` = 5,

`attendance\_record\_id` = 3

WHERE `employee\_Attendance\_id` = 2;

UPDATE `employee\_attendance` SET

`Leave\_Record\_id` = 8,

`Leave\_req\_id` = 6,

`OT\_Record\_id` = 4,

`attendance\_record\_id` = 2

WHERE `employee\_Attendance\_id` = 3;

UPDATE `employee\_attendance` SET

`Leave\_Record\_id` = 7,

`Leave\_req\_id` = 5,

`OT\_Record\_id` = 3,

`attendance\_record\_id` = 1

WHERE `employee\_Attendance\_id` = 4;

UPDATE `employee\_attendance` SET

`Leave\_Record\_id` = 6,

`Leave\_req\_id` = 4,

`OT\_Record\_id` = 2,

`attendance\_record\_id` = 10

WHERE `employee\_Attendance\_id` = 5;

UPDATE `employee\_attendance` SET

`Leave\_Record\_id` = 5,

`Leave\_req\_id` = 3,

`OT\_Record\_id` = 1,

`attendance\_record\_id` = 9

WHERE `employee\_Attendance\_id` = 6;

UPDATE `employee\_attendance` SET

`Leave\_Record\_id` = 4,

`Leave\_req\_id` = 2,

`OT\_Record\_id` = 10,

`attendance\_record\_id` = 8

WHERE `employee\_Attendance\_id` = 7;

UPDATE `employee\_attendance` SET

`Leave\_Record\_id` = 3,

`Leave\_req\_id` = 1,

`OT\_Record\_id` = 9,

`attendance\_record\_id` = 7

WHERE `employee\_Attendance\_id` = 8;

UPDATE `employee\_attendance` SET

`Leave\_Record\_id` = 2,

`Leave\_req\_id` = 10,

`OT\_Record\_id` = 8,

`attendance\_record\_id` = 6

WHERE `employee\_Attendance\_id` = 9;

UPDATE `employee\_attendance` SET

`Leave\_Record\_id` = 1,

`Leave\_req\_id` = 9,

`OT\_Record\_id` = 7,

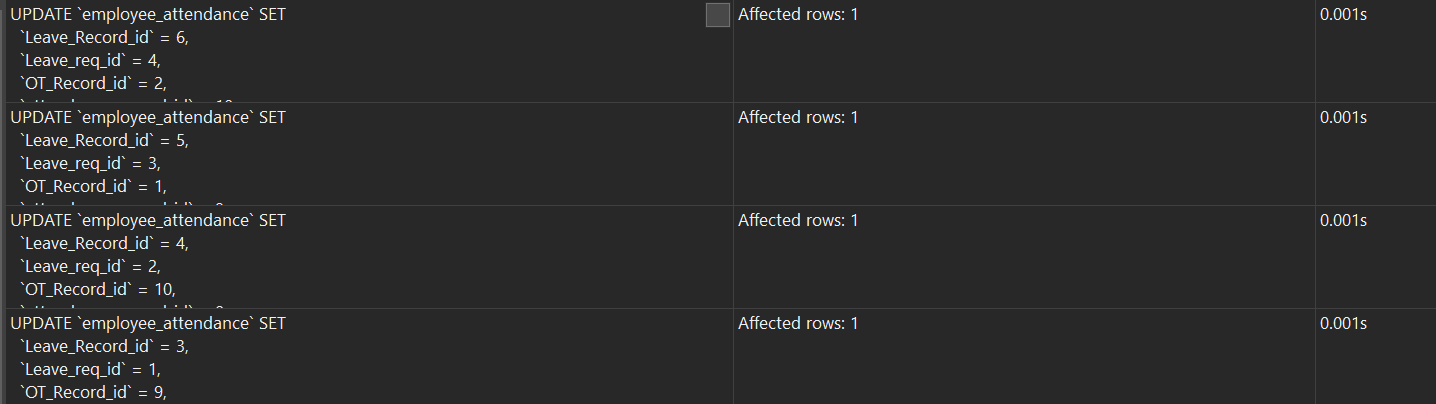
`attendance\_record\_id` = 5

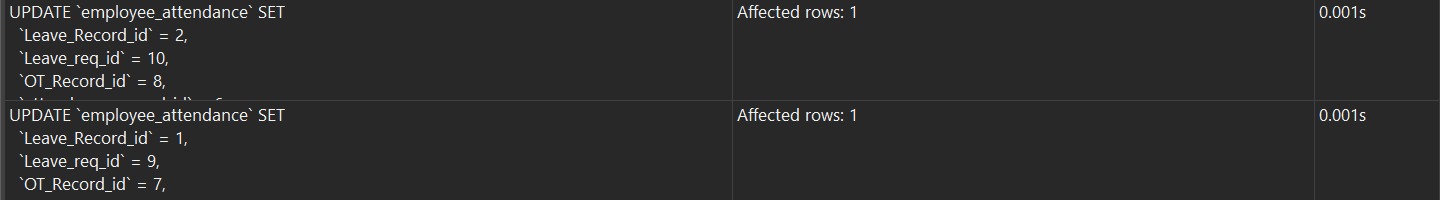
WHERE `employee\_Attendance\_id` = 10;

**Excuted Result (with Query Commands):**

(Excute successfully)

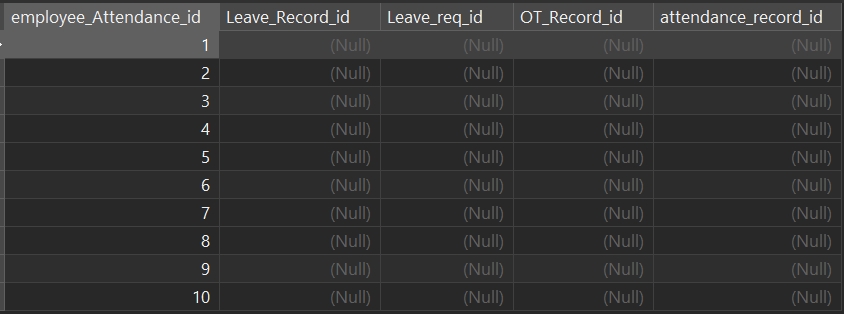
****

****

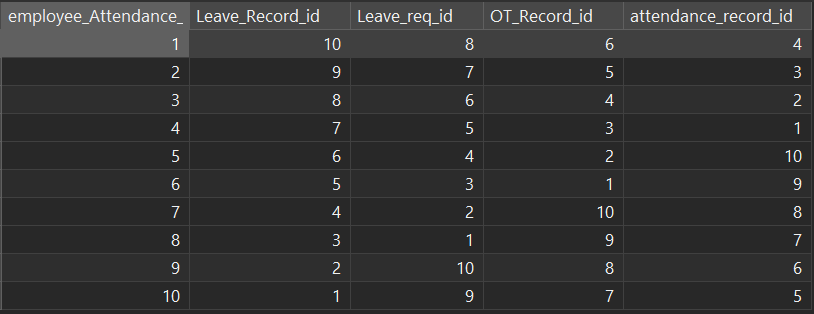
****

SELECT \* FROM employee\_attendance;

***Before:***

****

***After:***

****

**UPDATE TABLE skillset**

**UPDATE Commands:**

-- skillset

UPDATE skillset SET emp\_pnd\_id = 6 WHERE skillset\_id = 1;

UPDATE skillset SET emp\_pnd\_id = 7 WHERE skillset\_id = 2;

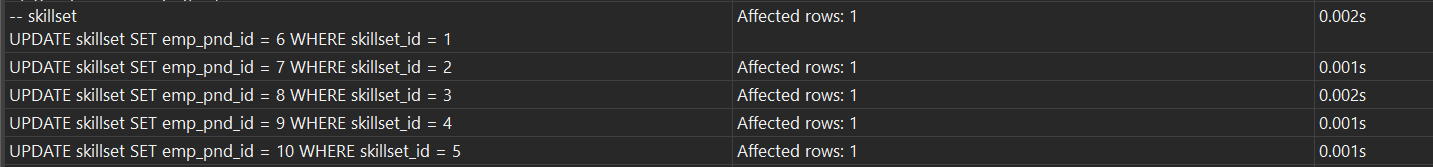
UPDATE skillset SET emp\_pnd\_id = 8 WHERE skillset\_id = 3;

UPDATE skillset SET emp\_pnd\_id = 9 WHERE skillset\_id = 4;

UPDATE skillset SET emp\_pnd\_id = 10 WHERE skillset\_id = 5;

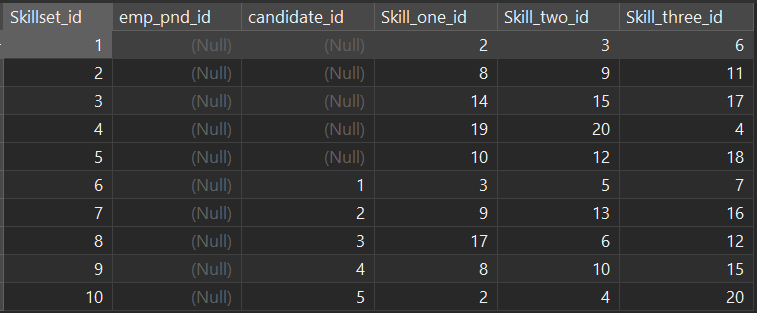
**Excuted Result (with Query Commands):**

(Excute successfully)

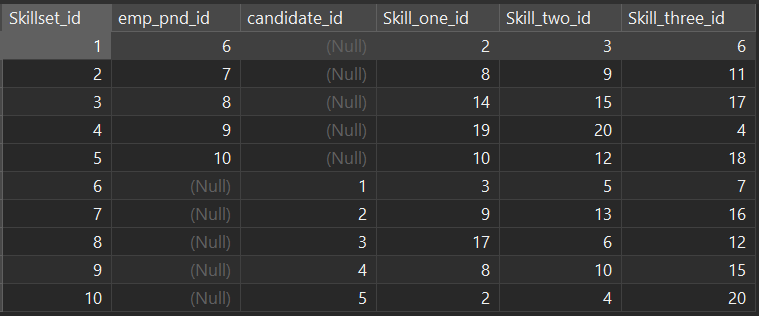
****

SELECT \* FROM skillset;

***Before:***

****

***After:***

****

**UPDATE TABLE employee\_performance\_and\_development**

**UPDATE Commands:**

-- employee\_performance\_and\_development

UPDATE `employee\_performance\_and\_development` SET

`employee\_promotion\_id` = 2,

`credit\_score\_grader\_id` = 3

WHERE `emp\_pnd\_id` = 1;

UPDATE `employee\_performance\_and\_development` SET

`employee\_promotion\_id` = 5,

`credit\_score\_grader\_id` = 4

WHERE `emp\_pnd\_id` = 2;

UPDATE `employee\_performance\_and\_development` SET

`employee\_promotion\_id` = 8,

`credit\_score\_grader\_id` = 9

WHERE `emp\_pnd\_id` = 3;

UPDATE `employee\_performance\_and\_development` SET

`employee\_promotion\_id` = 1,

`credit\_score\_grader\_id` = 7

WHERE `emp\_pnd\_id` = 4;

UPDATE `employee\_performance\_and\_development` SET

`employee\_promotion\_id` = 10,

`credit\_score\_grader\_id` = 6

WHERE `emp\_pnd\_id` = 5;

UPDATE `employee\_performance\_and\_development` SET

`employee\_promotion\_id` = 3,

`credit\_score\_grader\_id` = 5

WHERE `emp\_pnd\_id` = 6;

UPDATE `employee\_performance\_and\_development` SET

`employee\_promotion\_id` = 6,

`credit\_score\_grader\_id` = 8

WHERE `emp\_pnd\_id` = 7;

UPDATE `employee\_performance\_and\_development` SET

`employee\_promotion\_id` = 9,

`credit\_score\_grader\_id` = 1

WHERE `emp\_pnd\_id` = 8;

UPDATE `employee\_performance\_and\_development` SET

`employee\_promotion\_id` = 4,

`credit\_score\_grader\_id` = 2

WHERE `emp\_pnd\_id` = 9;

UPDATE `employee\_performance\_and\_development` SET

`employee\_promotion\_id` = 7,

`credit\_score\_grader\_id` = 10

WHERE `emp\_pnd\_id` = 10;

**Excuted Result (with Query Commands):**

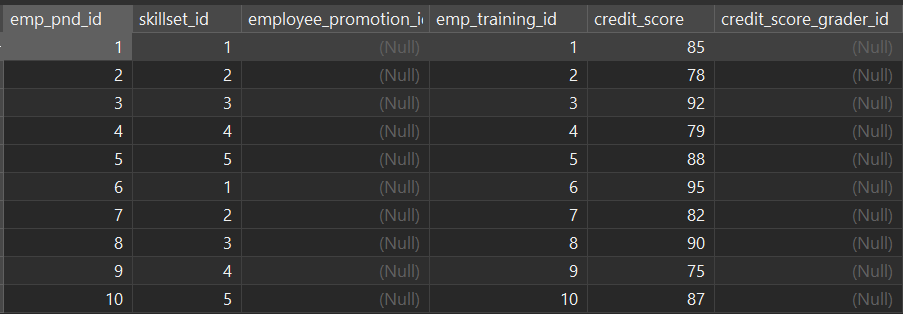
(Excute successfully)

****

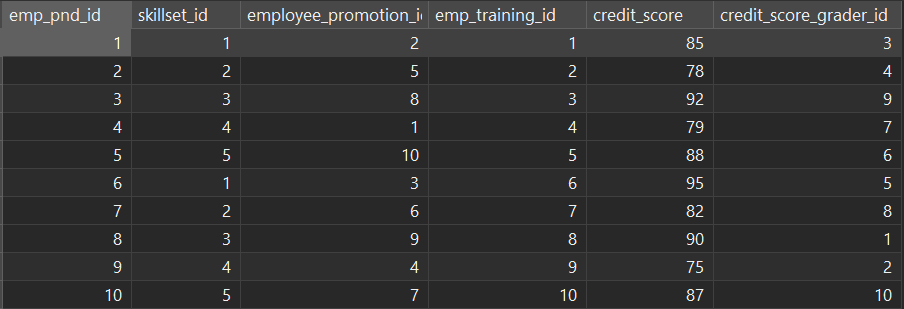
****

SELECT \* FROM employee\_performance\_and\_development;

***Before:***

****

***After:***



**2.3.3 DELETE**

**DELETE(data that matches conditions)**

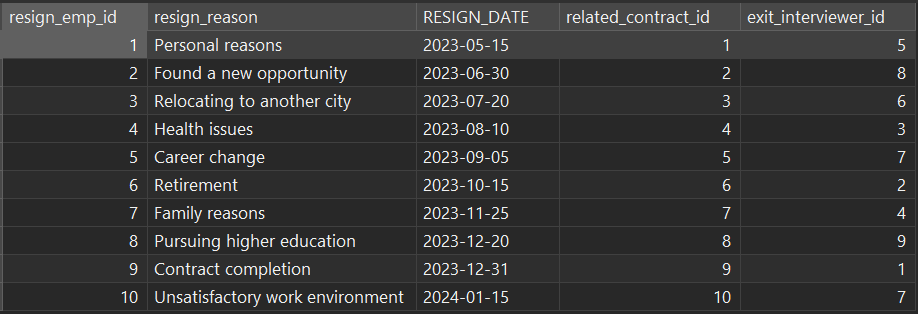
**DELETE Commands:**

DELETE FROM resignations WHERE resign\_emp\_id = 5;

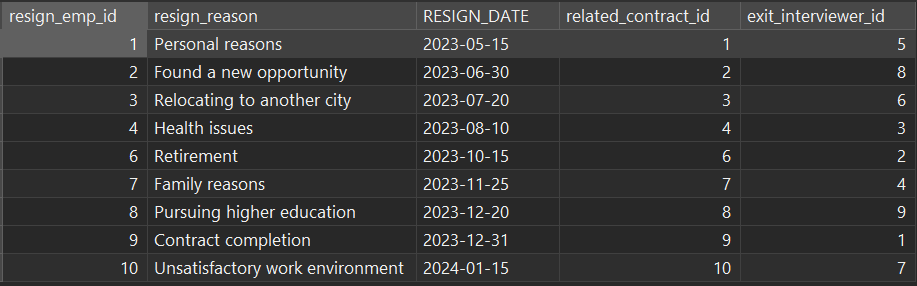
**Excuted Result (with Query Commands):**

SELECT \* FROM resignations;

***Before:***

****

***After:***



**DELETE(all data)**

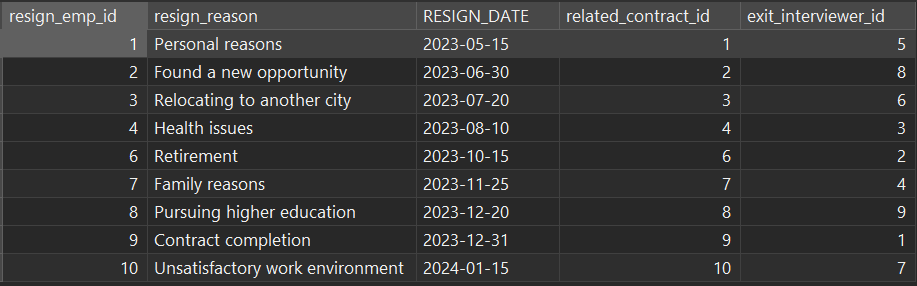
**DELETE Commands:**

DELETE FROM resignations;

**Excuted Result (with Query Commands):**

SELECT \* FROM resignations;

***Before:***



***After:***

****

**2.3.4 SELECT**

- In MySQL, SELECT is a query statement used to retrieve data from the database. It allows us to get the required data from the table based on specific conditions and requirements. The SELECT statement is one of the most commonly used operations in databases, which makes it easy and efficient to obtain information from complex database structures.

- For this human resource management system, we designed 7 efficient queries exmple using the SELECT statement to meet different data analysis and management needs.

1. Retrieves all records for all columns from the "departments" table.

**SELECT Commands:**

SELECT \* FROM departments;

**Excuted Result (with Query Commands):**

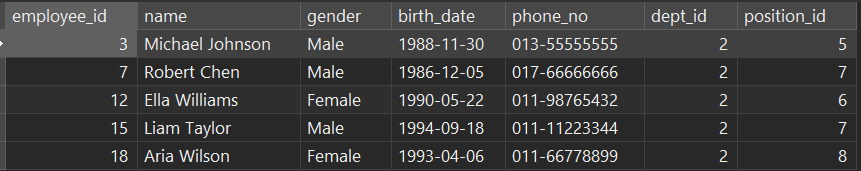
****

1. Retrieve all information about department number 2 from the "employees" table.

**SELECT Commands:**

SELECT \* FROM employees WHERE dept\_id = 2;

**Excuted Result (with Query Commands):**

****

1. Retrieve the employee ID, name, and department name from the "employees" and "departments" tables and join on the "dept\_id" field.

**SELECT Commands:**

SELECT e.emp\_id, e.name, d.dept\_name FROM employees e

JOIN departments d ON e.dept\_id = d.dept\_id;

**Excuted Result (with Query Commands):**

****

1. Retrieve the numbers and names of employees who work in the "Department of Information Technology" department from the "employees" table.

**SELECT Commands:**

SELECT emp\_id, name FROM employees

WHERE dept\_id IN (SELECT dept\_id FROM departments WHERE dept\_name = 'Department of Information Technology');

**Excuted Result (with Query Commands):**

****

1. Group by department from the "employees" table and count the number of employees in each department.

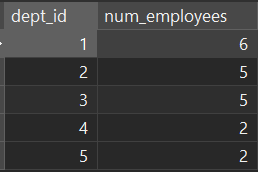
**SELECT Commands:**

SELECT dept\_id, COUNT(\*) AS num\_employees

FROM employees

GROUP BY dept\_id;

**Excuted Result (with Query Commands):**

****

1. Sort by employee number in descending order from the "employees" table and retrieve the employee number and name (retirive 10 data only).

**SELECT Commands:**

SELECT emp\_id, name FROM employees ORDER BY emp\_id DESC LIMIT 10;

**Excuted Result (with Query Commands):**

****

1. Calculates the average salary of all employees of department IT and displays the result as "avg\_salary".

**SELECT Commands:**

SELECT d.dept\_id, d.dept\_name, AVG(esd.BASE\_SALARY) AS average\_salary

FROM employees e

LEFT JOIN departments d ON e.dept\_id = d.dept\_id

LEFT JOIN employee\_benefits eb ON e.emp\_id = eb.Employee\_Benefits\_id

LEFT JOIN employees\_salary\_details esd ON eb.Employee\_Benefits\_id = esd.Employee\_Benefits\_id

WHERE d.dept\_name = 'Department of information technology'

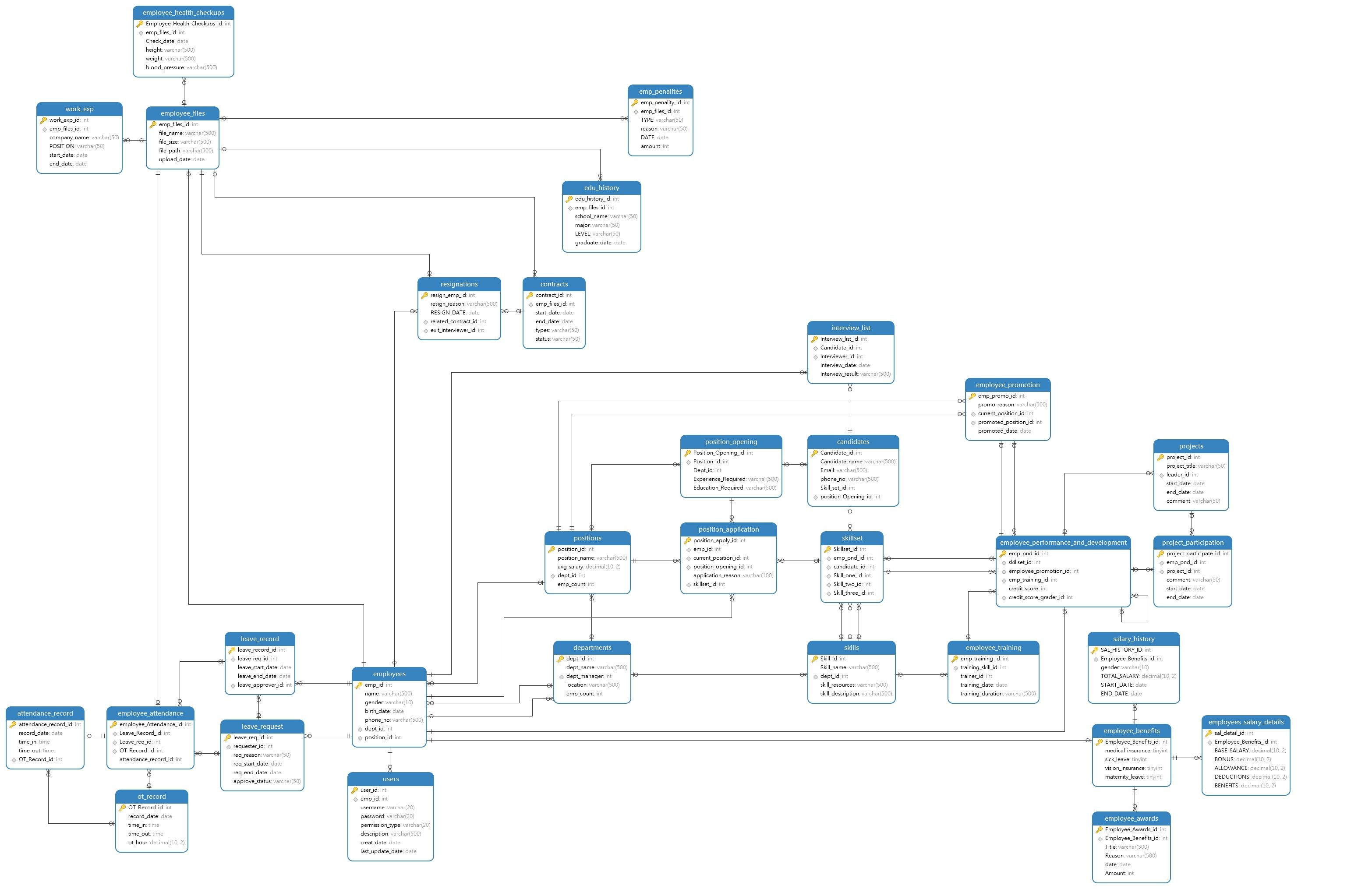
GROUP BY d.dept\_id, d.dept\_name;

**Excuted Result (with Query Commands):**

****

1. **Diagrams**

**3.1 ER Diagram**

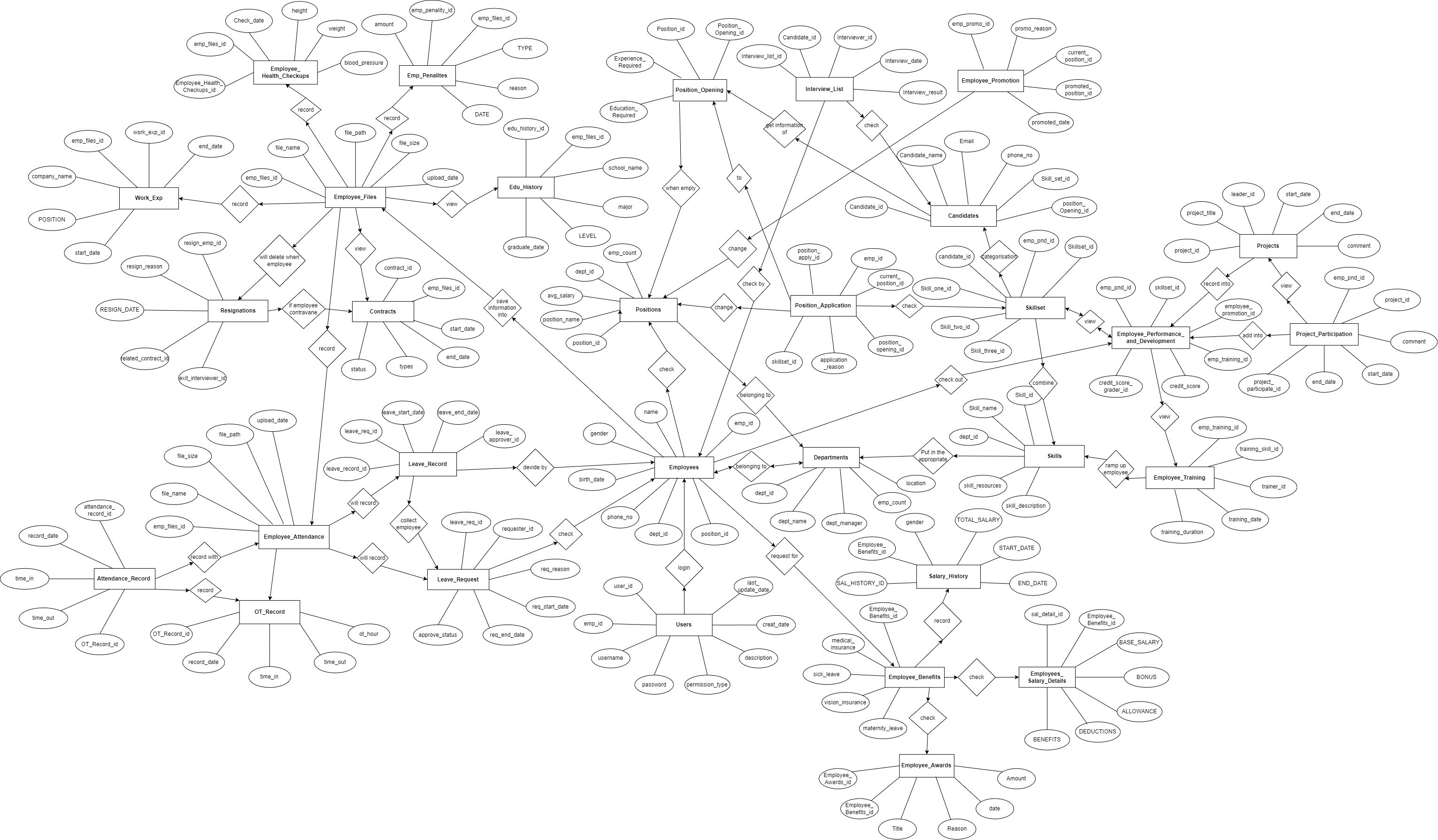


(Clearer picture in folder “extra documents”)

**3.2 Table Relationship**

|  |  |  |
| --- | --- | --- |
| **Table A** | **Relationship** | **Table B** |
| Employee health checkup | Many to one | Employee files |
| Work exp | Many to one | Employee files |
| Emp penalites | Many to one | Employee files |
| Edu history | Many to one | Employee files |
| contract | Many to one | Employee files |
| resignations | Many to one | contract |
| resignations | Many to one | Employee files |
| resignations | one to Many | Employees |
| Employee files | one to one | Employees |
| Employee files | one to Many | Employee attendance |
| Attendance record | Many to one | Employee attendance |
| Ot record | Many to one | Attendance record |
| Employee attendance | one to Many | Leave request |
| Leave record | One to one | Leave request |
| Leave record | Many to one | Employees |
| Leave request | Many to one | Employees |
| Users | one to one | Employees |
| Employees | One to Many | Employee benefits |
| Salary history | Many to one | Employee benefits |
| Employees salary details | Many to one | Employee benefits |
| Employee awards | Many to one | Employee benefits |
| Interview list | Many to one | Employees |
| Employees | Many to one | Positions |
| Position application | Many to one | Employees |
| Departments | one to many | Employees |
| Employees | One to many | Employee performance and development |
| Positions | Many to one | Departments |
| Position opening | many to one | Positions |
| Employee promotion | Many to one | Positions |
| Interview list | Many to one | Candidates |
| Position application | Mnay to one | Positions |
| skills | Many to one | Departments |
| Candidates | Many to one | Position opening |
| Position application | Many to one | Position opening |
| Position application | Many to one | Skillset |
| Skillset | Many to Many | Candidates |
| Skills | Many to one | Skillset |
| Skillset | Many to many | Employee performance and development |
| Employee performance and development | Many to many | Skillset |
| Employee performance and development | Many to one | Employee promotion |
| Employee training | many to one | skills |
| Employee performance and development | One to many | Employee training |
| Employee performance and development | Many to one | Employee performance and development |
| Project participation | one to many | Employee performance and development |
| Projects | one to many | Employee performance and development |
| Project participation | Many to one | Projects |

**3.3 Chen ER Diagram**



(Clearer picture in folder “extra documents”)

**3.4 Data Flow Diagram (DFD)**

**Explanation of 2 type Permissions**

- There are 2 types of permission, which is general user(normal employee) and administrator. Below is their fuctions:

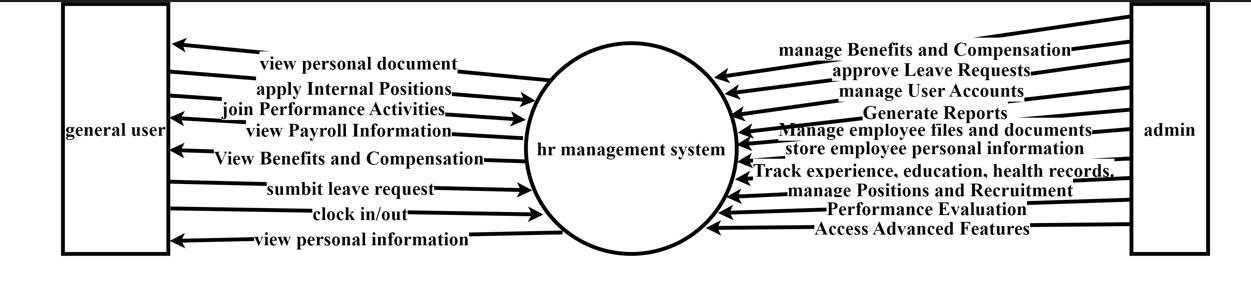
**General User (Normal Employee):**

* clock In/Ot: employee can take attendant and overtime record
* view employee file: view employee file
* summit leave / resignation form: employee can summit leave and resignation file when their want
* participate in Performance Activities: employee can participate in project in training
* apply for Internal Positions: employee also can apply for higher position

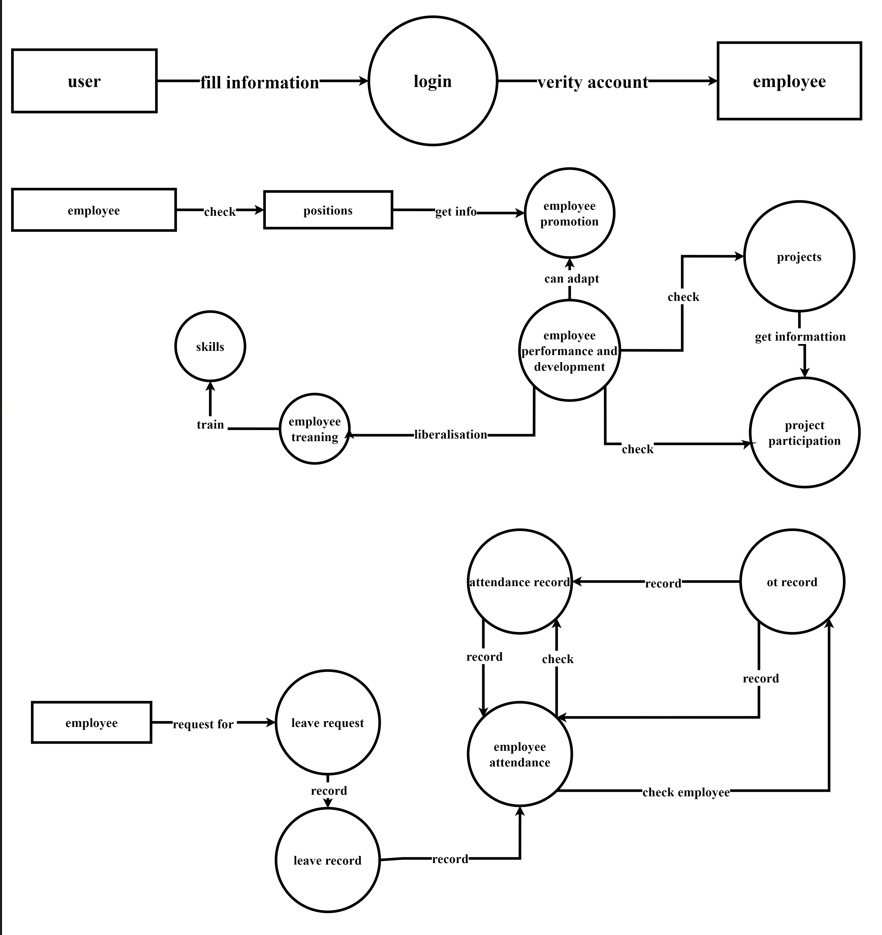
**Administrator:**

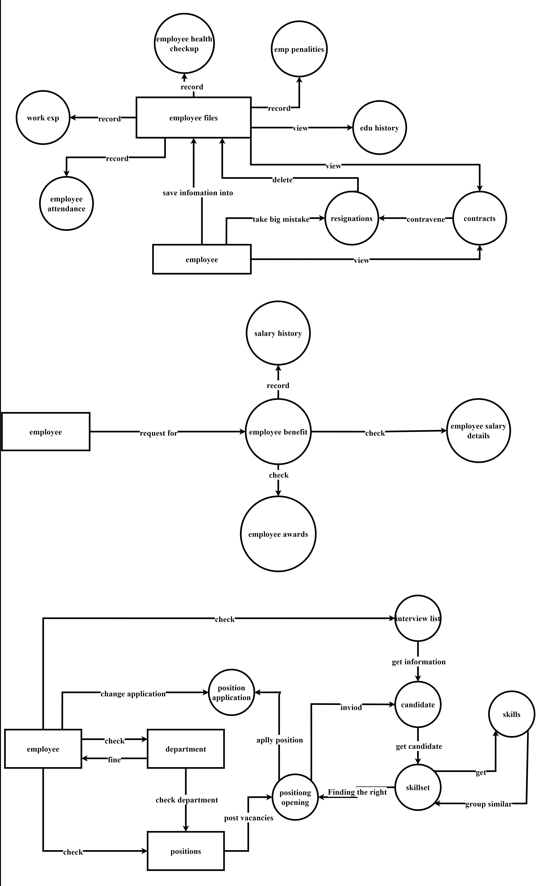
* manage Positions and Recruitment: manage employee position and hiring candidates
* manage employee development: manage employee training
* manage form: accept or reject leave form and resignation form
* manage User Accounts: can update employee username and password
* manage employee file: update all employee file
* give penalites: record employee penalites
* manage salary history / detail: manage employee salary history and detail
* manage Benefits and Compensation: manage employee award and benefit

**Level 0**

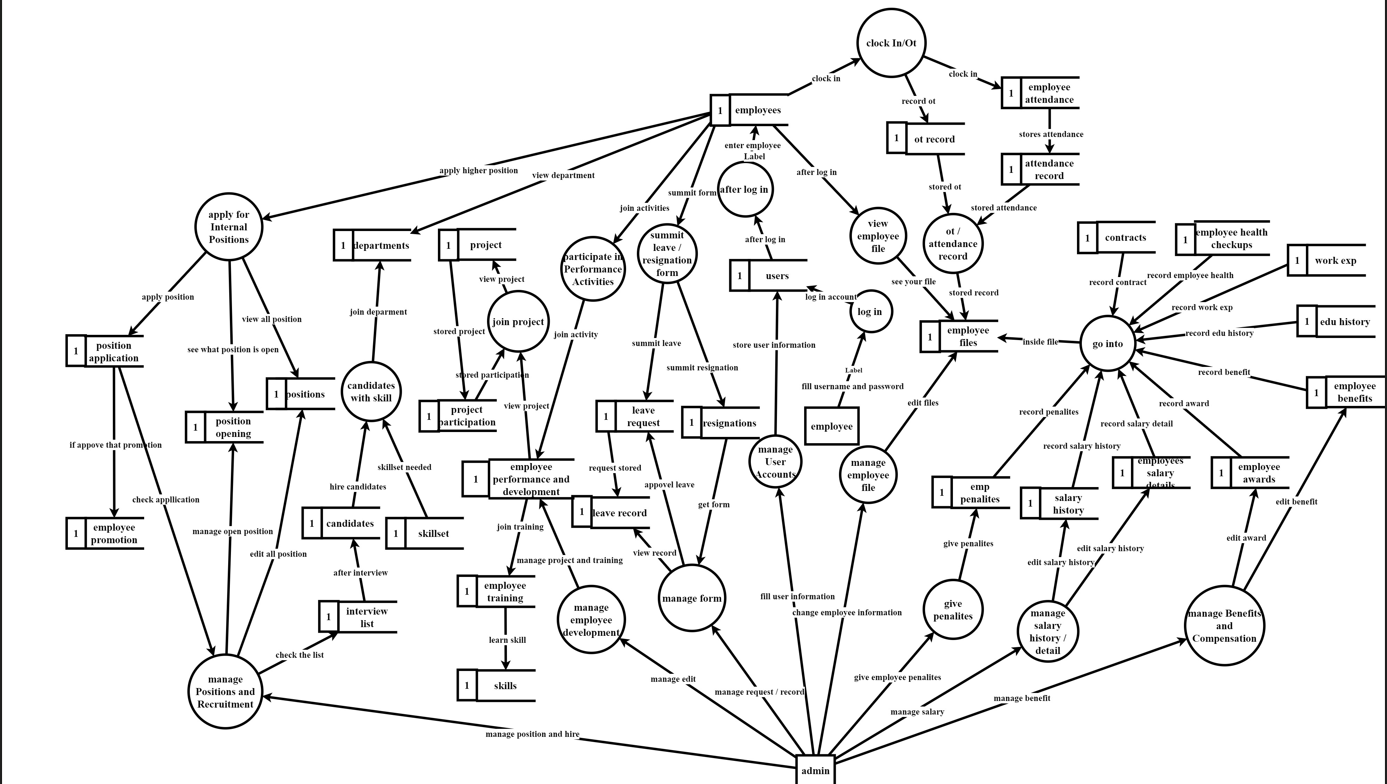


**Level 1**





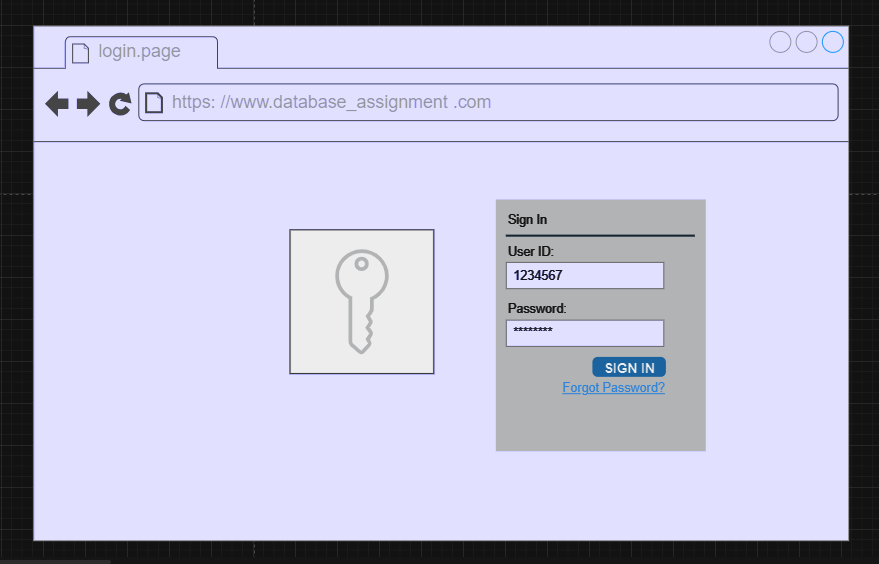
**Level 2**

****

(Clearer picture in folder “extra documents”)

**3.5 Mock-ups**

**Login Page**



The company's login page provides a convenient and secure way for employees to access their accounts. During the login process, employees simply enter their unique employee ID and password to quickly access the system. However, if they forget their password, there is a prominent 'Forgot Password' option on the page. By clicking on this option, employees can initiate the password recovery process, which will guide them through the identity verification process to reset the password.

Once the correct password has been successfully entered, the system automatically redirects to a dedicated employee verification interface. This interface not only further confirms the employee's identity, but also provides personalized functionality and information. Through this dual verification approach, the company ensures the security of the login process and prevents unauthorized access. In summary, this login page combines convenience and security, providing employees with an efficient way to access the company's systems.

**User Confirm Page**

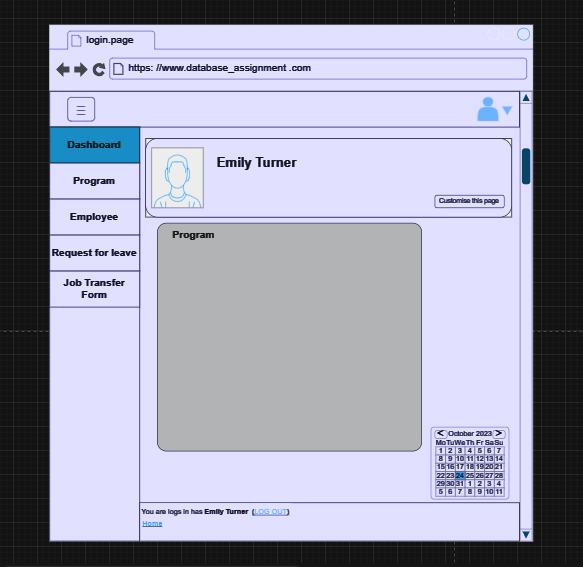


After entering their password and successfully logging in, the system will direct employees to a specially designed employee verification interface. This interface allows employees to view their name, employee ID, gender, position, and contact number, as well as their personal profile picture for further identity verification.

Within this employee verification interface, there is a checkbox labelled 'I confirm it's me'. Employees simply need to tick this box to confirm that the information provided is accurate and that they are currently accessing their own accounts. Once employees have ticked the 'I confirm it's me' box, a 'Continue' button will appear below.

After confirming the accuracy of the information and ticking the confirmation box, employees must click the 'Continue' button. When this button is clicked, the system will validate the information provided and ensure that the employee's identity is confirmed. Upon successful validation, employees will be directed to their personalized employee page. Within this employee page, they will be able to access and manage personal information, perform specific job tasks and use other functionalities related to their position.

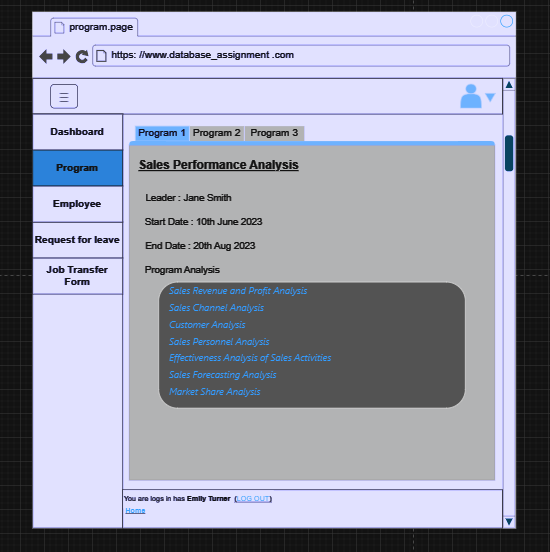
**User Page**



Once an employee has successfully logged in, they are taken to their personal interface, which displays their avatar and name. On the left-hand side of the page, there is a multi-function sidebar which, when clicked, displays five options, namely 'Dashboard', 'Program', 'Employee', 'Request for leave', and 'Job Transfer Form'. The screen shows the progress of the employee's tasks in detail, as well as their calendar.

At the bottom of the page, the information of the currently logged-in user is displayed in real-time and a logout option is provided to allow the employee to safely exit the system. The user interface is designed to provide an intuitive and personalized work environment for employees to manage tasks, view project progress, manage employee information, request leave and perform job transfers. Through this interface, employees can not only organize their work efficiently but also keep track of their work schedules and team activities. Overall, the company's web design provides employees with a central, easy-to-use work platform that helps improve productivity and collaboration.

**Program Page**

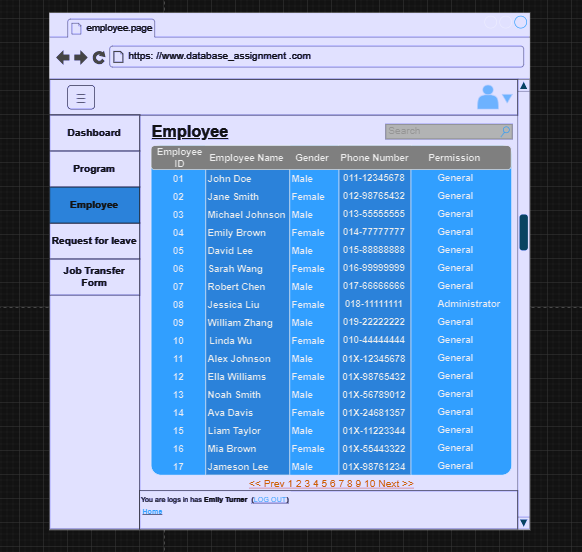


The program page on the company website provides a comprehensive overview of the projects in which employees are currently involved. On this page, employees can easily see a clear list of the projects they are involved in, providing a visual representation of their ongoing commitments.

Each project entry includes key information such as the project manager, start and end dates, and detailed project activities. This allows employees to quickly grasp key project details, including leadership and planned timeframes.

They can also easily click on each project to access more detailed information. Within the project-specific page, employees can delve into the internal structure of the project, including specific tasks, activities, and objectives. This enables people to better understand the project's requirements and goals, as well as their specific roles within the project.

**Employee Member Page**



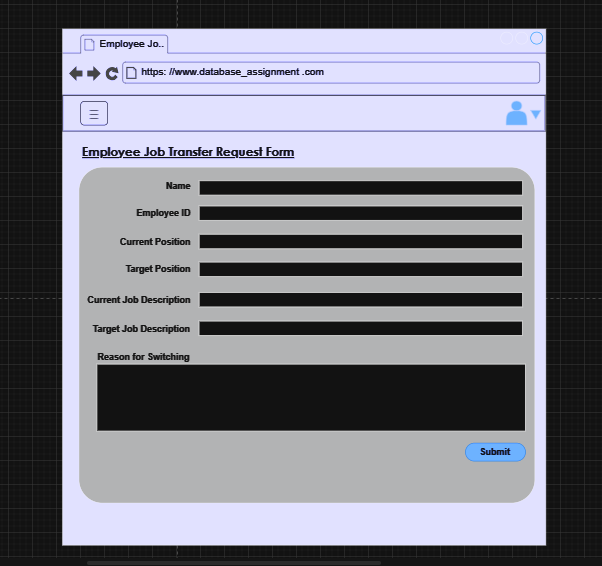
The 'Employees' page on the company website acts as a comprehensive directory, providing a complete list of all employees. This table displays essential details for each employee, including their unique employee ID, full name, gender, contact number, and assigned permissions. This dedicated page greatly streamlines and organizes the management of employee information.

A notable feature of this page is its powerful search bar functionality. This is particularly useful when HR needs to locate specific employees or perform specific tasks. By simply typing an employee's name or ID into the search bar, HR can quickly locate and retrieve the required employee details. This robust search capability not only saves significant time but also increases overall efficiency.

The design philosophy behind the Employee page is to provide the organization with a centralized tool for managing employee information. This allows HR and other relevant personnel to seamlessly navigate and access relevant employee information. Whether for routine administrative tasks or to carry out specific tasks, this page provides a repository of resources that is both efficient and user-friendly.

Essentially, this site enables the organization to manage employee information more effectively, resulting in improved operational efficiency. By addressing a variety of human resource management needs, the Employee page plays a key role in supporting the company's efforts, fostering efficient communication, and ensuring that vital personnel information is easily accessible and well organized.

**Employee Job Transfer Request Page (Position Application)**



Within the company's website, the 'Employee Job Transfer Request Form' page provides the Human Resources (HR) department with a seamless way to manage employee position changes. This dedicated page simplifies the process of changing an employee's role and provides a comprehensive set of fields for detailed input. This page displays the employee's name, employee ID, current position, target position, descriptions of the current and target positions, and the reason for the transfer.

During the completion process, HR can enter accurate employee details, including name and employee ID, to ensure the identity of the person being transferred. In the Current and Target Position fields, HR must explicitly state the employee's current role and desired new position. In addition, the inclusion of position description fields ensures a clear understanding of post-transfer responsibilities for both the employee and HR.

Once the form has been completed and reviewed, a simple click on the 'Submit' button initiates the change to the employee's position details. At the same time, the system immediately distributes notifications to the relevant employees, ensuring that all parties are aware of the impending change.

In essence, this "Employee Job Transfer Request Form" page serves as an efficient and centralized tool for the HR department to facilitate the management of employee job transfers. This functionality allows the organization to smoothly adapt to internal and external changes while facilitating the professional growth of employees. The automated notification mechanism also promotes transparent communication and efficiency throughout the process.

**Leave Request Page**



The Request for Leave page on the company's website is an efficient way for employees to submit their leave requests. This page provides a comprehensive form that requires employees to enter a number of necessary details to initiate the leave request. Within the form, employees are asked to provide their name, position, type of leave, start and end dates, contact number, and reason for leave.

During the completion process, employees must enter their information accurately, including their name and position, to ensure it matches their identity. In the 'type of leave' field, employees must specify the exact category of leave they are requesting, such as annual leave, sick leave, or personal leave. At the same time, employees must provide the start and end dates of the leave period, as well as their contact number for HR communications.

A brief description of the reason for the leave is also required so that HR can understand the circumstances behind the request. Once the form is completed, employees simply click the "Submit" button. Once submitted, the system automatically forwards the leave request to HR for approval.

The Request Leave page has been designed to provide employees with a quick and transparent method of requesting time off. This page not only makes it convenient for employees to initiate leave requests but also ensures that their requests are processed in a timely manner. This helps the company manage its leave process effectively while providing a streamlined communication channel to improve information sharing and collaboration.