



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

1.0 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.

2.0 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_penalty_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)

CREATE DATABASE human_resources_management_system	OK	0.003s
---	----	--------

USE human_resources_management_system	OK	0s
---------------------------------------	----	----

```
SELECT DATABASE();
```

► DATABASE()
► human_resource_management_system

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)

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);	OK	0.113s
--	----	--------

DESC attendance_record;

Field	Type	Null	Key	Default	Extra
► attendance_record_id	int	NO	PRI	(Null)	auto_increment
record_date	date	YES		(Null)	
time_in	time	YES		(Null)	
time_out	time	YES		(Null)	
OT_Record_id	int	YES	MUL	(Null)	

SELECT * FROM attendance_record;

attendance_record_id	record_date	time_in	time_out	OT_Record_id
	(N/A)	(N/A)	(N/A)	(N/A)

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)

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, `position_Opening_id` int NULL) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_0900_ai_ci ROW_FORMAT=DYNAMIC;	OK	0.021s
---	----	--------

DESC candidates;

Field	Type	Null	Key	Default	Extra
► Candidate_id	int	NO	PRI	(Null)	auto_increment
Candidate_name	varchar(500)	YES		(Null)	
Email	varchar(500)	YES		(Null)	
phone_no	varchar(500)	YES		(Null)	
Skill_set_id	int	YES		(Null)	
position_Opening_id	int	YES	MUL	(Null)	

SELECT * FROM candidates;

Candidate_id	Candidate_name	Email	phone_no	Skill_set_id	position_Opening_id
	(N/A)	(N/A)	(N/A)	(N/A)	(N/A)

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)

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) NULL, `status` varchar(50) NULL);	OK	0.019s
---	----	--------

DESC contracts;

Field	Type	Null	Key	Default	Extra
▶ contract_id	int	NO	PRI	(Null)	auto_increment
emp_files_id	int	YES	MUL	(Null)	
start_date	date	YES		(Null)	
end_date	date	YES		(Null)	
types	varchar(50)	YES		(Null)	
status	varchar(50)	YES		(Null)	

SELECT * FROM contracts;

contract_id	emp_files_id	start_date	end_date	types	status
	(N/A)	(N/A)	(N/A)	(N/A)	(N/A)

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)

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,) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;	OK	0.016s
---	----	--------

DESC departments;

Field	Type	Null	Key	Default	Extra
dept_id	int	NO	PRI	(Null)	auto_increment
dept_name	varchar(500)	YES		(Null)	
dept_manager	int	YES	MUL	(Null)	
location	varchar(500)	YES		(Null)	
emp_count	int	YES		(Null)	

SELECT * FROM departments;

dept_id	dept_name	dept_manager	location	emp_count
	(N/A)	(N/A)	(N/A)	(N/A)

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)

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) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;	OK	0.014s
--	----	--------

DESC edu_history;

Field	Type	Null	Key	Default	Extra
edu_history_id	int	NO	PRI	(Null)	auto_increment
emp_files_id	int	YES	MUL	(Null)	
school_name	varchar(50)	YES		(Null)	
major	varchar(50)	YES		(Null)	
LEVEL	varchar(50)	YES		(Null)	
graduate_date	date	YES		(Null)	

SELECT * FROM edu_history;

edu_history_id	emp_files_id	school_name	major	LEVEL	graduate_date
	(N/A)	(N/A)	(N/A)	(N/A)	(N/A)

CREATE TABLE Commands:

```
CREATE TABLE `emp_penalites` (
    `emp_penalty_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_penalty_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)

CREATE TABLE `emp_penalites` (`emp_penalty_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) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;	OK	0.018s
--	----	--------

DESC emp_penalites;

Field	Type	Null	Key	Default	Extra
emp_penalty_id	int	NO	PRI	(Null)	auto_increment
emp_files_id	int	YES	MUL	(Null)	
TYPE	varchar(50)	YES		(Null)	
reason	varchar(50)	YES		(Null)	
DATE	date	YES		(Null)	
amount	int	YES		(Null)	

SELECT * FROM emp_penalites;

emp_penalty_id	emp_files_id	TYPE	reason	DATE	amount
	(N/A)	(N/A)	(N/A)	(N/A)	(N/A)

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)

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) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_0900_ai_ci;	OK	0.018s
--	----	--------

DESC employee_attendance;

Field	Type	Null	Key	Default	Extra
employee_Attendance_id	int	NO	PRI	(Null)	auto_increment
Leave_Record_id	int	YES	MUL	(Null)	
Leave_req_id	int	YES	MUL	(Null)	
OT_Record_id	int	YES	MUL	(Null)	
attendance_record_id	int	YES		(Null)	

SELECT * FROM employee_attendance;

employee_Attendance_id	Leave_Record_id	Leave_req_id	OT_Record_id	attendance_record_id
	(N/A)	(N/A)	(N/A)	(N/A)

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)

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) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;	OK	0.015s
---	----	--------

DESC employee_awards;

Field	Type	Null	Key	Default	Extra
Employee_Awards_id	int	NO	PRI	(Null)	auto_increment
Employee_Benefits_id	int	NO	MUL	(Null)	
Title	varchar(500)	YES		(Null)	
Reason	varchar(500)	YES		(Null)	
date	date	YES		(Null)	
Amount	int	YES		(Null)	

SELECT * FROM employee_awards;

Employee_Awards_id	Employee_Benefits_id	Title	Reason	date	Amount
		(N/A)	(N/A)	(N/A)	(N/A)

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)

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) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_0900_ai_ci;	OK	0.014s
---	----	--------

DESC employee_benefits;

Field	Type	Null	Key	Default	Extra
Employee_Benefits_id	int	NO	PRI	(Null)	auto_increment
medical_insurance	tinyint	YES		(Null)	
sick_leave	tinyint	YES		(Null)	
vision_insurance	tinyint	YES		(Null)	
maternity_leave	tinyint	YES		(Null)	

SELECT * FROM employee_benefits;

Employee_Benefits_id	medical_insurance	sick_leave	vision_insurance	maternity_leave
	(N/A)	(N/A)	(N/A)	(N/A)

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)

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) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_0900_ai_ci ROW_FORMAT=DYNAMIC;	OK	0.014s
--	----	--------

DESC employee_files;

Field	Type	Null	Key	Default	Extra
emp_files_id	int	NO	PRI	(Null)	auto_increment
file_name	varchar(500)	YES		(Null)	
file_size	varchar(500)	YES		(Null)	
file_path	varchar(500)	YES		(Null)	
upload_date	date	YES		(Null)	

SELECT * FROM employee_files;

emp_files_id	file_name	file_size	file_path	upload_date
	(N/A)	(N/A)	(N/A)	(N/A)

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)

CREATE TABLE `employee_health_checkups` (OK	0.018s
'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)		
'weight' varchar(500)		
'blood_pressure' varchar(500)		

DESC employee_health_checkups;

Field	Type	Null	Key	Default	Extra
Employee_Health_Checkups_id	int	NO	PRI	(Null)	auto_increment
emp_files_id	int	YES	MUL	(Null)	
Check_date	date	YES		(Null)	
height	varchar(500)	YES		(Null)	
weight	varchar(500)	YES		(Null)	
blood_pressure	varchar(500)	YES		(Null)	

SELECT * FROM employee_health_checkups;

Employee_Health_Checkups_id	emp_files_id	Check_date	height	weight	blood_pressure
	(N/A)	(N/A)	(N/A)	(N/A)	(N/A)

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)

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) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;	OK	0.03s
--	----	-------

DESC employee_performance_and_development;

Field	Type	Null	Key	Default	Extra
emp_pnd_id	int	NO	PRI	(Null)	auto_increment
skillset_id	int	YES	MUL	(Null)	
employee_promotion_id	int	YES	MUL	(Null)	
emp_training_id	int	YES	MUL	(Null)	
credit_score	int	YES		(Null)	
credit_score_grader_id	int	YES	MUL	(Null)	

SELECT * FROM employee_performance_and_development;

emp_pnd_id	skillset_id	employee_promotion_id	emp_training_id	credit_score	credit_score_grader_id
	(N/A)	(N/A)	(N/A)	(N/A)	(N/A)

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)

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,) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;	OK	0.015s
--	----	--------

DESC employee_promotion;

Field	Type	Null	Key	Default	Extra
▶ emp_promo_id	int	NO	PRI	(Null)	auto_increment
promo_reason	varchar(500)	YES		(Null)	
current_position_id	int	NO	MUL	(Null)	
promoted_position_id	int	NO	MUL	(Null)	
promoted_date	date	YES		(Null)	

SELECT * FROM employee_promotion;

emp_promo_id	promo_reason	current_position_id	promoted_position_id	promoted_date
(N/A)		(N/A)	(N/A)	(N/A)

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)

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, `training_duration` varchar(500) NULL) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;	OK	0.013s
--	----	--------

DESC employee_training;

Field	Type	Null	Key	Default	Extra
emp_training_id	int	NO	PRI	(Null)	auto_increment
training_skill_id	int	YES	MUL	(Null)	
trainer_id	int	YES		(Null)	
training_date	date	YES		(Null)	
training_duration	varchar(500)	YES		(Null)	

SELECT * FROM employee_training;

emp_training_id	training_skill_id	trainer_id	training_date	training_duration
	(N/A)	(N/A)	(N/A)	(N/A)

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)

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) `birth_date` date `phone_no` varchar(500) `dept_id` int `position_id` int	OK	0.019s
---	----	--------

DESC employees;

Field	Type	Null	Key	Default	Extra
▶ emp_id	int	NO	PRI	(Null)	auto_increment
name	varchar(500)	YES		(Null)	
gender	varchar(10)	YES		(Null)	
birth_date	date	YES		(Null)	
phone_no	varchar(500)	YES		(Null)	
dept_id	int	YES	MUL	(Null)	
position_id	int	YES	MUL	(Null)	

SELECT * FROM employees;

emp_id	name	gender	birth_date	phone_no	dept_id	position_id
	(N/A)	(N/A)	(N/A)	(N/A)	(N/A)	(N/A)

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)

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,	OK	0.015s
---	----	--------

DESC employees_salary_details;

Field	Type	Null	Key	Default	Extra
► sal_detail_id	int	NO	PRI	(Null)	auto_increment
Employee_Benefits_id	int	NO	MUL	(Null)	
BASE_SALARY	decimal(10,2)	YES		(Null)	
BONUS	decimal(10,2)	YES		(Null)	
ALLOWANCE	decimal(10,2)	YES		(Null)	
DEDUCTIONS	decimal(10,2)	YES		(Null)	
BENEFITS	decimal(10,2)	YES		(Null)	

SELECT * FROM employees_salary_details;

sal_detail_id	Employee_Benefits_id	BASE_SALARY	BONUS	ALLOWANCE	DEDUCTIONS	BENEFITS
	(N/A)	(N/A)	(N/A)	(N/A)	(N/A)	(N/A)

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)

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, `Interview_result` varchar(500)) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;	OK	0.015s
--	----	--------

DESC interview_list;

Field	Type	Null	Key	Default	Extra
▶ Interview_list_id	int	NO	PRI	(Null)	auto_increment
Candidate_id	int	NO	MUL	(Null)	
Interviewer_id	int	NO	MUL	(Null)	
Interview_date	date	YES		(Null)	
Interview_result	varchar(500)	YES		(Null)	

SELECT * FROM interview_list;

Interview_list_id	Candidate_id	Interviewer_id	Interview_date	Interview_result
	(N/A)	(N/A)	(N/A)	(N/A)

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)

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 NULL DEFAULT NULL) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_0900_ai_ci;	OK	0.015s
--	----	--------

DESC leave_record;

Field	Type	Null	Key	Default	Extra
leave_record_id	int	NO	PRI	(Null)	auto_increment
leave_req_id	int	YES	MUL	(Null)	
leave_start_date	date	YES		(Null)	
leave_end_date	date	YES		(Null)	
leave_approver_id	int	NO	MUL	(Null)	

SELECT * FROM leave_record;

leave_record_id	leave_req_id	leave_start_date	leave_end_date	leave_approver_id
	(N/A)	(N/A)	(N/A)	(N/A)

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)

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	OK	0.016s
---	----	--------

DESC leave_request;

Field	Type	Null	Key	Default	Extra
▶ leave_req_id	int	NO	PRI	(Null)	auto_increment
requester_id	int	NO	MUL	(Null)	
req_reason	varchar(50)	YES		(Null)	
req_start_date	date	YES		(Null)	
req_end_date	date	YES		(Null)	
approve_status	varchar(50)	YES		(Null)	

SELECT * FROM leave_request;

leave_req_id	requester_id	req_reason	req_start_date	req_end_date	approve_status
	(N/A)	(N/A)	(N/A)	(N/A)	(N/A)

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)

CREATE TABLE `ot_record` (OK	0.012s
`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,		

DESC ot_record;

Field	Type	Null	Key	Default	Extra
▶ OT_Record_id	int	NO	PRI	(Null)	auto_increment
record_date	date	YES		(Null)	
time_in	time	YES		(Null)	
time_out	time	YES		(Null)	
ot_hour	decimal(10,2)	YES		(Null)	

SELECT * FROM ot_record;

OT_Record_id	record_date	time_in	time_out	ot_hour
	(N/A)	(N/A)	(N/A)	(N/A)

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)

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) DEFAULT NULL, `skillset_id` int NOT NULL) ENGINE=InnoDB AUTO_INCREMENT=1 DEFAULT CHARSET=utf8mb4;	OK	0.017s
--	----	--------

DESC position_application;

Field	Type	Null	Key	Default	Extra
position_apply_id	int	NO	PRI	(Null)	auto_increment
emp_id	int	NO	MUL	(Null)	
current_position_id	int	NO	MUL	(Null)	
position_opening_id	int	NO	MUL	(Null)	
application_reason	varchar(100)	YES		(Null)	
skillset_id	int	YES	MUL	(Null)	

SELECT * FROM position_application;

position_apply_id	emp_id	current_position_id	position_opening_id	application_reason	skillset_id
	(N/A)	(N/A)	(N/A)	(N/A)	(N/A)

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)

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) NULL, `Education_Required` varchar(500) NULL);	OK	0.016s
---	----	--------

DESC position_opening;

Field	Type	Null	Key	Default	Extra
Position_Opening_id	int	NO	PRI	(Null)	auto_increment
Position_id	int	YES	MUL	(Null)	
Dept_id	int	YES		(Null)	
Experience_Required	varchar(500)	YES		(Null)	
Education_Required	varchar(500)	YES		(Null)	

SELECT * FROM position_opening;

Position_Opening_id	Position_id	Dept_id	Experience_Required	Education_Required
	(N/A)	(N/A)	(N/A)	(N/A)

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)

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, ...);	OK	0.015s
--	----	--------

DESC positions;

Field	Type	Null	Key	Default	Extra
▶ position_id	int	NO	PRI	(Null)	auto_increment
position_name	varchar(500)	YES		(Null)	
avg_salary	decimal(10,2)	YES		(Null)	
dept_id	int	YES	MUL	(Null)	
emp_count	int	YES		(Null)	

SELECT * FROM positions;

position_id	position_name	avg_salary	dept_id	emp_count
	(N/A)	(N/A)	(N/A)	(N/A)

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)

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,	OK	0.016s
--	----	--------

DESC project_participation;

Field	Type	Null	Key	Default	Extra
► project_participate_id	int	NO	PRI	(Null)	auto_increment
emp_pnd_id	int	YES	MUL	(Null)	
project_id	int	YES	MUL	(Null)	
comment	varchar(50)	YES		(Null)	
start_date	date	YES		(Null)	
end_date	date	YES		(Null)	

SELECT * FROM project_participation;

project_participate_id	emp_pnd_id	project_id	comment	start_date	end_date
	(N/A)	(N/A)	(N/A)	(N/A)	(N/A)

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)

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,	OK	0.013s
---	----	--------

DESC projects;

Field	Type	Null	Key	Default	Extra
▶ project_id	int	NO	PRI	(Null)	auto_increment
project_title	varchar(50)	YES		(Null)	
leader_id	int	YES	MUL	(Null)	
start_date	date	YES		(Null)	
end_date	date	YES		(Null)	
comment	varchar(50)	YES		(Null)	

SELECT * FROM projects;

project_id	project_title	leader_id	start_date	end_date	comment
	(N/A)	(N/A)	(N/A)	(N/A)	(N/A)

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)

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,	OK	0.015s
---	----	--------

DESC resignations;

Field	Type	Null	Key	Default	Extra
► resign_emp_id	int	NO	PRI	(Null)	auto_increment
resign_reason	varchar(500)	YES		(Null)	
RESIGN_DATE	date	YES		(Null)	
related_contract_id	int	YES	MUL	(Null)	
exit_interviewer_id	int	YES	MUL	(Null)	

SELECT * FROM resignations;

resign_emp_id	resign_reason	RESIGN_DATE	related_contract_id	exit_interviewer_id
(N/A)	(N/A)	(N/A)	(N/A)	(N/A)

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_Bene
fits_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)

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) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;	OK	0.019s
--	----	--------

DESC salary_history;

Field	Type	Null	Key	Default	Extra
SAL_HISTORY_ID	int	NO	PRI	(Null)	auto_increment
Employee_Benefits_id	int	NO	MUL	(Null)	
gender	varchar(10)	YES		(Null)	
TOTAL_SALARY	decimal(10,2)	YES		(Null)	
START_DATE	date	YES		(Null)	
END_DATE	date	YES		(Null)	

SELECT * FROM salary_history;

SAL_HISTORY_ID	Employee_Benefits_id	gender	TOTAL_SALARY	START_DATE	END_DATE
		(N/A)	(N/A)	(N/A)	(N/A)

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)

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,);	OK	0.012s
--	----	--------

DESC skills;

Field	Type	Null	Key	Default	Extra
► Skill_id	int	NO	PRI	(Null)	auto_increment
Skill_name	varchar(500)	YES		(Null)	
dept_id	int	YES	MUL	(Null)	
skill_resources	varchar(500)	YES		(Null)	
skill_description	varchar(500)	YES		(Null)	

SELECT * FROM skills;

Skill_id	Skill_name	dept_id	skill_resources	skill_description
	(N/A)	(N/A)	(N/A)	(N/A)

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)

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);	OK	0.016s
---	----	--------

DESC skillset;

Field	Type	Null	Key	Default	Extra
► Skillset_id	int	NO	PRI	(Null)	auto_increment
emp_pnd_id	int	YES	MUL	(Null)	
candidate_id	int	YES	MUL	(Null)	
Skill_one_id	int	YES	MUL	(Null)	
Skill_two_id	int	YES	MUL	(Null)	
Skill_three_id	int	YES	MUL	(Null)	

SELECT * FROM skillset;

Skillset_id	emp_pnd_id	candidate_id	Skill_one_id	Skill_two_id	Skill_three_id
	(N/A)	(N/A)	(N/A)	(N/A)	(N/A)

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)

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 UNIQUE KEY `username` (`username`), `password` varchar(20) NOT NULL, `permission_type` varchar(20) NOT NULL, `description` varchar(500) NOT NULL, `creat_date` date NOT NULL, `last_update_date` date NOT NULL);	OK	0.01s
--	----	-------

DESC users;

Field	Type	Null	Key	Default	Extra
► user_id	int	NO	PRI	(Null)	auto_increment
emp_id	int	NO	MUL	(Null)	
username	varchar(20)	NO		(Null)	
password	varchar(20)	NO		(Null)	
permission_type	varchar(20)	YES		(Null)	
description	varchar(500)	YES		(Null)	
creat_date	date	YES		(Null)	

SELECT * FROM users;

user_id	emp_id	username	password	permission_type	description	creat_date	last_update_date
	(N/A)	(N/A)	(N/A)	(N/A)	(N/A)	(N/A)	(N/A)

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 Executed Result (with Query Commands):

(Excute successfully)

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 ...	OK	0.011s
---	----	--------

DESC work_exp;

Field	Type	Null	Key	Default	Extra
▶ work_exp_id	int	NO	PRI	(Null)	auto_increment
emp_files_id	int	YES	MUL	(Null)	
company_name	varchar(50)	YES		(Null)	
POSITION	varchar(50)	YES		(Null)	
start_date	date	YES		(Null)	
end_date	date	YES		(Null)	

SELECT * FROM work_exp;

work_exp_id	emp_files_id	company_name	POSITION	start_date	end_date
	(N/A)	(N/A)	(N/A)	(N/A)	(N/A)

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:

Field	Type	Null	Key	Default	Extra
emp_id	int	NO	PRI	(Null)	auto_increment
name	varchar(500)	YES		(Null)	
gender	varchar(10)	YES		(Null)	
birth_date	date	YES		(Null)	
phone_no	varchar(500)	YES		(Null)	
dept_id	int	YES	MUL	(Null)	
position_id	int	YES	MUL	(Null)	

After:

Field	Type	Null	Key	Default	Extra
emp_id	int	NO	PRI	(Null)	auto_increment
name	varchar(500)	YES		(Null)	
gender	varchar(10)	YES		(Null)	
birth_date	date	YES		(Null)	
phone_no	varchar(500)	YES		(Null)	
dept_id	int	YES	MUL	(Null)	
position_id	int	YES	MUL	(Null)	
manager_id	int	YES		(Null)	

MODIFY COLUMN

MODIFY COLUMN Commands:

```
ALTER TABLE employees MODIFY COLUMN manager_id DATE;
```

Excuted Result (with Query Commands):

```
DESC employees;
```

Before:

Field	Type	Null	Key	Default	Extra
emp_id	int	NO	PRI	(Null)	auto_increment
name	varchar(500)	YES		(Null)	
gender	varchar(10)	YES		(Null)	
birth_date	date	YES		(Null)	
phone_no	varchar(500)	YES		(Null)	
dept_id	int	YES	MUL	(Null)	
position_id	int	YES	MUL	(Null)	
manager_id	int	YES		(Null)	

After:

Field	Type	Null	Key	Default	Extra
emp_id	int	NO	PRI	(Null)	auto_increment
name	varchar(500)	YES		(Null)	
gender	varchar(10)	YES		(Null)	
birth_date	date	YES		(Null)	
phone_no	varchar(500)	YES		(Null)	
dept_id	int	YES	MUL	(Null)	
position_id	int	YES	MUL	(Null)	
manager_id	date	YES		(Null)	

CHANGE COLUMN

CHANGE COLUMN Commands:

```
ALTER TABLE employees CHANGE COLUMN manager_id supervisor_id INT(100);
```

Excuted Result (with Query Commands):

```
DESC employees;
```

Before:

Field	Type	Null	Key	Default	Extra
emp_id	int	NO	PRI	(Null)	auto_increment
name	varchar(500)	YES		(Null)	
gender	varchar(10)	YES		(Null)	
birth_date	date	YES		(Null)	
phone_no	varchar(500)	YES		(Null)	
dept_id	int	YES	MUL	(Null)	
position_id	int	YES	MUL	(Null)	
manager_id	int	YES		(Null)	

After:

Field	Type	Null	Key	Default	Extra
emp_id	int	NO	PRI	(Null)	auto_increment
name	varchar(500)	YES		(Null)	
gender	varchar(10)	YES		(Null)	
birth_date	date	YES		(Null)	
phone_no	varchar(500)	YES		(Null)	
dept_id	int	YES	MUL	(Null)	
position_id	int	YES	MUL	(Null)	
supervisor_id	int	YES		(Null)	

DROP COLUMN

DROP COLUMN Commands:

```
ALTER TABLE employees DROP COLUMN supervisor_id;
```

Excuted Result (with Query Commands):

```
DESC employees;
```

Before:

Field	Type	Null	Key	Default	Extra
emp_id	int	NO	PRI	(Null)	auto_increment
name	varchar(500)	YES		(Null)	
gender	varchar(10)	YES		(Null)	
birth_date	date	YES		(Null)	
phone_no	varchar(500)	YES		(Null)	
dept_id	int	YES	MUL	(Null)	
position_id	int	YES	MUL	(Null)	
supervisor_id	int	YES		(Null)	

After:

Field	Type	Null	Key	Default	Extra
emp_id	int	NO	PRI	(Null)	auto_increment
name	varchar(500)	YES		(Null)	
gender	varchar(10)	YES		(Null)	
birth_date	date	YES		(Null)	
phone_no	varchar(500)	YES		(Null)	
dept_id	int	YES	MUL	(Null)	
position_id	int	YES	MUL	(Null)	

RENAME TO

RENAME TO Commands:

```
ALTER TABLE edu_history RENAME TO education_history;
```

Excuted Result (with Query Commands):

```
DESC edu_history;
```

Before:

Field	Type	Null	Key	Default	Extra
edu_history_id	int	NO	PRI	(Null)	auto_increment
emp_files_id	int	YES	MUL	(Null)	
school_name	varchar(50)	YES		(Null)	
major	varchar(50)	YES		(Null)	
LEVEL	varchar(50)	YES		(Null)	
graduate_date	date	YES		(Null)	

After:

```
1146 - Table 'test_report.edu_history' doesn't exist
```

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)

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	Affected rows: 0	0.12s
--	------------------	-------

```
SELECT * FROM information_schema.TABLE_CONSTRAINTS
```

```
WHERE CONSTRAINT_SCHEMA = 'human_resource_management_system' AND  
TABLE_NAME = 'attendance_record' AND CONSTRAINT_TYPE = 'FOREIGN KEY';
```

CONSTRAINT_CATALOG	CONSTRAINT_SCHEMA	CONSTRAINT_NAME	TABLE_SCHEMA	TABLE_NAME	CONSTRAINT_TYPE	ENFORCED
def	human_resource_management_system	attendance_record_ibfk_1	human_resource_management_system	attendance_record	FOREIGN KEY	YES

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)

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	Affected rows: 0	0.037s
---	------------------	--------

```
SELECT * FROM information_schema.TABLE_CONSTRAINTS
```

```
WHERE CONSTRAINT_SCHEMA = 'human_resource_management_system' AND
TABLE_NAME = 'attendance_record' AND CONSTRAINT_TYPE = 'FOREIGN KEY';
```

CONSTRAINT_CATALOG	CONSTRAINT_SCHEMA	CONSTRAINT_NAME	TABLE_SCHEMA	TABLE_NAME	CONSTRAINT_TYPE	ENFORCED
def	human_resource_management_system	attendance_record_ibfk_1	human_resource_management_system	attendance_record	FOREIGN KEY	YES
► def	human_resource_management_system	FK_Attendance_Record_OT_Record_id	human_resource_management_system	attendance_record	FOREIGN KEY	YES

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)

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	Affected rows: 0	0.038s
--	------------------	--------

```
SELECT * FROM information_schema.TABLE_CONSTRAINTS
```

```
WHERE CONSTRAINT_SCHEMA = 'human_resource_management_system' AND
TABLE_NAME = 'candidates' AND CONSTRAINT_TYPE = 'FOREIGN KEY';
```

CONSTRAINT_CATALOG	CONSTRAINT_SCHEMA	CONSTRAINT_NAME	TABLE_SCHEMA	TABLE_NAME	CONSTRAINT_TYPE	ENFORCED
def	human_resource_management_system	FK_CANDIDATES_POSITION_OPENING_ID	human_resource_management_system	candidates	FOREIGN KEY	YES

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)

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	Affected rows: 0	0.035s
---	------------------	--------

```
SELECT * FROM information_schema.TABLE_CONSTRAINTS WHERE  
CONSTRAINT_SCHEMA = 'human_resource_management_system' AND TABLE_NAME  
= 'contracts' AND CONSTRAINT_TYPE = 'FOREIGN KEY';
```

CONSTRAINT_CATALOG	CONSTRAINT_SCHEMA	CONSTRAINT_NAME	TABLE_SCHEMA	TABLE_NAME	CONSTRAINT_TYPE	ENFORCED
def	human_resource_management_system	FK_contracts.emp_file_id	human_resource_management_system	contracts	FOREIGN KEY	YES

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)

ALTER TABLE `departments` ADD CONSTRAINT `FK_departments_dept_manager` FOREIGN KEY (`dept_manager`) REFERENCES `employees` (`emp_id`) ON DELETE RESTRICT ON UPDATE RESTRICT	Affected rows: 0	0.034s
---	------------------	--------

```
SELECT * FROM information_schema.TABLE_CONSTRAINTS WHERE  
CONSTRAINT_SCHEMA = 'human_resource_management_system' AND TABLE_NAME  
= 'departments' AND CONSTRAINT_TYPE = 'FOREIGN KEY';
```

CONSTRAINT_CATALOG	CONSTRAINT_SCHEMA	CONSTRAINT_NAME	TABLE_SCHEMA	TABLE_NAME	CONSTRAINT_TYPE	ENFORCED
► def	human_resource_management_system	FK_departments_dept_manager	human_resource_management_system	departments	FOREIGN KEY	YES

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)

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	Affected rows: 0	0.066s
--	------------------	--------

```
SELECT * FROM information_schema.TABLE_CONSTRAINTS WHERE  
CONSTRAINT_SCHEMA = 'human_resource_management_system' AND TABLE_NAME  
= 'edu_history' AND CONSTRAINT_TYPE = 'FOREIGN KEY';
```

CONSTRAINT_CATALOG	CONSTRAINT_SCHEMA	CONSTRAINT_NAME	TABLE_SCHEMA	TABLE_NAME	CONSTRAINT_TYPE	ENFORCED
def	human_resource_management_system	FK_Edu_History_emp_files_id	human_resource_management_system	edu_history	FOREIGN KEY	YES

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)

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	Affected rows: 0	0.037s
--	------------------	--------

```
SELECT * FROM information_schema.TABLE_CONSTRAINTS WHERE  
CONSTRAINT_SCHEMA = 'human_resource_management_system' AND TABLE_NAME  
= 'emp_penalites' AND CONSTRAINT_TYPE = 'FOREIGN KEY';
```

CONSTRAINT_CATALOG	CONSTRAINT_SCHEMA	CONSTRAINT_NAME	TABLE_SCHEMA	TABLE_NAME	CONSTRAINT_TYPE	ENFORCED
def	human_resource_management_system	FK_Emp_Penalites_emp_files_id	human_resource_management_system	emp_penalites	FOREIGN KEY	YES

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)

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	Affected rows: 0	0.039s
--	------------------	--------

```
SELECT * FROM information_schema.TABLE_CONSTRAINTS WHERE
CONSTRAINT_SCHEMA = 'human_resource_management_system' AND TABLE_NAME
= 'employee_attendance' AND CONSTRAINT_TYPE = 'FOREIGN KEY';
```

CONSTRAINT_CATALOG	CONSTRAINT_SCHEMA	CONSTRAINT_NAME	TABLE_SCHEMA	TABLE_NAME	CONSTRAINT_TYPE	ENFORCED
def	human_resource_management_system	employee_attendance_ibfk_1	human_resource_management_system	employee_attendance	FOREIGN KEY	YES

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)

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	Affected rows: 0	0.038s
--	------------------	--------

```
SELECT * FROM information_schema.TABLE_CONSTRAINTS WHERE
CONSTRAINT_SCHEMA = 'human_resource_management_system' AND TABLE_NAME
= 'employee_attendance' AND CONSTRAINT_TYPE = 'FOREIGN KEY';
```

CONSTRAINT_CATALOG	CONSTRAINT_SCHEMA	CONSTRAINT_NAME	TABLE_SCHEMA	TABLE_NAME	CONSTRAINT_TYPE	ENFORCED
def	human_resource_management_system	employee_attendance_ibfk_1	human_resource_management_system	employee_attendance	FOREIGN KEY	YES
def	human_resource_management_system	FK_employee_Attendance_Leave_Record_id	human_resource_management_system	employee_attendance	FOREIGN KEY	YES

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)

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	Affected rows: 0	0.041s
---	------------------	--------

```
SELECT * FROM information_schema.TABLE_CONSTRAINTS WHERE
CONSTRAINT_SCHEMA = 'human_resource_management_system' AND TABLE_NAME
= 'employee_attendance' AND CONSTRAINT_TYPE = 'FOREIGN KEY';
```

CONSTRAINT_CATALOG	CONSTRAINT_SCHEMA	CONSTRAINT_NAME	TABLE_SCHEMA	TABLE_NAME	CONSTRAINT_TYPE	ENFORCED
def	human_resource_management_system	employee_attendance_ibfk_1	human_resource_management_system	employee_attendance	FOREIGN KEY	YES
def	human_resource_management_system	FK_employee_Attendance_Leave_Record_id	human_resource_management_system	employee_attendance	FOREIGN KEY	YES
► def	human_resource_management_system	FK_employee_Attendance_Leave_req_id	human_resource_management_system	employee_attendance	FOREIGN KEY	YES

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)

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	Affected rows: 0	0.045s
---	------------------	--------

```
SELECT * FROM information_schema.TABLE_CONSTRAINTS WHERE
CONSTRAINT_SCHEMA = 'human_resource_management_system' AND TABLE_NAME
= 'employee_attendance' AND CONSTRAINT_TYPE = 'FOREIGN KEY';
```

CONSTRAINT_CATALOG	CONSTRAINT_SCHEMA	CONSTRAINT_NAME	TABLE_SCHEMA	TABLE_NAME	CONSTRAINT_TYPE	ENFORCED
def	human_resource_management_system	employee_attendance_ibfk_1	human_resource_management_system	employee_attendance	FOREIGN KEY	YES
def	human_resource_management_system	FK_employee_Attendance_Leave_Record_id	human_resource_management_system	employee_attendance	FOREIGN KEY	YES
def	human_resource_management_system	FK_employee_Attendance_Leave_req_id	human_resource_management_system	employee_attendance	FOREIGN KEY	YES
def	human_resource_management_system	FK_employee_attendance_OT_record_id	human_resource_management_system	employee_attendance	FOREIGN KEY	YES

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)

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	Affected rows: 0	0.028s
--	------------------	--------

```
SELECT * FROM information_schema.TABLE_CONSTRAINTS WHERE  
CONSTRAINT_SCHEMA = 'human_resource_management_system' AND TABLE_NAME  
= 'employee_awards' AND CONSTRAINT_TYPE = 'FOREIGN KEY';
```

CONSTRAINT_CATALOG	CONSTRAINT_SCHEMA	CONSTRAINT_NAME	TABLE_SCHEMA	TABLE_NAME	CONSTRAINT_TYPE	ENFORCED
def	human_resource_management_system	employee_awards_ibfk_1	human_resource_management_system	employee_awards	FOREIGN KEY	YES

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)

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	Affected rows: 0	0.031s
--	------------------	--------

```
SELECT * FROM information_schema.TABLE_CONSTRAINTS WHERE  
CONSTRAINT_SCHEMA = 'human_resource_management_system' AND TABLE_NAME  
= 'employee_benefits' AND CONSTRAINT_TYPE = 'FOREIGN KEY';
```

CONSTRAINT_CATALOG	CONSTRAINT_SCHEMA	CONSTRAINT_NAME	TABLE_SCHEMA	TABLE_NAME	CONSTRAINT_TYPE	ENFORCED
def	human_resource_management_system	employee_benefits_ibfk_1	human_resource_management_system	employee_benefits	FOREIGN KEY	YES

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)

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	Affected rows: 0	0.036s
--	------------------	--------

```
SELECT * FROM information_schema.TABLE_CONSTRAINTS WHERE  
CONSTRAINT_SCHEMA = 'human_resource_management_system' AND TABLE_NAME  
= 'employee_files' AND CONSTRAINT_TYPE = 'FOREIGN KEY';
```

CONSTRAINT_CATALOG	CONSTRAINT_SCHEMA	CONSTRAINT_NAME	TABLE_SCHEMA	TABLE_NAME	CONSTRAINT_TYPE	ENFORCED
def	human_resource_management_system	employee_files_ibfk_1	human_resource_management_system	employee_files	FOREIGN KEY	YES

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)

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	Affected rows: 0	0.028s
---	------------------	--------

```
SELECT * FROM information_schema.TABLE_CONSTRAINTS WHERE
CONSTRAINT_SCHEMA = 'human_resource_management_system' AND TABLE_NAME
= 'employee_health_checkups' AND CONSTRAINT_TYPE = 'FOREIGN KEY';
```

CONSTRAINT_CATALOG	CONSTRAINT_SCHEMA	CONSTRAINT_NAME	TABLE_SCHEMA	TABLE_NAME	CONSTRAINT_TYPE	ENFORCED
def	human_resource_management_system	FK_Employee_Health_Checkups.emp_files_id	human_resource_management_system	employee_health_checkups	FOREIGN KEY	YES

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)

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	Affected rows: 0	0.029s
--	------------------	--------

```
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';
```

CONSTRAINT_CATALOG	CONSTRAINT_SCHEMA	CONSTRAINT_NAME	TABLE_SCHEMA	TABLE_NAME	CONSTRAINT_TYPE	ENFORCED
def	human_resource_management_system	employee_performance_and_development_ibfk_1	human_resource_management_system	employee_performance_and_development	FOREIGN KEY	YES

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)

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	Affected rows: 0	0.042s
---	------------------	--------

```
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';
```

CONSTRAINT_CATALOG	CONSTRAINT_SCHEMA	CONSTRAINT_NAME	TABLE_SCHEMA	TABLE_NAME	CONSTRAINT_TYPE	ENFORCED
def	human_resource_management_system	employee_performance_and_development_ibfk_1	human_resource_management_system	employee_performance_and_development	FOREIGN KEY	YES
def	human_resource_management_system	employee_performance_and_development_ibfk_2	human_resource_management_system	employee_performance_and_development	FOREIGN KEY	YES

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)

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	Affected rows: 0	0.028s
---	------------------	--------

```
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';
```

CONSTRAINT_CATALOG	CONSTRAINT_SCHEMA	CONSTRAINT_NAME	TABLE_SCHEMA	TABLE_NAME	CONSTRAINT_TYPE	ENFORCED
def	human_resource_management_system	employee_performance_and_development_ibfk_1	human_resource_management_system	employee_performance_and	FOREIGN KEY	YES
def	human_resource_management_system	employee_performance_and_development_ibfk_2	human_resource_management_system	employee_performance_and	FOREIGN KEY	YES
▶ def	human_resource_management_system	employee_performance_and_development_ibfk_3	human_resource_management_system	employee_performance_and	FOREIGN KEY	YES

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)

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	Affected rows: 0	0.047s
--	------------------	--------

```
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';
```

CONSTRAINT_CATALOG	CONSTRAINT_SCHEMA	CONSTRAINT_NAME	TABLE_SCHEMA	TABLE_NAME	CONSTRAINT_TYPE	ENFORCED
def	human_resource_management_system	employee_performance_and_development_ibfk_1	human_resource_management_system	employee_performance_and	FOREIGN KEY	YES
def	human_resource_management_system	employee_performance_and_development_ibfk_2	human_resource_management_system	employee_performance_and	FOREIGN KEY	YES
def	human_resource_management_system	employee_performance_and_development_ibfk_3	human_resource_management_system	employee_performance_and	FOREIGN KEY	YES
def	human_resource_management_system	employee_performance_and_development_ibfk_4	human_resource_management_system	employee_performance_and	FOREIGN KEY	YES

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)

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	Affected rows: 0	0.037s
--	------------------	--------

```
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';
```

CONSTRAINT_CATALOG	CONSTRAINT_SCHEMA	CONSTRAINT_NAME	TABLE_SCHEMA	TABLE_NAME	CONSTRAINT_TYPE	ENFORCED
def	human_resource_management_system	employee_performance_and_development_ibfk_1	human_resource_management_system	employee_performance_and_development	FOREIGN KEY	YES
def	human_resource_management_system	employee_performance_and_development_ibfk_2	human_resource_management_system	employee_performance_and_development	FOREIGN KEY	YES
def	human_resource_management_system	employee_performance_and_development_ibfk_3	human_resource_management_system	employee_performance_and_development	FOREIGN KEY	YES
def	human_resource_management_system	employee_performance_and_development_ibfk_4	human_resource_management_system	employee_performance_and_development	FOREIGN KEY	YES
def	human_resource_management_system	employee_performance_and_development_ibfk_5	human_resource_management_system	employee_performance_and_development	FOREIGN KEY	YES

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)

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	Affected rows: 0	0.032s
--	------------------	--------

```
SELECT * FROM information_schema.TABLE_CONSTRAINTS WHERE  
CONSTRAINT_SCHEMA = 'human_resource_management_system' AND TABLE_NAME  
= 'employee_promotion' AND CONSTRAINT_TYPE = 'FOREIGN KEY';
```

CONSTRAINT_CATALOG	CONSTRAINT_SCHEMA	CONSTRAINT_NAME	TABLE_SCHEMA	TABLE_NAME	CONSTRAINT_TYPE	ENFORCED
def	human_resource_management_system	employee_promotion_ibfk_1	human_resource_management_system	employee_promotion	FOREIGN KEY	YES

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)

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	Affected rows: 0	0.032s
---	------------------	--------

```
SELECT * FROM information_schema.TABLE_CONSTRAINTS WHERE  
CONSTRAINT_SCHEMA = 'human_resource_management_system' AND TABLE_NAME  
= 'employee_promotion' AND CONSTRAINT_TYPE = 'FOREIGN KEY';
```

CONSTRAINT_CATALOG	CONSTRAINT_SCHEMA	CONSTRAINT_NAME	TABLE_SCHEMA	TABLE_NAME	CONSTRAINT_TYPE	ENFORCED
def	human_resource_management_system	employee_promotion_ibfk_1	human_resource_management_system	employee_promotion	FOREIGN KEY	YES
def	human_resource_management_system	employee_promotion_ibfk_2	human_resource_management_system	employee_promotion	FOREIGN KEY	YES

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)

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	Affected rows: 0	0.042s
---	------------------	--------

```
SELECT * FROM information_schema.TABLE_CONSTRAINTS WHERE
CONSTRAINT_SCHEMA = 'human_resource_management_system' AND TABLE_NAME
= 'employee_promotion' AND CONSTRAINT_TYPE = 'FOREIGN KEY';
```

CONSTRAINT_CATALOG	CONSTRAINT_SCHEMA	CONSTRAINT_NAME	TABLE_SCHEMA	TABLE_NAME	CONSTRAINT_TYPE	ENFORCED
def	human_resource_management_system	employee_promotion_ibfk_1	human_resource_management_system	employee_promotion	FOREIGN KEY	YES
def	human_resource_management_system	employee_promotion_ibfk_2	human_resource_management_system	employee_promotion	FOREIGN KEY	YES
def	human_resource_management_system	FK_employee_promotion_emp_promo_id	human_resource_management_system	employee_promotion	FOREIGN KEY	YES

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)

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	Affected rows: 0	0.062s
--	------------------	--------

```
SELECT * FROM information_schema.TABLE_CONSTRAINTS WHERE  
CONSTRAINT_SCHEMA = 'human_resource_management_system' AND TABLE_NAME  
= 'employee_training' AND CONSTRAINT_TYPE = 'FOREIGN KEY';
```

CONSTRAINT_CATALOG	CONSTRAINT_SCHEMA	CONSTRAINT_NAME	TABLE_SCHEMA	TABLE_NAME	CONSTRAINT_TYPE	ENFORCED
def	human_resource_management_system	employee_training_ibfk_1	human_resource_management_system	employee_training	FOREIGN KEY	YES

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)

ALTER TABLE `employees` ADD CONSTRAINT 'FK_EMPLOYEES_DEPT_ID' FOREIGN KEY ('dept_id') REFERENCES `departments` ('dept_id') ON DELETE RESTRICT ON UPDATE RESTRICT	0.035s
--	--------

```
SELECT * FROM information_schema.TABLE_CONSTRAINTS WHERE  
CONSTRAINT_SCHEMA = 'human_resource_management_system' AND TABLE_NAME  
= 'employees' AND CONSTRAINT_TYPE = 'FOREIGN KEY';
```

CONSTRAINT_CATALOG	CONSTRAINT_SCHEMA	CONSTRAINT_NAME	TABLE_SCHEMA	TABLE_NAME	CONSTRAINT_TYPE	ENFORCED
def	human_resource_management_system	FK_EMPLOYEES_DEPT_ID	human_resource_management_system	employees	FOREIGN KEY	YES

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)

ALTER TABLE `employees` ADD CONSTRAINT `FK_EMPLOYEES_POSITION_ID` FOREIGN KEY (`position_id`) REFERENCES `positions` (`position_id`) ON DELETE RESTRICT ON UPDATE RESTRICT	Affected rows: 0	0.031s
--	------------------	--------

```
SELECT * FROM information_schema.TABLE_CONSTRAINTS WHERE  
CONSTRAINT_SCHEMA = 'human_resource_management_system' AND TABLE_NAME  
= 'employees' AND CONSTRAINT_TYPE = 'FOREIGN KEY';
```

CONSTRAINT_CATALOG	CONSTRAINT_SCHEMA	CONSTRAINT_NAME	TABLE_SCHEMA	TABLE_NAME	CONSTRAINT_TYPE	ENFORCED
def	human_resource_management_system	FK_EMPLOYEES_DEPT_ID	human_resource_management_system	employees	FOREIGN KEY	YES
▶ def	human_resource_management_system	FK_EMPLOYEES_POSITION_ID	human_resource_management_system	employees	FOREIGN KEY	YES

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)

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	Affected rows: 0	0.028s
--	------------------	--------

```
SELECT * FROM information_schema.TABLE_CONSTRAINTS WHERE  
CONSTRAINT_SCHEMA = 'human_resource_management_system' AND TABLE_NAME  
= 'employees_salary_details' AND CONSTRAINT_TYPE = 'FOREIGN KEY';
```

CONSTRAINT_CATALOG	CONSTRAINT_SCHEMA	CONSTRAINT_NAME	TABLE_SCHEMA	TABLE_NAME	CONSTRAINT_TYPE	ENFORCED
def	human_resource_management_system	FK_EMPLOYEES_SALARY_DETAILS_Employee_Benefits_id	human_resource_management_system	employees_salary_details	FOREIGN KEY	YES

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)

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	Affected rows: 0	0.023s
--	------------------	--------

```
SELECT * FROM information_schema.TABLE_CONSTRAINTS WHERE  
CONSTRAINT_SCHEMA = 'human_resource_management_system' AND TABLE_NAME  
= 'interview_list' AND CONSTRAINT_TYPE = 'FOREIGN KEY';
```

CONSTRAINT_CATALOG	CONSTRAINT_SCHEMA	CONSTRAINT_NAME	TABLE_SCHEMA	TABLE_NAME	CONSTRAINT_TYPE	ENFORCED
def	human_resource_management_system	FK_Interview_List_Candidate_id	human_resource_management_system	interview_list	FOREIGN KEY	YES

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)

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	Affected rows: 0	0.028s
---	------------------	--------

```
SELECT * FROM information_schema.TABLE_CONSTRAINTS WHERE  
CONSTRAINT_SCHEMA = 'human_resource_management_system' AND TABLE_NAME  
= 'interview_list' AND CONSTRAINT_TYPE = 'FOREIGN KEY';
```

CONSTRAINT_CATALOG	CONSTRAINT_SCHEMA	CONSTRAINT_NAME	TABLE_SCHEMA	TABLE_NAME	CONSTRAINT_TYPE	ENFORCED
def	human_resource_management_system	FK_Interview_List_Candidate_id	human_resource_management_system	interview_list	FOREIGN KEY	YES
▶ def	human_resource_management_system	FK_Interview_List_Interviewer_id	human_resource_management_system	interview_list	FOREIGN KEY	YES

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)

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	Affected rows: 0	0.027s
---	------------------	--------

```
SELECT * FROM information_schema.TABLE_CONSTRAINTS WHERE
CONSTRAINT_SCHEMA = 'human_resource_management_system' AND TABLE_NAME
= 'leave_record' AND CONSTRAINT_TYPE = 'FOREIGN KEY';
```

CONSTRAINT_CATALOG	CONSTRAINT_SCHEMA	CONSTRAINT_NAME	TABLE_SCHEMA	TABLE_NAME	CONSTRAINT_TYPE	ENFORCED
def	human_resource_management_system	FK_Leave_Record_leave_approver_id	human_resource_management_system	leave_record	FOREIGN KEY	YES

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)

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	Affected rows: 0	0.027s
---	------------------	--------

```
SELECT * FROM information_schema.TABLE_CONSTRAINTS WHERE  
CONSTRAINT_SCHEMA = 'human_resource_management_system' AND TABLE_NAME  
= 'leave_record' AND CONSTRAINT_TYPE = 'FOREIGN KEY';
```

CONSTRAINT_CATALOG	CONSTRAINT_SCHEMA	CONSTRAINT_NAME	TABLE_SCHEMA	TABLE_NAME	CONSTRAINT_TYPE	ENFORCED
def	human_resource_management_system	FK_Leave_Record_leave_approver_id	human_resource_management_system	leave_record	FOREIGN KEY	YES
► def	human_resource_management_system	FK_Leave_Record_leave_req_id	human_resource_management_system	leave_record	FOREIGN KEY	YES

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)

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	Affected rows: 0	0.03s
---	------------------	-------

```
SELECT * FROM information_schema.TABLE_CONSTRAINTS WHERE  
CONSTRAINT_SCHEMA = 'human_resource_management_system' AND TABLE_NAME  
= 'leave_request' AND CONSTRAINT_TYPE = 'FOREIGN KEY';
```

CONSTRAINT_CATALOG	CONSTRAINT_SCHEMA	CONSTRAINT_NAME	TABLE_SCHEMA	TABLE_NAME	CONSTRAINT_TYPE	ENFORCED
► def	human_resource_management_system	FK_Leave_Request_requester_id	human_resource_management_system	leave_request	FOREIGN KEY	YES

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)

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	Affected rows: 0	0.027s
--	------------------	--------

```
SELECT * FROM information_schema.TABLE_CONSTRAINTS WHERE
CONSTRAINT_SCHEMA = 'human_resource_management_system' AND TABLE_NAME
= 'position_application' AND CONSTRAINT_TYPE = 'FOREIGN KEY';
```

CONSTRAINT_CATALOG	CONSTRAINT_SCHEMA	CONSTRAINT_NAME	TABLE_SCHEMA	TABLE_NAME	CONSTRAINT_TYPE	ENFORCED
def	human_resource_management_system	fk_position_application_current_position_id	human_resource_management_system	position_application	FOREIGN KEY	YES

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)

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	Affected rows: 0	0.052s
--	------------------	--------

```
SELECT * FROM information_schema.TABLE_CONSTRAINTS WHERE
CONSTRAINT_SCHEMA = 'human_resource_management_system' AND TABLE_NAME
= 'position_application' AND CONSTRAINT_TYPE = 'FOREIGN KEY';
```

CONSTRAINT_CATALOG	CONSTRAINT_SCHEMA	CONSTRAINT_NAME	TABLE_SCHEMA	TABLE_NAME	CONSTRAINT_TYPE	ENFORCED
def	human_resource_management_system	fk_position_application_current_position_id	human_resource_management_system	position_application	FOREIGN KEY	YES
def	human_resource_management_system	fk_position_application_position_opening_id	human_resource_management_system	position_application	FOREIGN KEY	YES

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)

ALTER TABLE `position_application` ADD CONSTRAINT `fk_position_skillset_id` FOREIGN KEY (`skillset_id`) REFERENCES `skillset` (`Skillset_id`) ON DELETE RESTRICT ON UPDATE RESTRICT	Affected rows: 0	0.034s
---	------------------	--------

```
SELECT * FROM information_schema.TABLE_CONSTRAINTS WHERE  
CONSTRAINT_SCHEMA = 'human_resource_management_system' AND TABLE_NAME  
= 'position_application' AND CONSTRAINT_TYPE = 'FOREIGN KEY';
```

CONSTRAINT_CATALOG	CONSTRAINT_SCHEMA	CONSTRAINT_NAME	TABLE_SCHEMA	TABLE_NAME	CONSTRAINT_TYPE	ENFORCED
def	human_resource_management_system	fk_position_application_current_position_id	human_resource_management_system	position_application	FOREIGN KEY	YES
def	human_resource_management_system	fk_position_application_position_opening_id	human_resource_management_system	position_application	FOREIGN KEY	YES
▶ def	human_resource_management_system	fk_position_skillset_id	human_resource_management_system	position_application	FOREIGN KEY	YES

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)

ALTER TABLE `position_application` ADD CONSTRAINT `position_application_ibfk_1` FOREIGN KEY (`emp_id`) REFERENCES `employees` (`emp_id`) ON DELETE RESTRICT ON UPDATE RESTRICT	Affected rows: 0	0.044s
--	------------------	--------

```
SELECT * FROM information_schema.TABLE_CONSTRAINTS WHERE  
CONSTRAINT_SCHEMA = 'human_resource_management_system' AND TABLE_NAME  
= 'position_application' AND CONSTRAINT_TYPE = 'FOREIGN KEY';
```

CONSTRAINT_CATALOG	CONSTRAINT_SCHEMA	CONSTRAINT_NAME	TABLE_SCHEMA	TABLE_NAME	CONSTRAINT_TYPE	ENFORCED
def	human_resource_management_system	fk_position_application_current_position_id	human_resource_management_system	position_application	FOREIGN KEY	YES
def	human_resource_management_system	fk_position_application_position_opening_id	human_resource_management_system	position_application	FOREIGN KEY	YES
def	human_resource_management_system	fk_position_skillset_id	human_resource_management_system	position_application	FOREIGN KEY	YES
def	human_resource_management_system	position_application_ibfk_1	human_resource_management_system	position_application	FOREIGN KEY	YES

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)

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	Affected rows: 0	0.031s
--	------------------	--------

```
SELECT * FROM information_schema.TABLE_CONSTRAINTS WHERE
CONSTRAINT_SCHEMA = 'human_resource_management_system' AND TABLE_NAME
= 'position_opening' AND CONSTRAINT_TYPE = 'FOREIGN KEY';
```

CONSTRAINT_CATALOG	CONSTRAINT_SCHEMA	CONSTRAINT_NAME	TABLE_SCHEMA	TABLE_NAME	CONSTRAINT_TYPE	ENFORCED
def	human_resource_management_system	FK_position_Opening_Position_id	human_resource_management_system	position_opening	FOREIGN KEY	YES

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)

ALTER TABLE `positions` ADD CONSTRAINT `FK_POSITIONS_DEPT_ID` FOREIGN KEY (`dept_id`) REFERENCES `departments` (`dept_id`) ON DELETE RESTRICT ON UPDATE RESTRICT	Affected rows: 0	0.041s
--	------------------	--------

```
SELECT * FROM information_schema.TABLE_CONSTRAINTS WHERE  
CONSTRAINT_SCHEMA = 'human_resource_management_system' AND TABLE_NAME  
= 'positions' AND CONSTRAINT_TYPE = 'FOREIGN KEY';
```

CONSTRAINT_CATALOG	CONSTRAINT_SCHEMA	CONSTRAINT_NAME	TABLE_SCHEMA	TABLE_NAME	CONSTRAINT_TYPE	ENFORCED
► def	human_resource_management_system	FK_POSITIONS_DEPT_ID	human_resource_management_system	positions	FOREIGN KEY	YES

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)

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	Affected rows: 0	0.057s
--	------------------	--------

```
SELECT * FROM information_schema.TABLE_CONSTRAINTS WHERE
CONSTRAINT_SCHEMA = 'human_resource_management_system' AND TABLE_NAME
= 'project_participation' AND CONSTRAINT_TYPE = 'FOREIGN KEY';
```

CONSTRAINT_CATALOG	CONSTRAINT_SCHEMA	CONSTRAINT_NAME	TABLE_SCHEMA	TABLE_NAME	CONSTRAINT_TYPE	ENFORCED
► def	human_resource_management_system	FK_project_participation_project_id	human_resource_management_system	project_participation	FOREIGN KEY	YES

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)

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	Affected rows: 0	0.038s
---	------------------	--------

```
SELECT * FROM information_schema.TABLE_CONSTRAINTS WHERE  
CONSTRAINT_SCHEMA = 'human_resource_management_system' AND TABLE_NAME  
= 'project_participation' AND CONSTRAINT_TYPE = 'FOREIGN KEY';
```

CONSTRAINT_CATALOG	CONSTRAINT_SCHEMA	CONSTRAINT_NAME	TABLE_SCHEMA	TABLE_NAME	CONSTRAINT_TYPE	ENFORCED
def	human_resource_management_system	FK_project_participation_project_id	human_resource_management_system	project_participation	FOREIGN KEY	YES
def	human_resource_management_system	project_participation_ibfk_1	human_resource_management_system	project_participation	FOREIGN KEY	YES

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)

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	Affected rows: 0	0.03s
--	------------------	-------

```
SELECT * FROM information_schema.TABLE_CONSTRAINTS WHERE  
CONSTRAINT_SCHEMA = 'human_resource_management_system' AND TABLE_NAME  
= 'projects' AND CONSTRAINT_TYPE = 'FOREIGN KEY';
```

CONSTRAINT_CATALOG	CONSTRAINT_SCHEMA	CONSTRAINT_NAME	TABLE_SCHEMA	TABLE_NAME	CONSTRAINT_TYPE	ENFORCED
def	human_resource_management_system	projects_ibfk_1	human_resource_management_system	projects	FOREIGN KEY	YES

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)

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	Affected rows: 0	0.024s
--	------------------	--------

```
SELECT * FROM information_schema.TABLE_CONSTRAINTS WHERE  
CONSTRAINT_SCHEMA = 'human_resource_management_system' AND TABLE_NAME  
= 'resignations' AND CONSTRAINT_TYPE = 'FOREIGN KEY';
```

CONSTRAINT_CATALOG	CONSTRAINT_SCHEMA	CONSTRAINT_NAME	TABLE_SCHEMA	TABLE_NAME	CONSTRAINT_TYPE	ENFORCED
def	human_resource_management_system	resignations_ibfk_1	human_resource_management_system	resignations	FOREIGN KEY	YES

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)

ALTER TABLE `resignations` ADD CONSTRAINT `resignations_ibfk_2` FOREIGN KEY (`related_contract_id`) REFERENCES `contracts` (`contract_id`) ON DELETE RESTRICT ON UPDATE RESTRICT	Affected rows: 0	0.028s
--	------------------	--------

```
SELECT * FROM information_schema.TABLE_CONSTRAINTS WHERE  
CONSTRAINT_SCHEMA = 'human_resource_management_system' AND TABLE_NAME  
= 'resignations' AND CONSTRAINT_TYPE = 'FOREIGN KEY';
```

CONSTRAINT_CATALOG	CONSTRAINT_SCHEMA	CONSTRAINT_NAME	TABLE_SCHEMA	TABLE_NAME	CONSTRAINT_TYPE	ENFORCED
def	human_resource_management_system	resignations_ibfk_1	human_resource_management_system	resignations	FOREIGN KEY	YES
► def	human_resource_management_system	resignations_ibfk_2	human_resource_management_system	resignations	FOREIGN KEY	YES

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)

ALTER TABLE `resignations` ADD CONSTRAINT `resignations_ibfk_3` FOREIGN KEY (`exit_interviewer_id`) REFERENCES `employees` (`emp_id`) ON DELETE RESTRICT ON UPDATE RESTRICT	Affected rows: 0	0.026s
---	------------------	--------

```
SELECT * FROM information_schema.TABLE_CONSTRAINTS WHERE  
CONSTRAINT_SCHEMA = 'human_resource_management_system' AND TABLE_NAME  
= 'resignations' AND CONSTRAINT_TYPE = 'FOREIGN KEY';
```

CONSTRAINT_CATALOG	CONSTRAINT_SCHEMA	CONSTRAINT_NAME	TABLE_SCHEMA	TABLE_NAME	CONSTRAINT_TYPE	ENFORCED
def	human_resource_management_system	resignations_ibfk_1	human_resource_management_system	resignations	FOREIGN KEY	YES
def	human_resource_management_system	resignations_ibfk_2	human_resource_management_system	resignations	FOREIGN KEY	YES
def	human_resource_management_system	resignations_ibfk_3	human_resource_management_system	resignations	FOREIGN KEY	YES

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)

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	Affected rows: 0	0.029s
---	------------------	--------

```
SELECT * FROM information_schema.TABLE_CONSTRAINTS WHERE
CONSTRAINT_SCHEMA = 'human_resource_management_system' AND TABLE_NAME
= 'salary_history' AND CONSTRAINT_TYPE = 'FOREIGN KEY';
```

CONSTRAINT_CATALOG	CONSTRAINT_SCHEMA	CONSTRAINT_NAME	TABLE_SCHEMA	TABLE_NAME	CONSTRAINT_TYPE	ENFORCED
def	human_resource_management_system	FK_SALARY_HISTORY_SALARY_DETAILS_Employee_Benefits_id	human_resource_management_system	salary_history	FOREIGN KEY	YES

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)

ALTER TABLE `skills` ADD CONSTRAINT `skills_ibfk_1` FOREIGN KEY (`dept_id`) REFERENCES `departments` (`dept_id`) ON DELETE RESTRICT ON UPDATE RESTRICT	Affected rows: 0	0.033s
--	------------------	--------

```
SELECT * FROM information_schema.TABLE_CONSTRAINTS WHERE
CONSTRAINT_SCHEMA = 'human_resource_management_system' AND TABLE_NAME
= 'skills' AND CONSTRAINT_TYPE = 'FOREIGN KEY';
```

CONSTRAINT_CATALOG	CONSTRAINT_SCHEMA	CONSTRAINT_NAME	TABLE_SCHEMA	TABLE_NAME	CONSTRAINT_TYPE	ENFORCED
def	human_resource_management_system	skills_ibfk_1	human_resource_management_system	skills	FOREIGN KEY	YES

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)

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	Affected rows: 0	0.038s
--	------------------	--------

```
SELECT * FROM information_schema.TABLE_CONSTRAINTS WHERE  
CONSTRAINT_SCHEMA = 'human_resource_management_system' AND TABLE_NAME  
= 'skillset' AND CONSTRAINT_TYPE = 'FOREIGN KEY';
```

CONSTRAINT_CATALOG	CONSTRAINT_SCHEMA	CONSTRAINT_NAME	TABLE_SCHEMA	TABLE_NAME	CONSTRAINT_TYPE	ENFORCED
def	human_resource_management_system	FK_Skillset_emp_pnd_id	human_resource_management_system	skillset	FOREIGN KEY	YES

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)

ALTER TABLE `skillset` ADD CONSTRAINT `skillset_ibfk_1` FOREIGN KEY (`candidate_id`) REFERENCES `candidates` (`Candidate_id`) ON DELETE RESTRICT ON UPDATE RESTRICT	Affected rows: 0	0.037s
---	------------------	--------

```
SELECT * FROM information_schema.TABLE_CONSTRAINTS WHERE  
CONSTRAINT_SCHEMA = 'human_resource_management_system' AND TABLE_NAME  
= 'skillset' AND CONSTRAINT_TYPE = 'FOREIGN KEY';
```

CONSTRAINT_CATALOG	CONSTRAINT_SCHEMA	CONSTRAINT_NAME	TABLE_SCHEMA	TABLE_NAME	CONSTRAINT_TYPE	ENFORCED
def	human_resource_management_system	FK_Skillset_emp_prd_id	human_resource_management_system	skillset	FOREIGN KEY	YES
► def	human_resource_management_system	skillset_ibfk_1	human_resource_management_system	skillset	FOREIGN KEY	YES

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)

ALTER TABLE `skillset` ADD CONSTRAINT `skillset_ibfk_2` FOREIGN KEY ('Skill_one_id') REFERENCES `skills` ('Skill_id') ON DELETE RESTRICT	Affected rows: 0	0.04s
--	------------------	-------

```
SELECT * FROM information_schema.TABLE_CONSTRAINTS WHERE  
CONSTRAINT_SCHEMA = 'human_resource_management_system' AND TABLE_NAME  
= 'skillset' AND CONSTRAINT_TYPE = 'FOREIGN KEY';
```

CONSTRAINT_CATALOG	CONSTRAINT_SCHEMA	CONSTRAINT_NAME	TABLE_SCHEMA	TABLE_NAME	CONSTRAINT_TYPE	ENFORCED
def	human_resource_management_system	FK_Skillset_Emp_Pnd_id	human_resource_management_system	skillset	FOREIGN KEY	YES
def	human_resource_management_system	skillset_ibfk_1	human_resource_management_system	skillset	FOREIGN KEY	YES
def	human_resource_management_system	skillset_ibfk_2	human_resource_management_system	skillset	FOREIGN KEY	YES

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)

```
ALTER TABLE `skillset` ADD CONSTRAINT `skillset_ibfk_3` FOREIGN KEY (`Skill_two_id`) REFERENCES `skills` (`Skill_id`) ON DELETE RESTRICT ON UPDATE RESTRICT | Affected rows: 0 | 0.034s
```

```
SELECT * FROM information_schema.TABLE_CONSTRAINTS WHERE  
CONSTRAINT_SCHEMA = 'human_resource_management_system' AND TABLE_NAME  
= 'skillset' AND CONSTRAINT_TYPE = 'FOREIGN KEY';
```

CONSTRAINT_CATALOG	CONSTRAINT_SCHEMA	CONSTRAINT_NAME	TABLE_SCHEMA	TABLE_NAME	CONSTRAINT_TYPE	ENFORCED
def	human_resource_management_system	FK_Skillset_Emp_Pnd_id	human_resource_management_system	skillset	FOREIGN KEY	YES
def	human_resource_management_system	skillset_ibfk_1	human_resource_management_system	skillset	FOREIGN KEY	YES
def	human_resource_management_system	skillset_ibfk_2	human_resource_management_system	skillset	FOREIGN KEY	YES
▶ def	human_resource_management_system	skillset_ibfk_3	human_resource_management_system	skillset	FOREIGN KEY	YES

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)

ALTER TABLE `skillset` ADD CONSTRAINT `skillset_ibfk_4` FOREIGN KEY (`Skill_three_id`) REFERENCES `skills` (`Skill_id`) ON DELETE RESTRICT ON UPDATE	Affected rows: 0	0.041s
---	------------------	--------

```
SELECT * FROM information_schema.TABLE_CONSTRAINTS WHERE  
CONSTRAINT_SCHEMA = 'human_resource_management_system' AND TABLE_NAME  
= 'skillset' AND CONSTRAINT_TYPE = 'FOREIGN KEY';
```

CONSTRAINT_CATALOG	CONSTRAINT_SCHEMA	CONSTRAINT_NAME	TABLE_SCHEMA	TABLE_NAME	CONSTRAINT_TYPE	ENFORCED
def	human_resource_management_system	FK_Skillset_emp_prd_id	human_resource_management_system	skillset	FOREIGN KEY	YES
def	human_resource_management_system	skillset_ibfk_1	human_resource_management_system	skillset	FOREIGN KEY	YES
def	human_resource_management_system	skillset_ibfk_2	human_resource_management_system	skillset	FOREIGN KEY	YES
def	human_resource_management_system	skillset_ibfk_3	human_resource_management_system	skillset	FOREIGN KEY	YES
def	human_resource_management_system	skillset_ibfk_4	human_resource_management_system	skillset	FOREIGN KEY	YES

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)

ALTER TABLE `users` ADD CONSTRAINT `users_ibfk_1` FOREIGN KEY (`emp_id`) REFERENCES `employees` (`emp_id`) ON DELETE RESTRICT ON UPDATE RESTRICT	Affected rows: 0	0.026s
--	------------------	--------

```
SELECT * FROM information_schema.TABLE_CONSTRAINTS
```

```
WHERE CONSTRAINT_SCHEMA = 'human_resource_management_system' AND
TABLE_NAME = 'users' AND CONSTRAINT_TYPE = 'FOREIGN KEY';
```

CONSTRAINT_CATALOG	CONSTRAINT_SCHEMA	CONSTRAINT_NAME	TABLE_SCHEMA	TABLE_NAME	CONSTRAINT_TYPE	ENFORCED
def	human_resource_management_system	users_ibfk_1	human_resource_management_system	users	FOREIGN KEY	YES

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)

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	0.023s
---	--------

```
SELECT * FROM information_schema.TABLE_CONSTRAINTS WHERE  
CONSTRAINT_SCHEMA = 'human_resource_management_system' AND TABLE_NAME  
= 'work_exp' AND CONSTRAINT_TYPE = 'FOREIGN KEY';
```

CONSTRAINT_CATALOG	CONSTRAINT_SCHEMA	CONSTRAINT_NAME	TABLE_SCHEMA	TABLE_NAME	CONSTRAINT_TYPE	ENFORCED
def	human_resource_management_system	FK_Work_Exp_emp_files_id	human_resource_management_system	work_exp	FOREIGN KEY	YES

DROP CONSTRAINT

DROP CONSTRAINT Commands:

```
ALTER TABLE users DROP CONSTRAINT users_ibfk_1;
```

Excuted Result (with Query Commands):

(Excute successfully)

ALTER TABLE users DROP CONSTRAINT users_ibfk_1	OK	0.114s
--	----	--------

```
SELECT * FROM information_schema.TABLE_CONSTRAINTS
```

```
WHERE CONSTRAINT_SCHEMA = 'human_resource_management_system' AND  
TABLE_NAME = 'users' AND CONSTRAINT_TYPE = 'FOREIGN KEY';
```

Before:

CONSTRAINT_CATALOG	CONSTRAINT_SCHEMA	CONSTRAINT_NAME	TABLE_SCHEMA	TABLE_NAME	CONSTRAINT_TYPE	ENFORCED
def	human_resource_management_system	users_ibfk_1	human_resource_management_system	users	FOREIGN KEY	YES

After:

CONSTRAINT_CATALOG	CONSTRAINT_SCHEMA	CONSTRAINT_NAME	TABLE_SCHEMA	TABLE_NAME	CONSTRAINT_TYPE	ENFORCED
(N/A)	(N/A)	(N/A)	(N/A)	(N/A)	(N/A)	(N/A)

2.2.3 RENAME

RENAME TABLE

Commands:

```
RENAME TABLE edu_history TO education_history;
```

Excuted Result (with Query Commands):

```
DESC edu_history;
```

Before:

Field	Type	Null	Key	Default	Extra
edu_history_id	int	NO	PRI	(Null)	auto_increment
emp_files_id	int	YES	MUL	(Null)	
school_name	varchar(50)	YES		(Null)	
major	varchar(50)	YES		(Null)	
LEVEL	varchar(50)	YES		(Null)	
graduate_date	date	YES		(Null)	

After:

```
1146 - Table 'test_report.edu_history' doesn't exist
```

RENAME COLUMN

Commands:

```
ALTER TABLE employees RENAME COLUMN emp_id TO employee_id;
```

Excuted Result (with Query Commands):

```
DESC employees;
```

Before:

Field	Type	Null	Key	Default	Extra
emp_id	int	NO	PRI	(Null)	auto_increment
name	varchar(500)	YES		(Null)	
gender	varchar(10)	YES		(Null)	
birth_date	date	YES		(Null)	
phone_no	varchar(500)	YES		(Null)	
dept_id	int	YES	MUL	(Null)	
position_id	int	YES	MUL	(Null)	

After:

Field	Type	Null	Key	Default	Extra
employee_id	int	NO	PRI	(Null)	auto_increment
name	varchar(500)	YES		(Null)	
gender	varchar(10)	YES		(Null)	
birth_date	date	YES		(Null)	
phone_no	varchar(500)	YES		(Null)	
dept_id	int	YES	MUL	(Null)	
position_id	int	YES	MUL	(Null)	

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)

INSERT INTO positions (position_name, avg_salary, dept_id, emp_count) VALUES -- position for Department of sale ('Sales Manager', 70000.00, NULL, 1), ('Sales Representative', 45000.00, NULL, 8),	Affected rows: 20	0.003s
---	-------------------	--------

SELECT * FROM positions;

Before:

position_id	position_name	avg_salary	dept_id	emp_count
	(N/A)	(N/A)	(N/A)	(N/A)

After:

position_id	position_name	avg_salary	dept_id	emp_count
1	Sales Manager	70000.00	(Null)	1
2	Sales Representative	45000.00	(Null)	8
3	Sales Associate	38000.00	(Null)	10
4	Sales Support Specialis	42000.00	(Null)	4
5	IT Manager	65000.00	(Null)	1
6	Software Engineer	58000.00	(Null)	7
7	Systems Analyst	52000.00	(Null)	5
8	IT Support Specialist	42000.00	(Null)	6
9	HR Manager	60000.00	(Null)	1
10	HR Assistant	45000.00	(Null)	3
11	HR Specialist	48000.00	(Null)	5
12	HR System Administrat	40000.00	(Null)	4
13	Marketing Manager	60000.00	(Null)	5
14	Marketing Specialist	45000.00	(Null)	8
15	Social Media Manager	50000.00	(Null)	3
16	Marketing Analyst	55000.00	(Null)	6
17	Finance Manager	62000.00	(Null)	1
18	Financial Analyst	55000.00	(Null)	3
19	Accountant	48000.00	(Null)	6
20	Auditor	53000.00	(Null)	2

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)

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),	Affected rows: 5	0.001s
--	------------------	--------

```
SELECT * FROM departments;
```

Before:

dept_id	dept_name	dept_manager	location	emp_count
	(N/A)	(N/A)	(N/A)	(N/A)

After:

dept_id	dept_name	dept_manager	location	emp_count
1	Department of sale	(Null)	floor1	10
2	Department of information technology	(Null)	floor2	9
3	Department of Human Resources	(Null)	floor3	5
4	Department of Marketing	(Null)	floor4	5
5	Department of Finance	(Null)	floor5	4

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)

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),	Affected rows: 20	0.007s
--	-------------------	--------

SELECT * FROM employees;

Before:

emp_id	name	gender	birth_date	phone_no	dept_id	position_id
	(N/A)	(N/A)	(N/A)	(N/A)	(N/A)	(N/A)

After:

emp_id	name	gender	birth_date	phone_no	dept_id	position_id
1	John Doe	Male	1990-05-15	011-12345678	3	9
2	Jane Smith	Female	1985-08-22	012-98765432	1	1
3	Michael Johnson	Male	1988-11-30	013-55555555	2	5
4	Emily Brown	Female	1992-03-25	014-77777777	4	13
5	David Lee	Male	1987-09-18	015-88888888	5	17
6	Sarah Wang	Female	1991-07-10	016-99999999	1	3
7	Robert Chen	Male	1986-12-05	017-66666666	2	7
8	Jessica Liu	Female	1989-04-12	018-11111111	3	12
9	William Zhang	Male	1993-02-08	019-22222222	4	16
10	Linda Wu	Female	1994-01-01	010-44444444	5	20
11	Alex Johnson	Male	1992-08-11	011-12345678	1	2
12	Ella Williams	Female	1990-05-22	011-98765432	2	6
13	Noah Smith	Male	1988-11-19	011-56789012	1	3
14	Ava Davis	Female	1995-03-25	011-24681357	3	10
15	Liam Taylor	Male	1994-09-18	011-11223344	2	7
16	Mia Brown	Female	1991-07-11	011-55443322	1	4
17	Jameson Lee	Male	1989-12-29	011-98761234	3	11
18	Aria Wilson	Female	1993-04-06	011-66778899	2	8
19	Ethan Miller	Male	1996-06-30	011-13579135	1	2
20	Scarlett Martin	Female	1987-02-14	011-11235813	3	12

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)

-- 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')	Affected rows: 10	0.008s
---	-------------------	--------

SELECT * FROM employee_files;

Before:

emp_files_id	file_name	file_size	file_path	upload_date
	(N/A)	(N/A)	(N/A)	(N/A)

After:

emp_files_id	file_name	file_size	file_path	upload_date
1	Resume_John_Doe.pdf	2.3 MB	/uploads/resumes/	2023-05-10
2	Profile_Picture_Jane_Smith.jpg	300 KB	/uploads/pictures/	2023-05-11
3	Performance_Review_James_Brown.docx	1.8 MB	/uploads/reviews/	2023-05-12
4	Contract_Annexure_Alice_Johnson.pdf	800 KB	/uploads/contracts/	2023-05-13
5	Training_Certificate_Robert_Lee.pdf	1.5 MB	/uploads/certificates/	2023-05-14
6	Promotion_Letter_Susan_Wilson.pdf	1.2 MB	/uploads/letters/	2023-05-15
7	Offer_Letter_Michael_Chen.pdf	900 KB	/uploads/offers/	2023-05-16
8	Resignation_Letter_Linda_Wang.pdf	700 KB	/uploads/resignations/	2023-05-17
9	Leave_Application_Kevin_Lin.pdf	600 KB	/uploads/leave/	2023-05-18
10	Training_Plan_Mary_Lee.docx	2.0 MB	/uploads/training/	2023-05-19

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)

-- 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 Administrato', '2017-09-12', '2019-01-15'), (10, 'Digital Marketing Co.', 'Marketing Manager', '2021-03-05', '2022-06-30')	Affected rows: 10	0.002s
---	-------------------	--------

SELECT * FROM work_exp;

Before:

work_exp_id	emp_files_id	company_name	POSITION	start_date	end_date
	(N/A)	(N/A)	(N/A)	(N/A)	(N/A)

After:

work_exp_id	emp_files_id	company_name	POSITION	start_date	end_date
1	1	ABC Corporation	Software Engineer	2020-01-05	2021-02-28
2	2	XYZ Tech	Web Developer	2019-03-15	2020-12-31
3	3	Acme Solutions	Project Manager	2018-06-10	2019-12-15
4	4	Tech Innovators	Data Analyst	2022-03-01	2022-08-31
5	5	Global Industries	Marketing Specialist	2021-05-20	2022-03-10
6	6	Software Solutions	Software Developer	2020-08-10	2021-11-30
7	7	Innovative Tech	Systems Administrator	2019-02-15	2020-05-31
8	8	Tech Solutions	Network Engineer	2022-01-01	2022-09-30
9	9	Data Systems	Database Administrato	2017-09-12	2019-01-15
10	10	Digital Marketing Co.	Marketing Manager	2021-03-05	2022-06-30

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)

-- 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')	Affected rows: 10	0.002s
--	-------------------	--------

SELECT * FROM employee_health_checkups;

Before:

Employee_Health_Checkups_id	emp_files_id	Check_date	height	weight	blood_pressure
	(N/A)	(N/A)	(N/A)	(N/A)	(N/A)

After:

Employee_Health_Chec	emp_files_id	Check_date	height	weight	blood_pressure
1	1	2022-05-15	170 cm	65 kg	120/80 mmHg
2	2	2021-12-30	165 cm	55 kg	110/70 mmHg
3	3	2023-02-20	175 cm	75 kg	130/85 mmHg
4	4	2022-08-10	160 cm	50 kg	115/75 mmHg
5	5	2021-06-25	180 cm	85 kg	140/90 mmHg
6	6	2023-01-05	172 cm	70 kg	125/80 mmHg
7	7	2022-03-12	168 cm	60 kg	118/78 mmHg
8	8	2021-11-18	178 cm	80 kg	135/85 mmHg
9	9	2022-01-30	162 cm	53 kg	112/72 mmHg
10	10	2023-04-05	185 cm	90 kg	145/95 mmHg

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)

-- 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', 150), (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', 250), (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', 180), (10, 'Fine', 'Insubordination', '2023-04-05', 180)	Affected rows: 10	0.004s
---	-------------------	--------

SELECT * FROM emp_penalites;

Before:

emp_penalty_id	emp_files_id	TYPE	reason	DATE	amount
		(N/A)	(N/A)	(N/A)	(N/A)

After:

emp_penalty_id	emp_files_id	TYPE	reason	DATE	amount
1	1	Fine	Late to work	2022-05-15	100
2	2	Deduction	Violation of company policy	2021-12-30	200
3	3	Warning	Unprofessional behavior	2023-02-20	(Null)
4	4	Fine	Unauthorized absence	2022-08-10	150
5	5	Deduction	Misuse of company resources	2021-06-25	300
6	6	Warning	Failure to meet performance targets	2023-01-05	(Null)
7	7	Fine	Violation of safety protocols	2022-03-12	120
8	8	Deduction	Breach of confidentiality	2021-11-18	250
9	9	Warning	Poor time management	2022-01-30	(Null)
10	10	Fine	Insubordination	2023-04-05	180

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)

-- 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'),	Affected rows: 10	0.003s
--	-------------------	--------

SELECT * FROM edu_history;

Before:

edu_history_id	emp_files_id	school_name	major	LEVEL	graduate_date
	(N/A)	(N/A)	(N/A)	(N/A)	(N/A)

After:

edu_history_id	emp_files_id	school_name	major	LEVEL	graduate_date
1	1	ABC University	Computer Science	Bachelor	2021-06-30
2	2	XYZ College	Business Administration	Master	2020-12-15
3	3	123 School of Engineering	Mechanical Engineering	Bachelor	2022-05-20
4	4	DEF Institute	Finance	Bachelor	2019-09-10
5	5	LMN College	Marketing	Master	2023-03-25
6	6	456 University	Electrical Engineering	Bachelor	2022-01-05
7	7	GHI College	Human Resources	Master	2021-07-12
8	8	UVW School of Medicine	Medicine	Doctorate	2020-11-18
9	9	PQR Institute of Technology	Information Technology	Bachelor	2023-01-30
10	10	JKL College	Psychology	Master	2022-04-05

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)

-- 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')	Affected rows: 10	0.002s
---	-------------------	--------

SELECT * FROM contracts;

Before:

contract_id	emp_files_id	start_date	end_date	types	status
	(N/A)	(N/A)	(N/A)	(N/A)	(N/A)

After:

contract_id	emp_files_id	start_date	end_date	types	status
1	1	2021-06-01	2023-05-31	Full-time	Active
2	2	2020-12-15	2022-12-14	Part-time	Active
3	3	2022-03-10	2024-03-09	Contractor	Active
4	4	2019-09-01	2022-08-31	Full-time	Inactive
5	5	2023-01-15	2023-07-14	Internship	Active
6	6	2022-08-20	2023-08-19	Part-time	Active
7	7	2021-07-01	2023-06-30	Full-time	Active
8	8	2020-11-01	2021-10-31	Contractor	Inactive
9	9	2023-02-05	2024-02-04	Full-time	Active
10	10	2022-04-10	2023-04-09	Internship	Active

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)

-- INSERT DATA INTO TABLE resignations INSERT INTO resignations (resign_reason, RESIGN_DATE, related_contract_id, exit_interviewer_id) VALUES ('Personal reasons', '2023-05-15', 1, 5),	Affected rows: 10	0.005s
--	-------------------	--------

SELECT * FROM resignations;

Before:

resign_emp_id	resign_reason	RESIGN_DATE	related_contract_id	exit_interviewer_id
	(N/A)	(N/A)	(N/A)	(N/A)

After:

resign_emp_id	resign_reason	RESIGN_DATE	related_contract_id	exit_interviewer_id
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

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)

-- INSERT DATA INTO TABLE employee_attendance INSERT INTO employee_attendance (Leave_Record_id, Leave_req_id, OT_Record_id, attendance_record_id) VALUES (NULL, NULL, NULL, NULL),	Affected rows: 10	0.005s
---	-------------------	--------

```
SELECT * FROM employee_attendance;
```

Before:

employee_Attendance_id	Leave_Record_id	Leave_req_id	OT_Record_id	attendance_record_id
	(N/A)	(N/A)	(N/A)	(N/A)

After:

employee_Attendance_id	Leave_Record_id	Leave_req_id	OT_Record_id	attendance_record_id
1	(Null)	(Null)	(Null)	(Null)
2	(Null)	(Null)	(Null)	(Null)
3	(Null)	(Null)	(Null)	(Null)
4	(Null)	(Null)	(Null)	(Null)
5	(Null)	(Null)	(Null)	(Null)
6	(Null)	(Null)	(Null)	(Null)
7	(Null)	(Null)	(Null)	(Null)
8	(Null)	(Null)	(Null)	(Null)
9	(Null)	(Null)	(Null)	(Null)
10	(Null)	(Null)	(Null)	(Null)

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)

-- 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);	Affected rows: 10	0.003s
--	-------------------	--------

SELECT * FROM ot_record;

Before:

OT_Record_id	record_date	time_in	time_out	ot_hour
	(N/A)	(N/A)	(N/A)	(N/A)

After:

OT_Record_id	record_date	time_in	time_out	ot_hour
1	2023-05-01	08:30:00	17:30:00	9.00
2	2023-05-02	09:00:00	18:00:00	8.00
3	2023-05-03	08:00:00	17:00:00	0.00
4	2023-05-04	10:00:00	19:00:00	9.00
5	2023-05-05	09:30:00	18:30:00	9.00
6	2023-05-06	08:45:00	18:45:00	10.00
7	2023-05-07	09:15:00	19:15:00	10.00
8	2023-05-08	08:30:00	17:30:00	9.00
9	2023-05-09	09:00:00	18:00:00	8.00
10	2023-05-10	08:00:00	17:00:00	0.00

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)

-- 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),	Affected rows: 10	0.003s
--	-------------------	--------

SELECT * FROM attendance_record;

Before:

attendance_record_id	record_date	time_in	time_out	OT_Record_id
	(N/A)	(N/A)	(N/A)	(N/A)

After:

attendance_record_id	record_date	time_in	time_out	OT_Record_id
1	2023-05-01	08:30:00	17:30:00	1
2	2023-05-02	09:00:00	18:00:00	2
3	2023-05-03	08:00:00	17:00:00	3
4	2023-05-04	10:00:00	19:00:00	4
5	2023-05-05	09:30:00	18:30:00	5
6	2023-05-06	08:45:00	18:45:00	6
7	2023-05-07	09:15:00	19:15:00	7
8	2023-05-08	08:30:00	17:30:00	8
9	2023-05-09	09:00:00	18:00:00	9
10	2023-05-10	08:00:00	17:00:00	10

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)

-- 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'),	Affected rows: 10	0.003s
---	-------------------	--------

SELECT * FROM leave_request;

Before:

leave_req_id	requester_id	req_reason	req_start_date	req_end_date	approve_status
	(N/A)	(N/A)	(N/A)	(N/A)	(N/A)

After:

leave_req_id	requester_id	req_reason	req_start_date	req_end_date	approve_status
1	1	Vacation	2023-05-15	2023-05-18	Pending
2	2	Sick Leave	2023-05-20	2023-05-22	Approved
3	3	Family Emergency	2023-05-10	2023-05-11	Approved
4	4	Personal Leave	2023-05-25	2023-05-25	Pending
5	5	Vacation	2023-05-12	2023-05-14	Approved
6	6	Sick Leave	2023-05-30	2023-05-31	Pending
7	7	Vacation	2023-05-23	2023-05-25	Approved
8	8	Personal Leave	2023-05-05	2023-05-06	Approved
9	9	Sick Leave	2023-05-02	2023-05-03	Approved
10	10	Family Emergency	2023-05-28	2023-05-29	Pending

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)

-- 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)	Affected rows: 10	0.003s
---	-------------------	--------

SELECT * FROM leave_record;

Before:

leave_record_id	leave_req_id	leave_start_date	leave_end_date	leave_approver_id
	(N/A)	(N/A)	(N/A)	(N/A)

After:

leave_record_id	leave_req_id	leave_start_date	leave_end_date	leave_approver_id
1	1	2023-05-15	2023-05-18	1
2	2	2023-05-20	2023-05-22	2
3	3	2023-05-10	2023-05-11	3
4	4	2023-05-25	2023-05-25	4
5	5	2023-05-12	2023-05-14	5
6	6	2023-05-30	2023-05-31	6
7	7	2023-05-23	2023-05-25	7
8	8	2023-05-05	2023-05-06	8
9	9	2023-05-02	2023-05-03	9
10	10	2023-05-28	2023-05-29	10

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)

-- 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', 'Bachelor's degree in Marketing or related field'), (3, 3, '3-5 years of experience', 'Bachelor's degree in Finance or Accounting'), (4, 4, '2-3 years of experience', 'High school diploma or equivalent'), (5, 5, '4-6 years of experience', 'Bachelor's degree in Business or a related field'), (6, 6, '3+ years of experience', 'Bachelor's degree in Project Management or a related field'), (7, 7, '5-7 years of experience', 'Bachelor's degree in Human Resources or a related field'), (8, 8, '2-4 years of experience', 'Bachelor's degree in Computer Science or Software Engineering'), (9, 9, '1-2 years of experience', 'Bachelor's degree in Data Science or a related field'), (10, 5, '6+ years of experience', 'Bachelor's degree in Business Administration or a related field')	Affected rows: 10	0.011s
--	-------------------	--------

SELECT * FROM position_opening;

Before:

Position_Opening_id	Position_id	Dept_id	Experience_Required	Education_Required
	(N/A)	(N/A)	(N/A)	(N/A)

After:

Position_Opening_id	Position_id	Dept_id	Experience_Required	Education_Required
1	1	1	2-4 years of relevant ex	Bachelor's degree in Computer Science
2	2	2	5+ years of experience	Bachelor's degree in Marketing or related field
3	3	3	3-5 years of experience	Bachelor's degree in Finance or Accounting
4	4	4	2-3 years of experience	High school diploma or equivalent
5	5	5	4-6 years of experience	Bachelor's degree in Business or a related field
6	6	1	3+ years of experience	Bachelor's degree in Project Management or a related field
7	7	2	5-7 years of experience	Bachelor's degree in Human Resources or a related field
8	8	3	2-4 years of experience	Bachelor's degree in Computer Science or Software Engineering
9	9	4	1-2 years of experience	Bachelor's degree in Data Science or a related field
10	10	5	6+ years of experience	Bachelor's degree in Business Administration or a related field

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)

-- INSERT DATA INTO TABLE employee_benefits INSERT INTO employee_benefits (medical_insurance, sick_leave, vision_insurance, maternity_leave) VALUES (1, 1, 0, 1), (0, 0, 1, 0), (1, 0, 1, 1), (1, 1, 1, 0), (0, 1, 0, 1), (1, 0, 0, 1), (1, 1, 0, 0), (0, 1, 1, 1), (1, 1, 1, 1), (0, 0, 1, 0);	Affected rows: 10	0.007s
--	-------------------	--------

SELECT * FROM employee_benefits;

Before:

Employee_Benefits_id	medical_insurance	sick_leave	vision_insurance	maternity_leave
	(N/A)	(N/A)	(N/A)	(N/A)

After:

Employee_Benefits_id	medical_insurance	sick_leave	vision_insurance	maternity_leave
1	1	1	0	1
2	0	1	1	0
3	1	1	1	1
4	1	0	0	1
5	1	1	0	0
6	0	1	1	1
7	1	0	1	1
8	1	1	1	0
9	0	0	1	1
10	1	1	0	1

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)

-- 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', ...	Affected rows: 10	0.004s
--	-------------------	--------

SELECT * FROM employee_awards;

Before:

Employee_Awards_id	Employee_Benefits_id	Title	Reason	date	Amount
	(N/A)	(N/A)	(N/A)	(N/A)	(N/A)

After:

Employee_Awards_id	Employee_Benefits_id	Title	Reason	date	Amount
1	1	Outstanding Performance Award	Exceeded sales targets for Q1	2023-04-15	1000
2	2	Team Collaboration Award	Contributed significantly to a successful project	2023-03-22	800
3	3	Innovation Excellence Award	Introduced a new cost-saving process	2023-04-05	1200
4	4	Leadership Achievement Award	Successfully led a cross-functional team	2023-02-10	1500
5	5	Customer Service Award	Received outstanding customer feedback	2023-05-02	600
6	1	Employee of the Month	Consistently high performance and dedication	2023-04-30	500
7	3	Quality Improvement Award	Improved product quality and customer satisfaction	2023-03-18	1000
8	2	Excellence in Problem-Solving	Resolved a critical issue with efficiency	2023-05-08	700
9	4	Inspirational Leadership Award	Motivated team members to achieve their best	2023-02-28	900
10	5	Safety Champion Award	Promoted a culture of safety in the workplace	2023-04-20	800

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)

-- 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),	Affected rows: 10	0.001s
--	-------------------	--------

SELECT * FROM employees_salary_details;

Before:

sal_detail_id	Employee_Benefits_id	BASE_SALARY	BONUS	ALLOWANCE	DEDUCTIONS	BENEFITS
		(N/A)	(N/A)	(N/A)	(N/A)	(N/A)

After:

sal_detail_id	Employee_Benefits_id	BASE_SALARY	BONUS	ALLOWANCE	DEDUCTIONS	BENEFITS
1	1	5000.00	1000.00	500.00	200.00	300.00
2	2	4800.00	800.00	450.00	180.00	280.00
3	3	5200.00	1200.00	600.00	250.00	320.00
4	4	5500.00	1500.00	700.00	280.00	350.00
5	5	4700.00	750.00	400.00	150.00	240.00
6	1	4900.00	900.00	480.00	190.00	290.00
7	3	5100.00	1100.00	580.00	230.00	310.00
8	2	4600.00	700.00	420.00	160.00	260.00
9	4	5300.00	1400.00	650.00	270.00	330.00
10	5	4800.00	800.00	450.00	180.00	280.00

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)

-- 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'), (6, 'Female', 5400.00, '2023-02-01', '2023-02-28'), (7, 'Male', 6000.00, '2023-02-01', '2023-02-28'), (8, 'Male', 5000.00, '2023-02-01', '2023-02-28'), (9, 'Female', 6200.00, '2023-02-01', '2023-02-28'), (10, 'Male', 5500.00, '2023-02-01', '2023-02-28')	Affected rows: 10	0.001s
--	-------------------	--------

SELECT * FROM salary_history;

Before:

SAL_HISTORY_ID	Employee_Benefits_id	gender	TOTAL_SALARY	START_DATE	END_DATE
	(N/A)	(N/A)	(N/A)	(N/A)	(N/A)

After:

SAL_HISTORY_ID	Employee_Benefits_id	gender	TOTAL_SALARY	START_DATE	END_DATE
1	1	Male	5500.00	2023-01-01	2023-01-31
2	2	Female	5200.00	2023-01-01	2023-01-31
3	3	Male	5800.00	2023-01-01	2023-01-31
4	4	Female	6100.00	2023-01-01	2023-01-31
5	5	Male	5300.00	2023-01-01	2023-01-31
6	1	Female	5400.00	2023-02-01	2023-02-28
7	3	Male	6000.00	2023-02-01	2023-02-28
8	2	Male	5000.00	2023-02-01	2023-02-28
9	4	Female	6200.00	2023-02-01	2023-02-28
10	5	Male	5500.00	2023-02-01	2023-02-28

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)

-- 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'),	Affected rows: 20	0.005s
---	-------------------	--------

SELECT * FROM skills;

Before:

Skill_id	Skill_name	dept_id	skill_resources	skill_description
	(N/A)	(N/A)	(N/A)	(N/A)

After:

Skill_id	Skill_name	dept_id	skill_resources	skill_description
1	Sales Techniques	1	Sales training materials	Techniques to improve sales performance
2	Negotiation Skills	1	Negotiation workshops	Skills for effective negotiation
3	Customer Relationship Management	1	CRM software	Managing customer relationships effectively
4	Presentation Skills	1	Presentation workshops	Improving presentation and communication skills
5	Java	2	Online Java tutorials	Java programming language
6	Python	2	Python IDEs	Python programming language
7	Web Development	2	Web development resources	Skills for building web applications
8	Database Management	2	Database management tools	Managing and optimizing databases
9	Recruitment and Selection	3	Recruitment guidelines	Best practices for hiring and selection
10	Performance Management	3	Performance appraisal tools	Managing employee performance
11	Training and Development	3	Training modules	Developing employee skills through training
12	Employee Relations	3	Employee relation resources	Managing employee relations and conflicts
13	Digital Marketing	4	Online marketing platforms	Skills for digital marketing campaigns
14	Market Research	4	Market research tools	Conducting market research and analysis
15	Brand Management	4	Brand management guides	Managing and building brand identity
16	Social Media Marketing	4	Social media marketing resources	Utilizing social media for marketing
17	Financial Analysis	5	Financial analysis software	Analyzing financial data and reports
18	Budgeting and Forecasting	5	Budgeting tools	Creating budgets and financial forecasts
19	Taxation	5	Taxation guidelines	Understanding and complying with taxation laws
20	Risk Management	5	Risk management strategies	Identifying and managing financial risks

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)

-- 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,	Affected rows: 10	0.003s
--	-------------------	--------

SELECT * FROM candidates;

Before:

Candidate_id	Candidate_name	Email	phone_no	Skill_set_id	position_Opening_id
	(N/A)	(N/A)	(N/A)	(N/A)	(N/A)

After:

Candidate_id	Candidate_name	Email	phone_no	Skill_set_id	position_Opening_id
1	Emily Turner	emily.turner@example.com	020-11111111	12	6
2	Jacob Parker	jacob.parker@example.com	021-22222222	13	8
3	Olivia Mitchell	olivia.mitchell@example.com	022-33333333	14	2
4	Michael Scott	michael.scott@example.com	023-44444444	15	4
5	Sophia Young	sophia.young@example.com	024-55555555	16	9
6	William Allen	william.allen@example.com	025-66666666	19	5
7	Emma Hill	emma.hill@example.com	026-77777777	20	10
8	Noah Turner	noah.turner@example.com	027-88888888	11	3
9	Ava Reed	ava.reed@example.com	028-99999999	6	7
10	James Evans	james.evans@example.com	029-00000000	8	2

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)

-- 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),	Affected rows: 10	0.001s
--	-------------------	--------

SELECT * FROM skillset;

Before:

Skillset_id	emp_pnd_id	candidate_id	Skill_one_id	Skill_two_id	Skill_three_id
	(N/A)	(N/A)	(N/A)	(N/A)	(N/A)

After:

Skillset_id	emp_pnd_id	candidate_id	Skill_one_id	Skill_two_id	Skill_three_id
1	(Null)	(Null)	2	3	6
2	(Null)	(Null)	8	9	11
3	(Null)	(Null)	14	15	17
4	(Null)	(Null)	19	20	4
5	(Null)	(Null)	10	12	18
6	(Null)	1	3	5	7
7	(Null)	2	9	13	16
8	(Null)	3	17	6	12
9	(Null)	4	8	10	15
10	(Null)	5	2	4	20

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)

-- 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, 4, 5, 'I believe I have the required skills', 5), (3, 5, 9, 'I want to take on new challenges', 10), (4, 6, 1, 'I have relevant experience in this area', 2), (5, 7, 3, 'I see this as an opportunity for growth', 8), (6, 8, 2, 'I am passionate about this role', 6), (7, 9, 4, 'I am excited about the company culture', 3), (8, 10, 1, 'I am looking for a new opportunity', 1), (9, 1, 3, 'I want to contribute to the team', 9), (10, 2, 2, 'I am ready to take on more responsibility', 4)	Affected rows: 10	0.003s
--	-------------------	--------

SELECT * FROM position_application;

Before:

position_apply_id	emp_id	current_position_id	position_opening_id	application_reason	skillset_id
	(N/A)	(N/A)	(N/A)	(N/A)	(N/A)

After:

position_apply_id	emp_id	current_position_id	position_opening_id	application_reason	skillset_id
1	1	3	8	I am interested in this new position	7
2	2	2	5	I believe I have the required skills	5
3	3	4	9	I want to take on new challenges	10
4	4	1	4	I have relevant experience in this area	2
5	5	3	7	I see this as an opportunity for growth	8
6	6	2	5	I am passionate about this role	6
7	7	4	10	I am excited about the company culture	3
8	8	1	3	I am looking for a new opportunity	1
9	9	3	6	I want to contribute to the team	9
10	10	2	2	I am ready to take on more responsibility	4

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)

-- 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.'	Affected rows: 10	0.004s
--	-------------------	--------

SELECT * FROM interview_list;

Before:

Interview_list_id	Candidate_id	Interviewer_id	Interview_date	Interview_result
	(N/A)	(N/A)	(N/A)	(N/A)

After:

Interview_list_id	Candidate_id	Interviewer_id	Interview_date	Interview_result
1	1	5	2023-05-01	Impressed with candidate's technical skills and problem-solving ability.
2	2	3	2023-05-02	Candidate demonstrated excellent communication and teamwork.
3	3	1	2023-05-03	Candidate has relevant experience and a strong work ethic.
4	4	4	2023-05-04	Interviewer recommended the candidate for further consideration.
5	5	2	2023-05-05	Candidate showed a proactive attitude and willingness to learn.
6	6	6	2023-05-06	Interviewer was impressed with the candidate's leadership potential.
7	7	7	2023-05-07	Candidate displayed a deep understanding of the industry.
8	8	9	2023-05-08	Interviewer noted the candidate's attention to detail and accuracy.
9	9	12	2023-05-09	Candidate demonstrated strong problem-solving skills.
10	10	8	2023-05-10	Interviewer was pleased with the candidate's passion for the role.

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)

-- 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'), (6, 4, '2023-05-06', '2 days'), (7, 8, '2023-05-07', '3 days'), (8, 8, '2023-05-08', '4 days'), (9, 8, '2023-05-09', '1 day'), (10, 8, '2023-05-10', '2 days')	Affected rows: 10	0.005s
--	-------------------	--------

SELECT * FROM employee_training;

Before:

emp_training_id	training_skill_id	trainer_id	training_date	training_duration
	(N/A)	(N/A)	(N/A)	(N/A)

After:

emp_training_id	training_skill_id	trainer_id	training_date	training_duration
1	1	4	2023-05-01	3 days
2	2	4	2023-05-02	2 days
3	3	4	2023-05-03	1 day
4	4	4	2023-05-04	4 days
5	5	8	2023-05-05	5 days
6	1	4	2023-05-06	2 days
7	2	8	2023-05-07	3 days
8	3	8	2023-05-08	4 days
9	4	8	2023-05-09	1 day
10	5	8	2023-05-10	2 days

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)

-- 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),	Affected rows: 10	0.003s
--	-------------------	--------

SELECT * FROM employee_performance_and_development;

Before:

emp_pnd_id	skillset_id	employee_promotion_id	emp_training_id	credit_score	credit_score_grader_id
	(N/A)	(N/A)	(N/A)	(N/A)	(N/A)

After:

emp_pnd_id	skillset_id	employee_promotion_id	emp_training_id	credit_score	credit_score_grader_id
1	1	(Null)	1	85	(Null)
2	2	(Null)	2	78	(Null)
3	3	(Null)	3	92	(Null)
4	4	(Null)	4	79	(Null)
5	5	(Null)	5	88	(Null)
6	1	(Null)	6	95	(Null)
7	2	(Null)	7	82	(Null)
8	3	(Null)	8	90	(Null)
9	4	(Null)	9	75	(Null)
10	5	(Null)	10	87	(Null)

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)

-- 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')	Affected rows: 10	0.003s
---	-------------------	--------

SELECT * FROM employee_promotion;

Before:

emp_promo_id	promo_reason	current_position_id	promoted_position_id	promoted_date
	(N/A)	(N/A)	(N/A)	(N/A)

After:

emp_promo_id	promo_reason	current_position_id	promoted_position_id	promoted_date
1	Excellent performance and leadership skills	3	8	2023-03-15
2	Completed advanced training program	5	10	2023-02-28
3	Consistent high-quality work	2	6	2023-04-10
4	Outstanding teamwork and problem-solving	4	9	2023-03-05
5	Demonstrated exceptional customer service	6	12	2023-04-20
6	Achieved remarkable sales performance	7	13	2023-03-22
7	Strong dedication and commitment	1	4	2023-02-12
8	Leadership skills and team management	9	15	2023-03-18
9	Innovative ideas and project management	8	14	2023-02-25
10	Exemplary communication and problem-solving	10	16	2023-04-05

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)

-- 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'),	Affected rows: 10	0.002s
---	-------------------	--------

SELECT * FROM projects;

Before:

project_id	project_title	leader_id	start_date	end_date	comment
	(N/A)	(N/A)	(N/A)	(N/A)	(N/A)

After:

project_id	project_title	leader_id	start_date	end_date	comment
1	Sales Performance Analysis	2	2023-06-10	2023-08-20	Analyzing sales data for improvement
2	IT System Upgrade	3	2023-07-15	2023-09-25	Enhancing technology infrastructure
3	Employee Training Initiative	1	2023-05-05	2023-07-15	Enhancing employee skills
4	Marketing Campaign Launch	4	2023-06-25	2023-09-05	Launching new marketing campaign
5	Financial Data Management	5	2023-07-01	2023-09-30	Optimizing financial data handling
6	Sales Team Performance Evaluation	2	2023-08-05	2023-10-15	Assessing sales team effectiveness
7	IT Security Enhancement	3	2023-06-15	2023-08-30	Improving IT security measures
8	Talent Acquisition Strategy	1	2023-07-20	2023-10-10	Developing talent acquisition plan
9	Product Marketing Research	4	2023-05-20	2023-07-30	Conducting market research for products
10	Financial Forecasting Project	5	2023-08-15	2023-11-05	Forecasting financial trends

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)

-- 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, 2, 'Working on IT system upgrade', '2023-07-15', '2023-09-25'), (3, 3, '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'), (6, 1, 'Evaluating sales team performance', '2023-08-05', '2023-10-15'), (7, 2, 'Assisting in IT security enhancement', '2023-06-15', '2023-08-30'), (8, 3, 'Participating in talent acquisition strategy', '2023-07-20', '2023-10-10'), (9, 4, 'Conducting product marketing research', '2023-05-20', '2023-07-30'), (10, 5, 'Contributing to financial forecasting', '2023-08-15', '2023-11-05')	Affected rows: 10	0.004s
--	-------------------	--------

SELECT * FROM project_participation;

Before:

project_participate_id	emp_pnd_id	project_id	comment	start_date	end_date
	(N/A)	(N/A)	(N/A)	(N/A)	(N/A)

After:

project_participate_id	emp_pnd_id	project_id	comment	start_date	end_date
1	1	1	Assisting in sales analysis	2023-06-10	2023-08-20
2	2	3	Working on IT system upgrade	2023-07-15	2023-09-25
3	3	2	Participating in employee training	2023-05-05	2023-07-15
4	4	4	Contributing to marketing campaign	2023-06-25	2023-09-05
5	5	5	Supporting financial data management	2023-07-01	2023-09-30
6	1	2	Evaluating sales team performance	2023-08-05	2023-10-15
7	2	3	Assisting in IT security enhancement	2023-06-15	2023-08-30
8	3	1	Participating in talent acquisition strategy	2023-07-20	2023-10-10
9	4	4	Conducting product marketing research	2023-05-20	2023-07-30
10	5	5	Contributing to financial forecasting	2023-08-15	2023-11-05

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)

-- 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')	Affected rows: 20	0.005s
--	-------------------	--------

SELECT * FROM users;

Before:

user_id	emp_id	username	password	permission_type	description	creat_date	last_update_date
	(N/A)	(N/A)	(N/A)	(N/A)	(N/A)	(N/A)	(N/A)

After:

user_id	emp_id	username	password	permission_type	description	creat_date	last_update_date
1	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	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	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	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	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	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	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	8	jessica_liu_9257	19880412	administrator	Users with this permission can add, delete, modify and query all tables and data.	2023-05-08	2023-05-08
9	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	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	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	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	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	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	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	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	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	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	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	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

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)

UPDATE positions SET dept_id = 1 WHERE position_id BETWEEN 1 AND 4	Affected rows: 4	0.003s
UPDATE positions SET dept_id = 2 WHERE position_id BETWEEN 5 AND 8	Affected rows: 4	0.002s
UPDATE positions SET dept_id = 3 WHERE position_id BETWEEN 9 AND 12	Affected rows: 4	0.002s
UPDATE positions SET dept_id = 4 WHERE position_id BETWEEN 13 AND 16	Affected rows: 4	0.001s
UPDATE positions SET dept_id = 5 WHERE position_id BETWEEN 17 AND 20	Affected rows: 4	0.001s

SELECT * FROM positions;

Before:

position_id	position_name	avg_salary	dept_id	emp_count
1	Sales Manager	70000.00	(Null)	1
2	Sales Representative	45000.00	(Null)	8
3	Sales Associate	38000.00	(Null)	10
4	Sales Support Specialist	42000.00	(Null)	4
5	IT Manager	65000.00	(Null)	1
6	Software Engineer	58000.00	(Null)	7
7	Systems Analyst	52000.00	(Null)	5
8	IT Support Specialist	42000.00	(Null)	6
9	HR Manager	60000.00	(Null)	1
10	HR Assistant	45000.00	(Null)	3
11	HR Specialist	48000.00	(Null)	5
12	HR System Administrator	40000.00	(Null)	4
13	Marketing Manager	60000.00	(Null)	5
14	Marketing Specialist	45000.00	(Null)	8
15	Social Media Manager	50000.00	(Null)	3
16	Marketing Analyst	55000.00	(Null)	6
17	Finance Manager	62000.00	(Null)	1
18	Financial Analyst	55000.00	(Null)	3
19	Accountant	48000.00	(Null)	6
20	Auditor	53000.00	(Null)	2

After:

position_id	position_name	avg_salary	dept_id	emp_count
1	Sales Manager	70000.00	1	1
2	Sales Representative	45000.00	1	8
3	Sales Associate	38000.00	1	10
4	Sales Support Specialist	42000.00	1	4
5	IT Manager	65000.00	2	1
6	Software Engineer	58000.00	2	7
7	Systems Analyst	52000.00	2	5
8	IT Support Specialist	42000.00	2	6
9	HR Manager	60000.00	3	1
10	HR Assistant	45000.00	3	3
11	HR Specialist	48000.00	3	5
12	HR System Administrator	40000.00	3	4
13	Marketing Manager	60000.00	4	5
14	Marketing Specialist	45000.00	4	8
15	Social Media Manager	50000.00	4	3
16	Marketing Analyst	55000.00	4	6
17	Finance Manager	62000.00	5	1
18	Financial Analyst	55000.00	5	3
19	Accountant	48000.00	5	6
20	Auditor	53000.00	5	2

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)

UPDATE departments SET dept_manager = 2 WHERE dept_id = 1	Affected rows: 1	0.002s
UPDATE departments SET dept_manager = 3 WHERE dept_id = 2	Affected rows: 1	0.002s
UPDATE departments SET dept_manager = 1 WHERE dept_id = 3	Affected rows: 1	0.001s
UPDATE departments SET dept_manager = 4 WHERE dept_id = 4	Affected rows: 1	0.002s
UPDATE departments SET dept_manager = 5 WHERE dept_id = 5	Affected rows: 1	0.001s

SELECT * FROM departments;

Before:

dept_id	dept_name	dept_manager	location	emp_count
1	Department of sale	(Null)	floor1	10
2	Department of information technology	(Null)	floor2	9
3	Department of Human Resources	(Null)	floor3	5
4	Department of Marketing	(Null)	floor4	5
5	Department of Finance	(Null)	floor5	4

After:

dept_id	dept_name	dept_manager	location	emp_count
1	Department of sale	2	floor1	10
2	Department of information technology	3	floor2	9
3	Department of Human Resources	1	floor3	5
4	Department of Marketing	4	floor4	5
5	Department of Finance	5	floor5	4

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)

-- UPDATE TABLES that need to supplement and adjust data (PART 2) -- employee_attendance UPDATE `employee_attendance` SET 'Leave_Record_id' = 10, 'Leave_req_id' = 10, 'OT_Record_id' = 10, 'OT_req_id' = 10; UPDATE `employee_attendance` SET 'Leave_Record_id' = 9, 'Leave_req_id' = 9, 'OT_Record_id' = 9, 'OT_req_id' = 9; UPDATE `employee_attendance` SET 'Leave_Record_id' = 8, 'Leave_req_id' = 8, 'OT_Record_id' = 8, 'OT_req_id' = 8; UPDATE `employee_attendance` SET 'Leave_Record_id' = 7, 'Leave_req_id' = 7, 'OT_Record_id' = 7, 'OT_req_id' = 7;	Affected rows: 1	0.014s
UPDATE `employee_attendance` SET 'Leave_Record_id' = 6, 'Leave_req_id' = 6, 'OT_Record_id' = 6, 'OT_req_id' = 6;	Affected rows: 1	0.002s
UPDATE `employee_attendance` SET 'Leave_Record_id' = 5, 'Leave_req_id' = 5, 'OT_Record_id' = 5, 'OT_req_id' = 5;	Affected rows: 1	0.002s
UPDATE `employee_attendance` SET 'Leave_Record_id' = 4, 'Leave_req_id' = 4, 'OT_Record_id' = 4, 'OT_req_id' = 4;	Affected rows: 1	0.001s
UPDATE `employee_attendance` SET 'Leave_Record_id' = 3, 'Leave_req_id' = 3, 'OT_Record_id' = 3, 'OT_req_id' = 3;	Affected rows: 1	0.001s
UPDATE `employee_attendance` SET 'Leave_Record_id' = 2, 'Leave_req_id' = 2, 'OT_Record_id' = 2, 'OT_req_id' = 2;	Affected rows: 1	0.001s
UPDATE `employee_attendance` SET 'Leave_Record_id' = 1, 'Leave_req_id' = 1, 'OT_Record_id' = 1, 'OT_req_id' = 1;	Affected rows: 1	0.001s
UPDATE `employee_attendance` SET 'Leave_Record_id' = 2, 'Leave_req_id' = 10, 'OT_Record_id' = 8, 'OT_req_id' = 8;	Affected rows: 1	0.001s
UPDATE `employee_attendance` SET 'Leave_Record_id' = 1, 'Leave_req_id' = 9, 'OT_Record_id' = 7, 'OT_req_id' = 7;	Affected rows: 1	0.001s

```
SELECT * FROM employee_attendance;
```

Before:

employee_Attendance_id	Leave_Record_id	Leave_req_id	OT_Record_id	attendance_record_id
1	(Null)	(Null)	(Null)	(Null)
2	(Null)	(Null)	(Null)	(Null)
3	(Null)	(Null)	(Null)	(Null)
4	(Null)	(Null)	(Null)	(Null)
5	(Null)	(Null)	(Null)	(Null)
6	(Null)	(Null)	(Null)	(Null)
7	(Null)	(Null)	(Null)	(Null)
8	(Null)	(Null)	(Null)	(Null)
9	(Null)	(Null)	(Null)	(Null)
10	(Null)	(Null)	(Null)	(Null)

After:

employee_Attendance_	Leave_Record_id	Leave_req_id	OT_Record_id	attendance_record_id
1	10	8	6	4
2	9	7	5	3
3	8	6	4	2
4	7	5	3	1
5	6	4	2	10
6	5	3	1	9
7	4	2	10	8
8	3	1	9	7
9	2	10	8	6
10	1	9	7	5

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)

-- skillset	Affected rows: 1	0.002s
UPDATE skillset SET emp_pnd_id = 6 WHERE skillset_id = 1	Affected rows: 1	0.001s
UPDATE skillset SET emp_pnd_id = 7 WHERE skillset_id = 2	Affected rows: 1	0.002s
UPDATE skillset SET emp_pnd_id = 8 WHERE skillset_id = 3	Affected rows: 1	0.001s
UPDATE skillset SET emp_pnd_id = 9 WHERE skillset_id = 4	Affected rows: 1	0.001s
UPDATE skillset SET emp_pnd_id = 10 WHERE skillset_id = 5	Affected rows: 1	0.001s

SELECT * FROM skillset;

Before:

Skillset_id	emp_pnd_id	candidate_id	Skill_one_id	Skill_two_id	Skill_three_id
1	(Null)	(Null)	2	3	6
2	(Null)	(Null)	8	9	11
3	(Null)	(Null)	14	15	17
4	(Null)	(Null)	19	20	4
5	(Null)	(Null)	10	12	18
6	(Null)	1	3	5	7
7	(Null)	2	9	13	16
8	(Null)	3	17	6	12
9	(Null)	4	8	10	15
10	(Null)	5	2	4	20

After:

Skillset_id	emp_pnd_id	candidate_id	Skill_one_id	Skill_two_id	Skill_three_id
1	6	(Null)	2	3	6
2	7	(Null)	8	9	11
3	8	(Null)	14	15	17
4	9	(Null)	19	20	4
5	10	(Null)	10	12	18
6	(Null)	1	3	5	7
7	(Null)	2	9	13	16
8	(Null)	3	17	6	12
9	(Null)	4	8	10	15
10	(Null)	5	2	4	20

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)

-- employee_performance_and_development UPDATE `employee_performance_and_development` SET `employee_promotion_id` = 2, `credit_score_grader_id` = 3 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	Affected rows: 1	0.002s
	Affected rows: 1	0.001s
	Affected rows: 1	0s
	Affected rows: 1	0.001s
	Affected rows: 1	0.001s

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	Affected rows: 1	0.001s
	Affected rows: 1	0.001s
	Affected rows: 1	0.001s
	Affected rows: 1	0s
	Affected rows: 1	0.001s

```
SELECT * FROM employee_performance_and_development;
```

Before:

emp_pnd_id	skillset_id	employee_promotion_id	emp_training_id	credit_score	credit_score_grader_id
1	1	(Null)	1	85	(Null)
2	2	(Null)	2	78	(Null)
3	3	(Null)	3	92	(Null)
4	4	(Null)	4	79	(Null)
5	5	(Null)	5	88	(Null)
6	1	(Null)	6	95	(Null)
7	2	(Null)	7	82	(Null)
8	3	(Null)	8	90	(Null)
9	4	(Null)	9	75	(Null)
10	5	(Null)	10	87	(Null)

After:

emp_pnd_id	skillset_id	employee_promotion_id	emp_training_id	credit_score	credit_score_grader_id
1	1	2	1	85	3
2	2	5	2	78	4
3	3	8	3	92	9
4	4	1	4	79	7
5	5	10	5	88	6
6	1	3	6	95	5
7	2	6	7	82	8
8	3	9	8	90	1
9	4	4	9	75	2
10	5	7	10	87	10

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:

resign_emp_id	resign_reason	RESIGN_DATE	related_contract_id	exit_interviewer_id
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

After:

resign_emp_id	resign_reason	RESIGN_DATE	related_contract_id	exit_interviewer_id
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
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

DELETE(all data)

DELETE Commands:

DELETE FROM resignations;

Excuted Result (with Query Commands):

SELECT * FROM resignations;

Before:

resign_emp_id	resign_reason	RESIGN_DATE	related_contract_id	exit_interviewer_id
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
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

After:

resign_emp_id	resign_reason	RESIGN_DATE	related_contract_id	exit_interviewer_id
	(N/A)	(N/A)	(N/A)	(N/A)

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 example 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):

dept_id	dept_name	dept_manager	location	emp_count
1	Department of sale	2	floor1	10
2	Department of information technology	3	floor2	9
3	Department of Human Resources	1	floor3	5
4	Department of Marketing	4	floor4	5
5	Department of Finance	5	floor5	4

2. 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):

employee_id	name	gender	birth_date	phone_no	dept_id	position_id
3	Michael Johnson	Male	1988-11-30	013-55555555	2	5
7	Robert Chen	Male	1986-12-05	017-66666666	2	7
12	Ella Williams	Female	1990-05-22	011-98765432	2	6
15	Liam Taylor	Male	1994-09-18	011-11223344	2	7
18	Aria Wilson	Female	1993-04-06	011-66778899	2	8

3. 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):

emp_id	name	dept_name
2	Jane Smith	Department of sale
6	Sarah Wang	Department of sale
11	Alex Johnson	Department of sale
13	Noah Smith	Department of sale
16	Mia Brown	Department of sale
19	Ethan Miller	Department of sale
3	Michael Johnson	Department of information technology
7	Robert Chen	Department of information technology
12	Ella Williams	Department of information technology
15	Liam Taylor	Department of information technology
18	Aria Wilson	Department of information technology
1	John Doe	Department of Human Resources

4. 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):

emp_id	name
3	Michael Johnson
7	Robert Chen
12	Ella Williams
15	Liam Taylor
18	Aria Wilson

5. 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):

dept_id	num_employees
1	6
2	5
3	5
4	2
5	2

6. Sort by employee number in descending order from the "employees" table and retrieve the employee number and name (retrieve 10 data only).

SELECT Commands:

```
SELECT emp_id, name FROM employees ORDER BY emp_id DESC LIMIT 10;
```

Excuted Result (with Query Commands):

emp_id	name
20	Scarlett Martin
19	Ethan Miller
18	Aria Wilson
17	Jameson Lee
16	Mia Brown
15	Liam Taylor
14	Ava Davis
13	Noah Smith
12	Ella Williams
11	Alex Johnson

7. Calculates the average salary of all employees of department IT and displays the result as "avg_salary".

SELECT Commands:

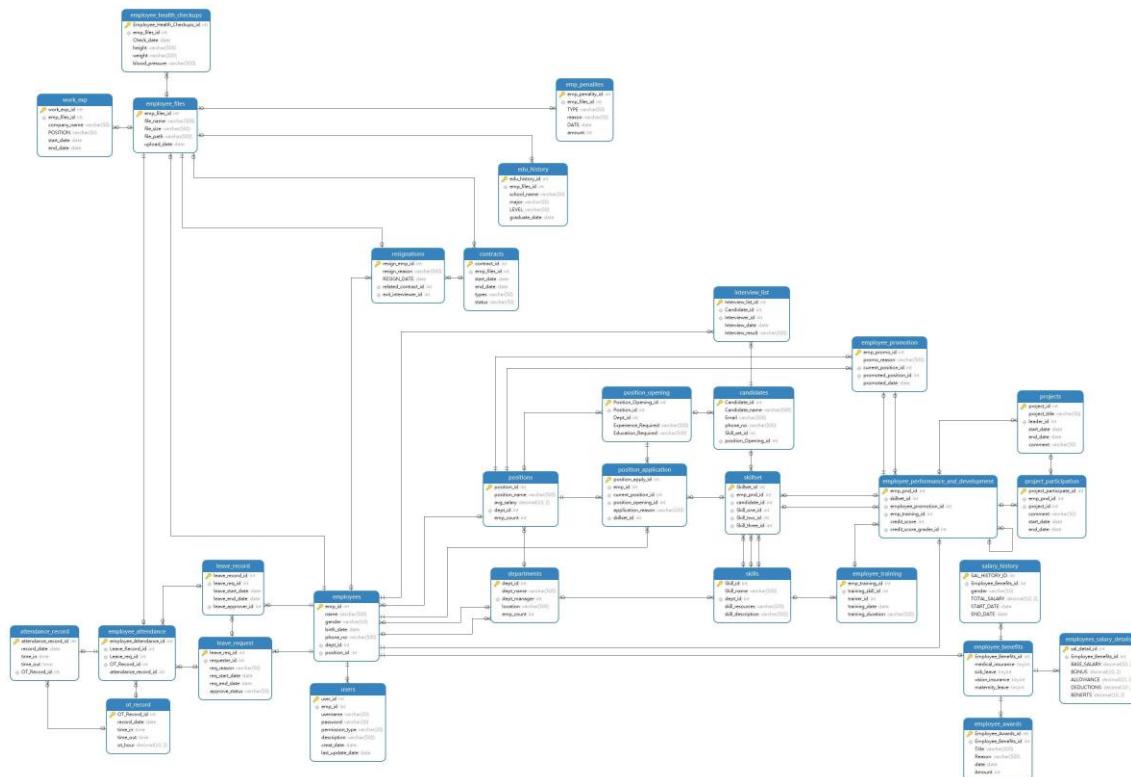
```
SELECT d.dept_id, d.dept_name, AVG(esd.BASE_SALARY) AS average_salary  
FROM employees e  
LEFT JOIN departments d ON e.dept_id = d.dept_id  
LEFT JOIN employee_benefits eb ON e.emp_id = eb.Employee_Benefits_id  
LEFT JOIN employees_salary_details esd ON eb.Employee_Benefits_id =  
esd.Employee_Benefits_id  
WHERE d.dept_name = 'Department of information technology'  
GROUP BY d.dept_id, d.dept_name;
```

Excuted Result (with Query Commands):

dept_id	dept_name	average_salary
2	Department of information technology	5150.000000

3.0 Diagrams

3.1 ER Diagram



(Clearer picture in folder “extra documents”)

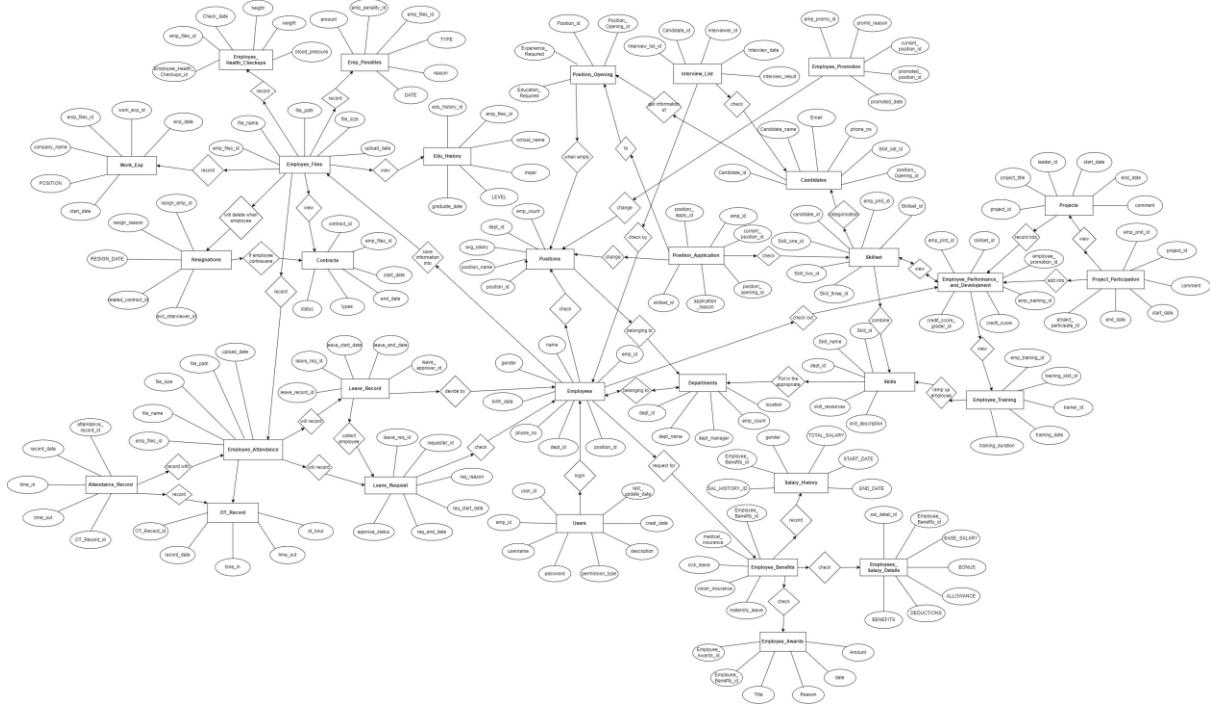
3.2 Table Relationship

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	Many 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:

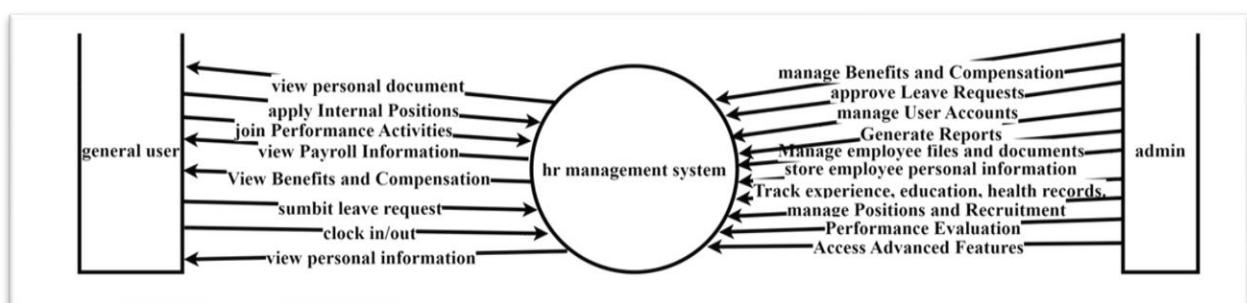
General User (Normal Employee):

- clock In/Ot: employee can take attendant and overtime record
- view employee file: view employee file
- submit leave / resignation form: employee can submit 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

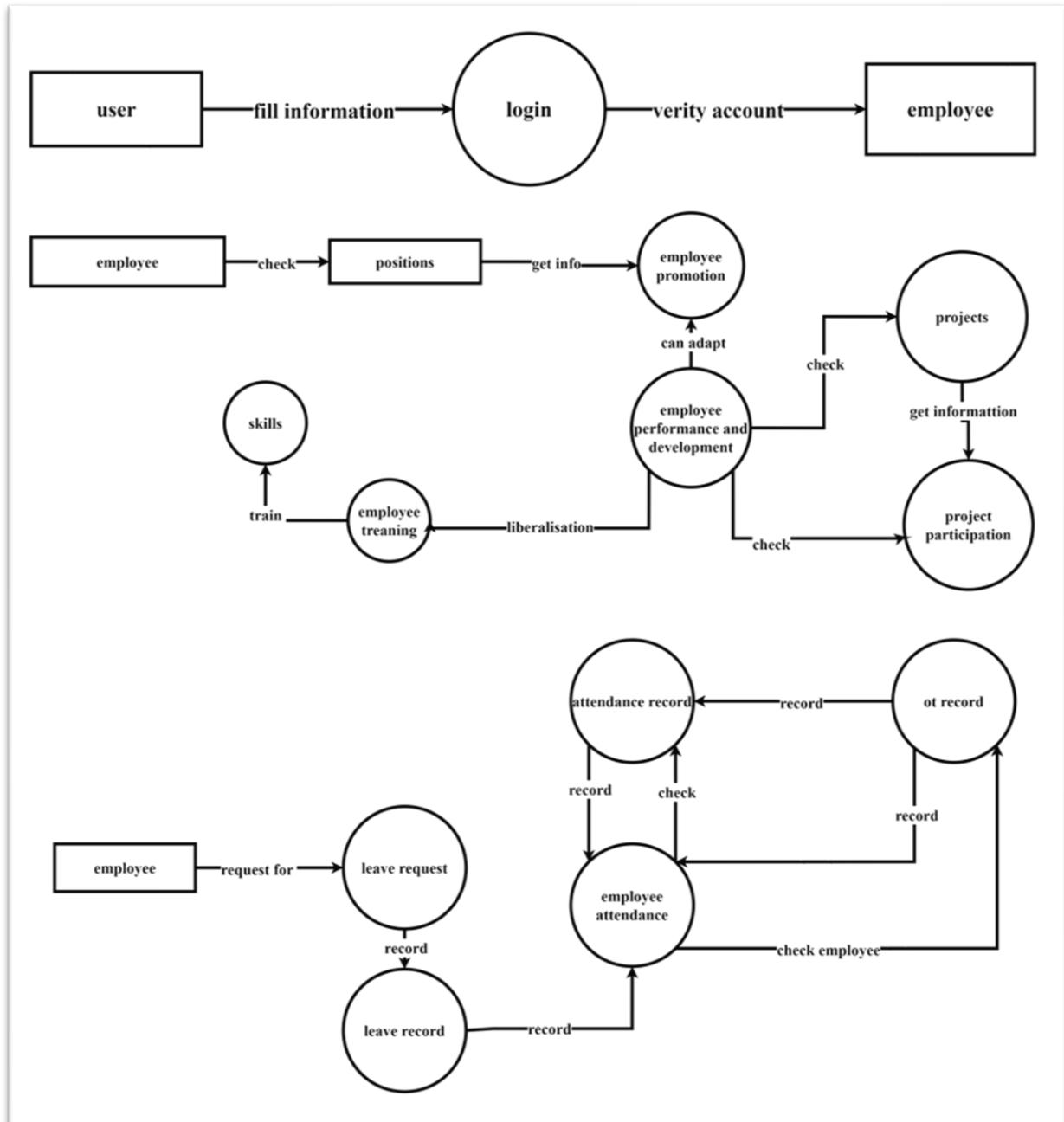
Administrator:

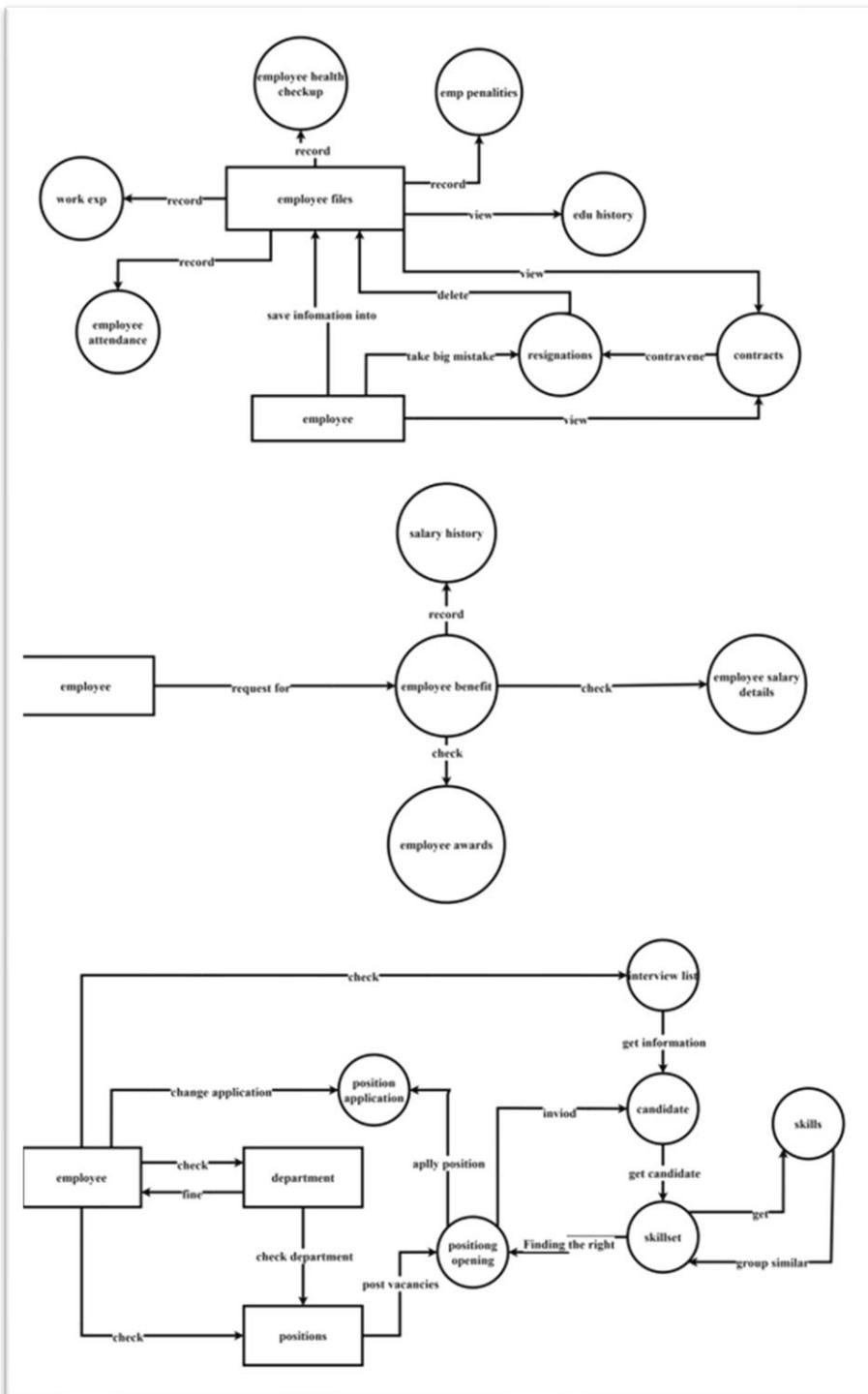
- manage Positions and Recruitment: manage employee position and hiring candidates
- manage employee development: manage employee training
- manage form: accept or reject leave form and resignation form
- manage User Accounts: can update employee username and password
- manage employee file: update all 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

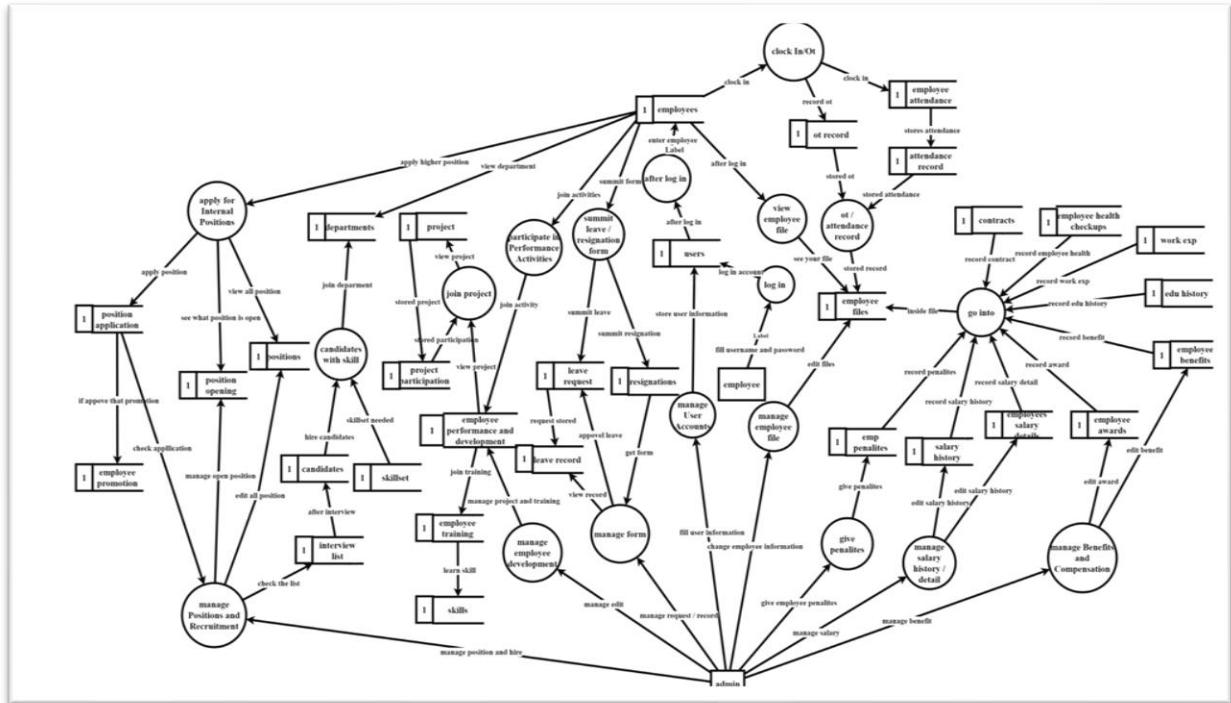


Level 1





Level 2



(Clearer picture in folder “extra documents”)

3.5 Mock-ups

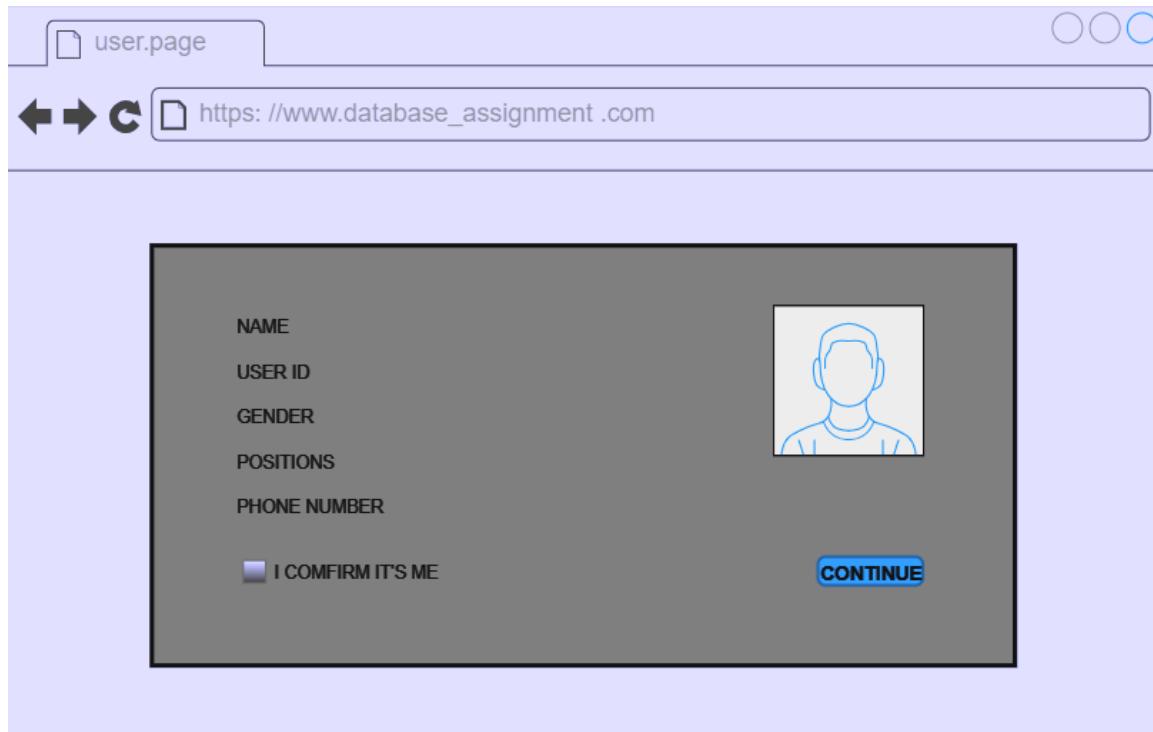
Login Page



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

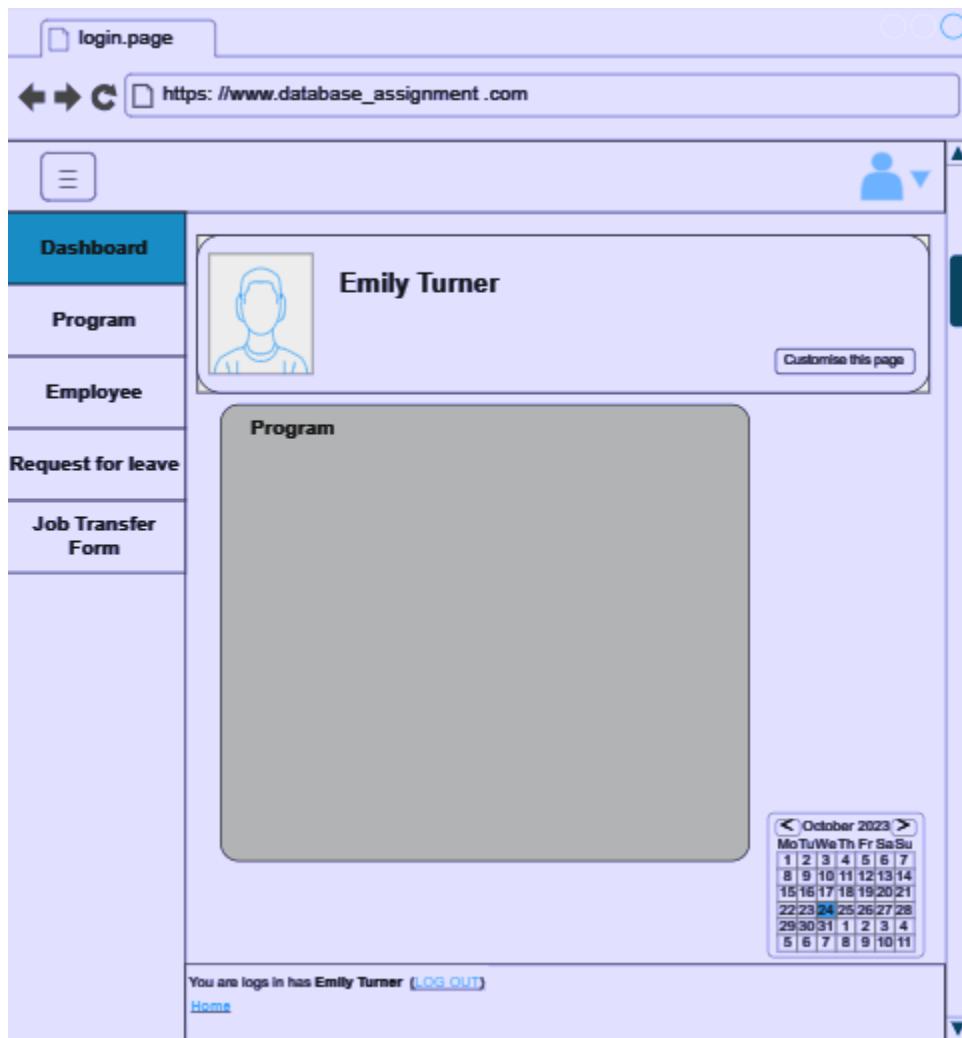


After entering their password and successfully logging in, the system will direct employees to a specially designed employee verification interface. This interface allows employees to view their name, employee ID, gender, position, and contact number, as well as their personal profile picture for further 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 screenshot shows a web application interface. At the top, there's a header bar with a back/forward button, a search icon, and the URL https://www.database_assignment.com. Below the header is a navigation sidebar on the left with the following menu items:

- Dashboard
- Program** (highlighted in blue)
- Employee
- Request for leave
- Job Transfer Form

The main content area has a title **Sales Performance Analysis**. It contains the following information:

Leader : Jane Smith
Start Date : 10th June 2023
End Date : 20th Aug 2023

Program Analysis

- Sales Revenue and Profit Analysis*
- Sales Channel Analysis*
- Customer Analysis*
- Sales Personnel Analysis*
- Effectiveness Analysis of Sales Activities*
- Sales Forecasting Analysis*
- Market Share Analysis*

At the bottom of the main content area, there's a message: "You are log in has Emily Turner ([LOG OUT](#))". Below that is a link to "Home".

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 screenshot shows a web browser window titled "employee.page" with the URL "https://www.database_assignment.com". The page has a sidebar on the left with links: Dashboard, Program, Employee (which is selected and highlighted in blue), Request for leave, and Job Transfer Form. The main content area is titled "Employee" and contains a table with columns: Employee ID, Employee Name, Gender, Phone Number, and Permission. The table lists 17 employees from ID 01 to 17. The permissions column shows mostly "General" except for Jessica Liu who is an "Administrator". Below the table is a navigation bar with links: << Prev 1 2 3 4 5 6 7 8 9 10 Next >>. At the bottom, there is a message: "You are log in has Emily Turner ([LOG OUT](#))" and a "Home" link.

Employee ID	Employee Name	Gender	Phone Number	Permission
01	John Doe	Male	011-12345678	General
02	Jane Smith	Female	012-98765432	General
03	Michael Johnson	Male	013-55555555	General
04	Emily Brown	Female	014-77777777	General
05	David Lee	Male	015-88888888	General
06	Sarah Wang	Female	016-99999999	General
07	Robert Chen	Male	017-66666666	General
08	Jessica Liu	Female	018-11111111	Administrator
09	William Zhang	Male	019-22222222	General
10	Linda Wu	Female	010-44444444	General
11	Alex Johnson	Male	01X-12345678	General
12	Ella Williams	Female	01X-98765432	General
13	Noah Smith	Male	01X-56789012	General
14	Ava Davis	Female	01X-24681357	General
15	Liam Taylor	Male	01X-11223344	General
16	Mia Brown	Female	01X-55443322	General
17	Jameson Lee	Male	01X-98761234	General

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)

The screenshot shows a web browser window with a light blue header bar. The address bar displays the URL https://www.database_assignment.com. Below the header, there's a user profile icon and a menu icon. The main content area has a light gray background and features a title "Employee Job Transfer Request Form". Inside this title, there are seven input fields: "Name", "Employee ID", "Current Position", "Target Position", "Current Job Description", "Target Job Description", and "Reason for Switching". Each field is followed by a black redacted box. At the bottom right of the form is a blue "Submit" button.

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 screenshot shows a web browser window with a light blue header bar. In the header, there is a 'Request For ..' button, a refresh/circular arrow icon, and a URL field containing 'https://www.database_assignment.com'. Below the header is a navigation bar with a list icon and a user profile icon. The main content area is titled 'Request For Leave' in bold. It contains several input fields: 'Name' (text input), 'Positions' (text input), 'Type Of Leave' (dropdown menu set to 'Option 1'), 'Start Date' (text input with a calendar icon), 'End Date' (text input with a calendar icon), 'Reason Of Leave' (large text area), and 'Contact' (text input). At the bottom right of the form is a blue 'Submit' button.

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.