

جامعة الامير سلطان
PRINCE SULTAN
U N I V E R S I T Y

Prince Sultan University
Department of Computer & Information Sciences

Introduction to Database Systems Project
Submitted to Ms. Roohi Jan

PSU Employee Payroll System

Prepared by:
Tala Alhazmi 220410073
Aljohara Alzaage 220410269
Dana Almashrafy 221410341
Reema BinFoiz 220410508
Sarah Alojayan 219410238

Table Of Contents:

Phase 1: Describing the Information System to be Developed	2
Introduction:	2
Background	2
Description of the Institution	2
Purpose and Scope of the System	2
User Interface and Prototype	3
Functional and Non-Functional Requirements:	4
Functional Requirements:	4
Non-functional Requirements:	5
Tables:	7
Entity Definition Table:	7
Employee	7
Department	7
Job	7
Faculty	8
Administrator	8
Rank	8
Full-Time Faculty	8
Part-Time Faculty	8
Leave	9
PaySlip	9
Allowance	9
Attribute Definition Table:	10
Department	12
Deduction	15
Allowance	15
Relation Definition Table	16
Prototype	17
Teamwork Distribution and Strategy	

Phase 1: Describing the Information System to be Developed

Introduction:

This phase aims to provide a detailed description of the employee payroll project. This document will cover the scope, purpose, background, organization description, data requirements, and non-functional requirements of the project. Additionally, it includes the entity, attributes, and relation definition tables.

Background:

Prince Sultan University is an academic institution that values excellence and development above all else. As stakeholders and end-users, PSU is highly invested in the development and implementation of this system. The PSU Employee Payroll System will facilitate a quick, easy, and user-friendly program dedicated to efficiently manage and produce payslips to all employees at PSU. As an institution, PSU is committed to fostering a dynamic and innovative learning environment, and the implementation of a payroll system aligns with the university's goal of efficient and effective administrative processes to support its academic mission to "provide the Middle East with quality education to the highest international standards".

Description of the Institution:

Prince Sultan University (PSU) is a private university located in Riyadh, Saudi Arabia. PSU offers both undergraduate and graduate programs in numerous fields. Since its establishment in 1998, PSU has grown exponentially and has become one of the world's leading academic institutions by providing high-quality education across six esteemed colleges, such as: Architecture & Design, Business Administration, Computer & Information Sciences, Engineering, Humanities and Sciences, and Law.

Purpose and Scope of the System:

Our goal is to implement a system that would fully automate the salary process. This system will allow administrators to streamline the payroll and human resources processes at PSU, increasing efficiency, improving accuracy, and maintaining transparency in managing employee compensation.

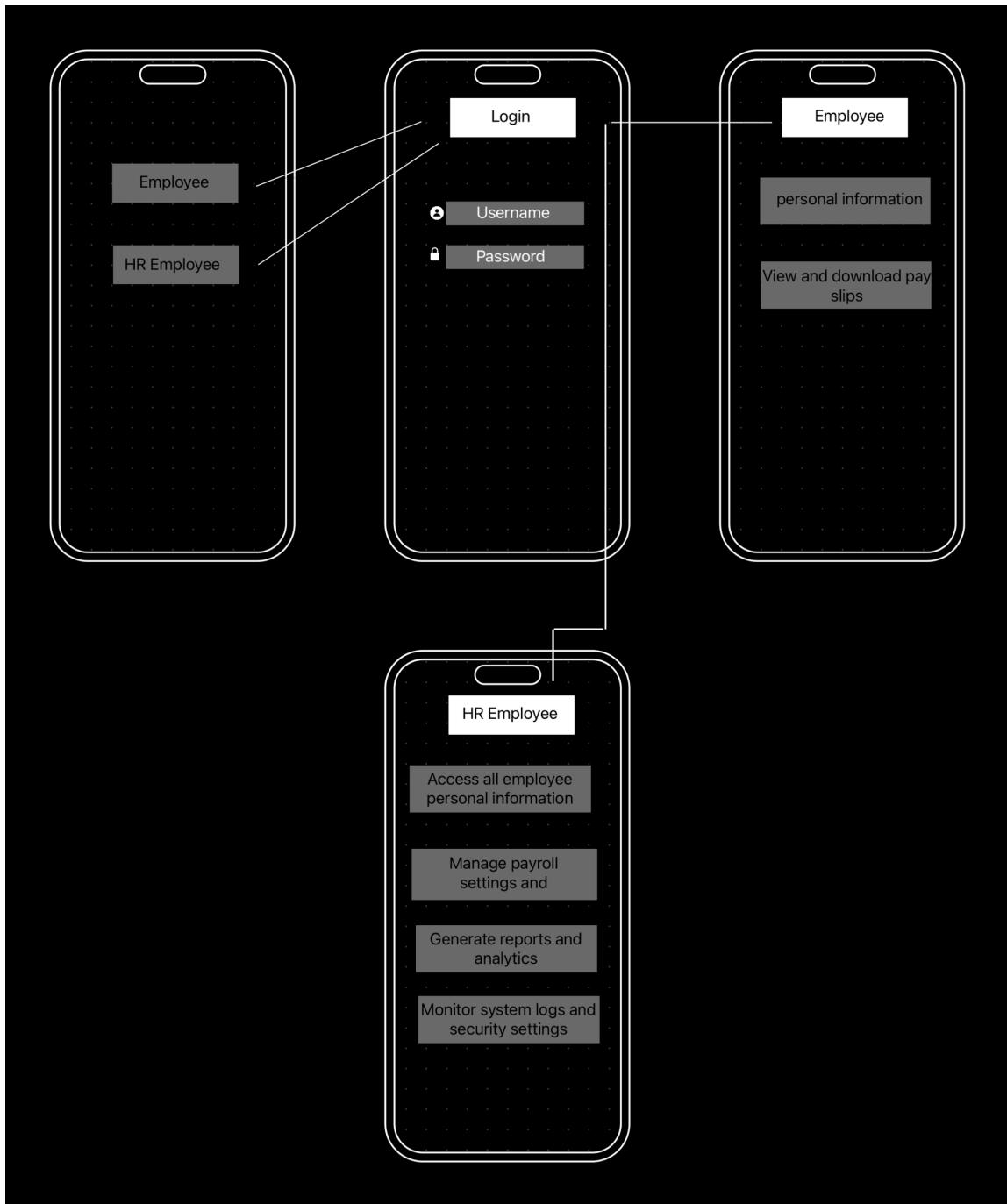
The system will manage payroll calculations, deductions, and benefits. Leave management will be included to handle various types of leaves. Integration with other PSU systems will be considered for data consistency.

User Interface and Prototype:

The system will cover all PSU employees, including full and part-time faculty. Our accessible user interface will be tailored to those accessing the system, implementing multiple views, such as:

- Employee View:
 - Login Page
 - Access to personal information.
 - View and download pay slips.
- HR Administrator View:
 - Login Page
 - Access to all employee information.
 - Manage payroll settings and calculations.
 - Generate reports and analytics.
 - Monitor system logs and security settings.

Prototype:



Functional and Non-Functional Requirements:

Functional Requirements:

User:

- The system shall allow the user to create a new account using the university email and password.
- The system shall allow for authentication and authorization by implementing access control, secure login credentials, and two-factor authentication.
- The system shall ask permission to capture and store employee details, including personal information, contact details, and job-related data.
- The system shall allow employees to update their personal information.
- The system shall calculate and manage salary details based on attendance, overtime, and other relevant factors.
- The system shall handle deductions, bonuses, and allowances.
- The system shall generate and store pay slips for each pay period (every month).
- The system shall record and monitor employee attendance.
- The system shall track and manage different types of leaves (sick leave, annual leave, etc.).

Administrator:

- The system shall allow the administrator to create a new account using the university email and password.
- The system shall allow for authentication and authorization by implementing access control, secure login credentials, and two-factor authentication.
- The system shall allow access to employee information.
- The system shall generate reports on payroll, attendance, and employee information.
- The system shall integrate with other PSU systems to ensure data consistency and accuracy.
- The system shall implement security measures to protect sensitive data.
- The system shall maintain logs for system activities.

Non-functional Requirements:

Performance:

- The system should handle simultaneous access by multiple users without significant degradation in performance.
- Response times for user interactions, especially during peak usage hours, should not exceed 3 seconds.

Scalability:

- The system should be scalable to accommodate the growing number of employees and any future expansions of the university.
- It should be capable of handling a minimum of 20% increase in user load within a year.

Reliability:

- The system should have a minimum uptime of 99.9% to ensure uninterrupted access for users.
- In case of unexpected system failures, there should be a backup and recovery mechanism to minimize data loss and downtime.

Security:

- The system should comply with industry-standard security practices and regulations to ensure the confidentiality and integrity of sensitive payroll and employee data.
- Access to the system should be role-based, with different levels of access granted to administrators, managers, and employees.

Usability:

- The user interface should be intuitive and user-friendly, requiring minimal training for administrators and end-users.
- The system should support multiple languages to accommodate the diverse linguistic preferences within the PSU community.

Data Requirements:

1. Each employee must have a unique EmployeeID, Fname, Lname, DateOfBirth, gender, contact information (email and phone number), UserType, HireDate, YearsOfExperience, and a unique NationalID or IqamaNumber among all faculties.
2. Each employee is assigned to a specific department, with each department identified by a unique DepartmentID and name.
3. An employee may be associated with at most one department, while one department can accommodate multiple employees.
4. Employees are faculty and administrators. Faculty members possess teaching experience and a rank.
5. Certain administrators, known as HR administrators have additional responsibilities such as controlling payslip processing and generating reports.
6. Administrators hold multiple job positions over several years, with their joining date and SalaryPerDay documented.
7. Each job is described by the JobID, JobTitle, BaseSalary, and NumberOfPaidleaves.
8. Faculty members can be either full-time or part-time.
9. Full-time faculty members have an academic rank, which is the same for many faculties.
10. A rank is defined by RankID, RankName, and BaseSalary.
11. Part-time faculty members are described by NumberOfTeachingHours_PerDay and Hourly_Rate. Attendance is marked by part-time faculty members every day, and recording the number of working hours. Each is identified by an AttendanceID, Date, StartTime, EndTime, and NumberOfHours (a derived attribute).
12. Employees have multiple leaves. It includes a LeaveID, LeaveType, LeaveDescription.

13. An employee avails of many leaves. A leave is availed by many employees. The start date, end date, and duration is stored (relationship attributes)
14. PaySlip manages details related to payslip processing for each employee. It must include a unique PaySlipID with a name or description, StartDate, EndDate, BaseSalary, GrossSalary, Deductions, Allowance, and NetSalary. A new Payslip ID is generated for each employee every month.
15. An employee can have multiple payslips, but each payslip record belongs to only one employee.
16. Employees can receive one or more allowances. Each allowance is received by more employees, with each allowance described by a unique AllowanceID and AllowanceType, AllowanceDescription. The allowance amount and month are recorded for each employee. (Note: This is an aggregation shown as Employee-allowance)
17. Employees can receive one or more deductions. Each deduction is received by many employees, with each deduction described by a unique DeductionID and DeductionType, Deduction Description. The deduction amount and month are recorded for each employee (relationship attributes). (Note: This is an aggregation shown as Employee-deduction)
18. A payslip includes one or many employee-allowance . An employee-allowance belongs to one payslip.
19. A payslip includes one or many employee-deduction. An employee-deduction belongs to one payslip

Tables:

Entity Definition Table:

Entity Name	Attributes	Description
Employee	EmployeeID	Unique identifier for each employee
	fname	First name of the employee
	lname	Last name of the employee
	gender	Gender of the employee
	dt_birth	Date of birth of the employee
	hire_date	Date when the employee was hired
	phone_number	Contact phone number of the employee

	email_address	Email address of the employee
	user_type	Type of user (e.g., faculty, administrator)
	Yrs_of_Experience	Years of experience of the employee
	NatID	Unique identifier among all faculty members
Department	DepartmentID	Unique identifier for each department
	name	Name of the department
Job	JobID	Unique identifier for each job
	job_title	Title or name of the job
	base_salary	Base salary for the job
	num_paid_leaves	Number of paid leaves allocated for the job
Faculty	EmployeeID	Key referencing EmployeeID in the Employee table
	teaching_experience	Experience in teaching
	rankID	Key referencing RankID in the Rank table
Administrator	EmployeeID	Key referencing EmployeeID in the Employee table
	is_HR_administrator	Indicates if the administrator is an HR administrator
	salary_per_day	Daily salary for the administrator
Rank	RankID	Unique identifier for each rank
	rank_name	Name or title of the rank
	base_salary	Base salary for the rank
Full-Time Faculty	EmployeeID	Key referencing EmployeeID in the Employee table
	academic_rankID	Key referencing RankID in the Rank table
Part-Time Faculty	EmployeeID	Key referencing EmployeeID in the Employee table
	teaching_hours_per_day	Number of teaching hours per day for the part-time faculty
	hourly_rate	Hourly rate for the part-time faculty
Leave	LeaveID	Unique identifier for each leave
	EmployeeID	Key referencing EmployeeID in the Employee table
	leave_type	Type of leave (e.g., vacation, sick leave)
	start_date	Start date of the leave
	end_date	End date of the leave
	duration	Duration of the leave in days
PaySlip	PaySlipID	Unique identifier for each payslip

	EmployeeID	Key referencing EmployeeID in the Employee table
	start_date	Start date of the payslip
	end_date	End date of the payslip
	base_salary	Base salary for the payslip
	allowance_amount	Amount of allowances for the payslip
	deduction_amount	Amount of deductions for the payslip
	gross_salary	Gross salary for the payslip
	net_salary	Net salary for the payslip
	date_created	Date when the payslip was created
Allowance	AllowanceID	Unique identifier for each allowance
	AllowanceType	Type of allowance received by employees
	allowance_amount	Amount of the allowance
	effectiveDate	Effective date of the allowance for the employee

Attribute Definition Table:

Entity	Attributes	Type	Description
	Name	Composite attribute (FName, LName) String	Describes the full name of the entity.
	BirthDate	Simple attribute Integer	The date of birth of the employee.
	Gender	Simple attribute String	The gender of the employee.
	Username	Simple attribute String	Describes the username of the entity.

<i>Employee</i>	NatID	Simple key attribute Integer	Describes the national ID of the user.
	PhoneNo	Simple attribute Integer	Contains the phone number of the entity.
	EmailAddress	Simple attribute String	Describes the email address of the entity.
	Address	Simple attribute String	The residential address of the employee.
	EmployeeType	Simple attribute String	Indicates whether the employee is teaching or non-teaching
	TeachingExperienceAllowance	Simple attribute Integer	Allowance specific to teaching employees based on experience.
	DependentsAllowancePercent	Simple Attribute Double	Percentage for allowances based on dependents (school, health, research, living)
	EmpID	Simple key attribute Integer	Describes the employee number of the entity.

	JoiningDate	Simple attribute Integer	the date on which an employee joined the organization
	DepartmentID (Foreign Key)	composite attribute Integer	Reference to the department assigned to the employee
	JobID (Foreign Key)	Composite attribute Integer	Reference to the job title or position held by the employee.
	PayrollID (Foreign Key)	Composite attribute Integer	Reference to the payroll associated with the employee.
Department	DepartmentID (Primary Key)	Simple key attribute Integer	Unique identifier for each department.
	DepartmentName	Simple attribute String	Descriptive name for the department.
Job	JobID (Primary Key)	Simple key attribute Integer	Unique identifier for each job title
	JobType	Simple Attribute String	Indicates whether the job is full time or part time
	HourlyRate (For part-time employees)	Simple Attribute Double	Hourly rate specific to part-time employees.
	PaidLeaveDays	Simple attribute Integer	Number of paid leave days associated with the job title.

	JobTitle	Simple attribute String	The official title or designation of the job.
	BaseSalary	Simple attribute Double	The base salary associated with the job title.
<i>Payroll</i>	PayrollID (Primary Key)	Simple key attribute Integer	Unique identifier for each payroll record.
	EmployeeID (Foreign Key)	Simple attribute Integer	Reference to the employee associated with the payroll.
	PayrollDate	Simple attribute Integer	The date for which the payroll is processed.
	Allowances	Simple attribute Integer	The total amount of allowances provided to the employee.
	GrossSalary	Derived attribute Double	Total salary before any deductions (BaseSalary + Allowances).
	Deductions	Simple attribute Double	The total amount of deductions applied to the salary.

	NetSalary	Derived attribute Double	The amount an employee receives after deductions (GrossSalary - Deductions).
<i>Deduction</i>	DeductionID (Primary Key)	Simple key attribute Integer	Unique identifier for each deduction record.
	PayrollID (Foreign Key)	Simple attribute Integer	Reference to the payroll associated with the deduction.
	DeductionType	Simple attribute String	Categorization or description of the type of deduction.
	Amount	Simple attribute Double	The monetary value or percentage of the employee's salary to be deducted.
<i>Allowance</i>	AllowanceID (Primary Key)	Simple key attribute Integer	Unique identifier for each allowance record.
	PayrollID (Foreign Key)	Simple attribute Integer	Reference to the payroll associated with the allowance.
	AllowanceType	Simple attribute String	Categorization or description of the type of allowance.

	Amount	Simple attribute Double	The monetary value or percentage of the employee's salary allocated as an allowance.
--	--------	----------------------------	--

Relation Definition Table :

Relationship Name	Type	Entities	Description with entities involved
Belongs_to	-Binary 1 to N -Binary 1 to M	Employees , department	each of employees and admins must belong to a specific department and each department can have many employees
	-Binary 1 to Many	faculty	faculty can have teaching experience as well as a rank.
has	-Binary 1 to Many	- Full-time faculty - Employees Leaves	a full time faculty can have a rank which is the same for many other faculties. Employees have many leaves
Receives	-Binary 1 to N -Binary 1 to many -Binary 1 to M	- Employees -pay slips -Allowances	Employees receive many pay slips and each pay slip goes for one employee. Employees receive one or many allowances
Have	Binary 1 to M	Employees, Leaves	Employees can have multiple leaves

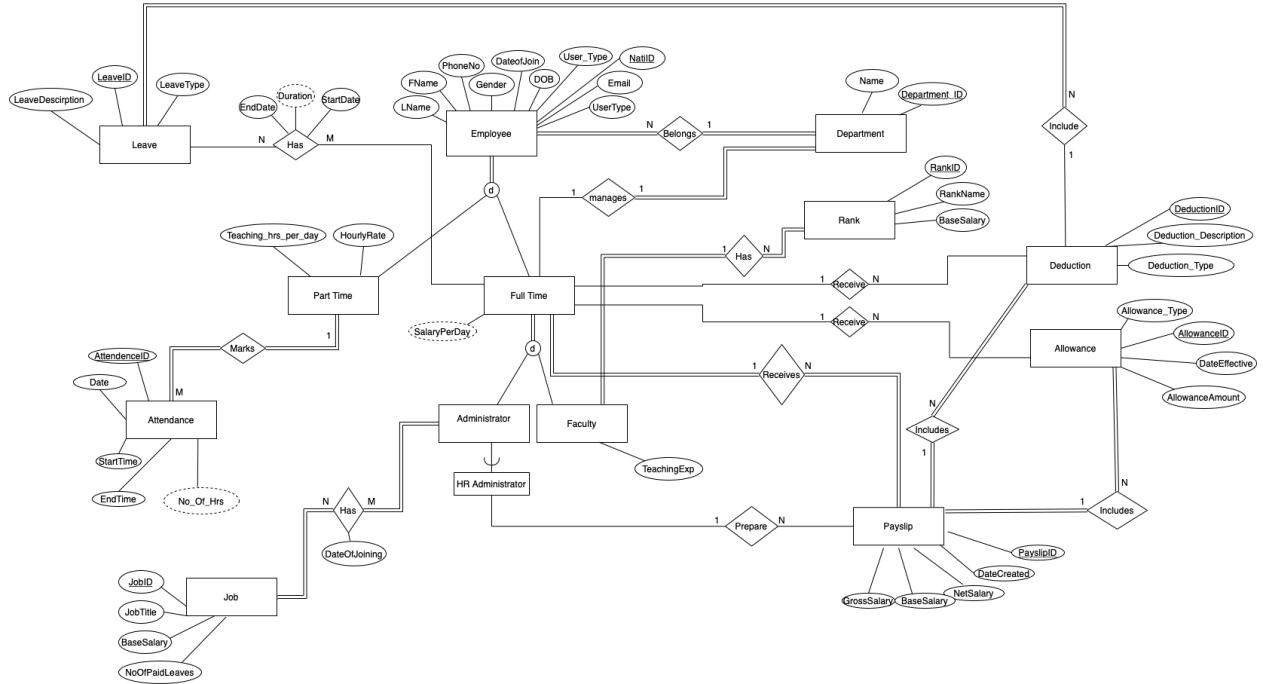
Teamwork Distribution and Strategy:

This table describes the teamwork distribution for every team member.

Team Member Name	Individual Contribution
Tala Alhazmi	Description of the Institution Background Scope and UI Functional Requirements EER Entities and attributes Helped review table and diagram in phase 3 MySQL Table Creation
Aljohara Alzaage	Non-Functional Requirements Entity Definition Table Some EER diagram entities Data dictionary table
Dana Almashrafy	Data Requirements Some EER diagram entities Relational diagram MySQL Table Data Insertion
Reema BinFoiz	Introduction Attribute definition table EER diagram relationships and entities. Business Constraints Helped arrange attributes into dictionary table
Sarah Alojayan	Relation definition table, prototype, EER diagram relationships and entities. Business Constraints Helped arrange entities in relational diagram

Phase II

EER Diagram:



Business Constraints:

- A part-timer get paid only by hours worked unlike Full-timers
- Only full-timers get monthly allowance
- Deduction is done if an employee misses their work day
- High turnover rates can increase the workload for payroll administrators, requiring constant updates to employee records and payroll calculations.

Phase III

Employee

Entity Name	Key Type	Constraints	FK Table	FK Column	Data Type	Length
EmployeeID	PK	Not null, unique			Varchar	10
fname		Not null			Varchar	15
lname		Not null			Varchar	15
gender		Not null			Varchar	1
dt_birth		Not null			Date (dd-mm-yyyy)	
hire_date		Not null			Date (dd-mm-yyyy)	
phone_number		Not null			char	15
email_address		Not null			Varchar	50
user_type		Not null			Varchar	20
Yrs_of_Experience					int	
NatID					Varchar	20

Department

Entity Name	Key Type	Constraints	FK Table	FK Column	Data Type	Length
DepartmentID	PK	Not null, unique			Varchar	10
name		Not null			Varchar	15

Job

Entity Name	Key Type	Constraints	FK Table	FK Column	Data Type	Length
JobID	PK	Not null, unique			Varchar	
job_title		Not null			Varchar	50
base_salary					Number	
num_paid_leaves					Number	

Faculty

Entity Name	Key Type	Key Constraints	FK Table	FK Column	Data Type	Length
EmployeeID	PK	Not null, Unique	Employee	EmployeeID	Varchar	10
teaching_experience		Not null			Varchar	3
rankID		Not null, unique	Rank	RankID	Varchar	10

Administrator

ENTITY Name	Key Tpe	Key Constraints	FK Table	FK Column	Data Type	Length
EmployeeID	PK	UNnique, Not null	Employee	EmployeeID	Varchar	10

is_HR_administrator		Not null			Varchar	3
salary_per_day		Not null			Number	50

JobRank

ENTITY Name	Key type	Key Constraints	FK Table	FK Column	Data Type	Length
RankID	PK	Primary Key, not null			Varchar	10
rank_name	Name or title of the rank	Not null			Varchar	20
base_salary	Base salary for the rank	Not null			Number	

Full time Faculty :

Entity Name	Key type	Key Constraints	FK Table	FK Column	Data Type	Length
EmployeeID	PK	Primary , not null	Employee	EmployeeID	Varchar	10
academic_rankID		Not null	Rank	RankID	Varchar	10

Part Time Faculty:

Entity Name	Key Type	Constraints	FK Table	FK Column	Data Type	Length

EmployeeID	PK	Not Null, Unique	Employee	EmployeeID	varchar	10
teaching_hours_per_day		Not null			Varchar	3
hourly_rate		Not null			Varchar	10

LEAVE:

ENTITY NAME	KEY TYPE	KEY CONSTRAINTS	FK TAB KE	FK COLU MN	DATA TYPE	LENGTH
LeaveID	PK	Not Null,unique			Varchar	10
EmployeeID		Not null, unique	Employee	EmployeeID	varchar	10
leave_type		Not Null			VARCHAR	10
start_date		Not null			date	
end_date		Not null			date	
duration		Not null				

PaySlip

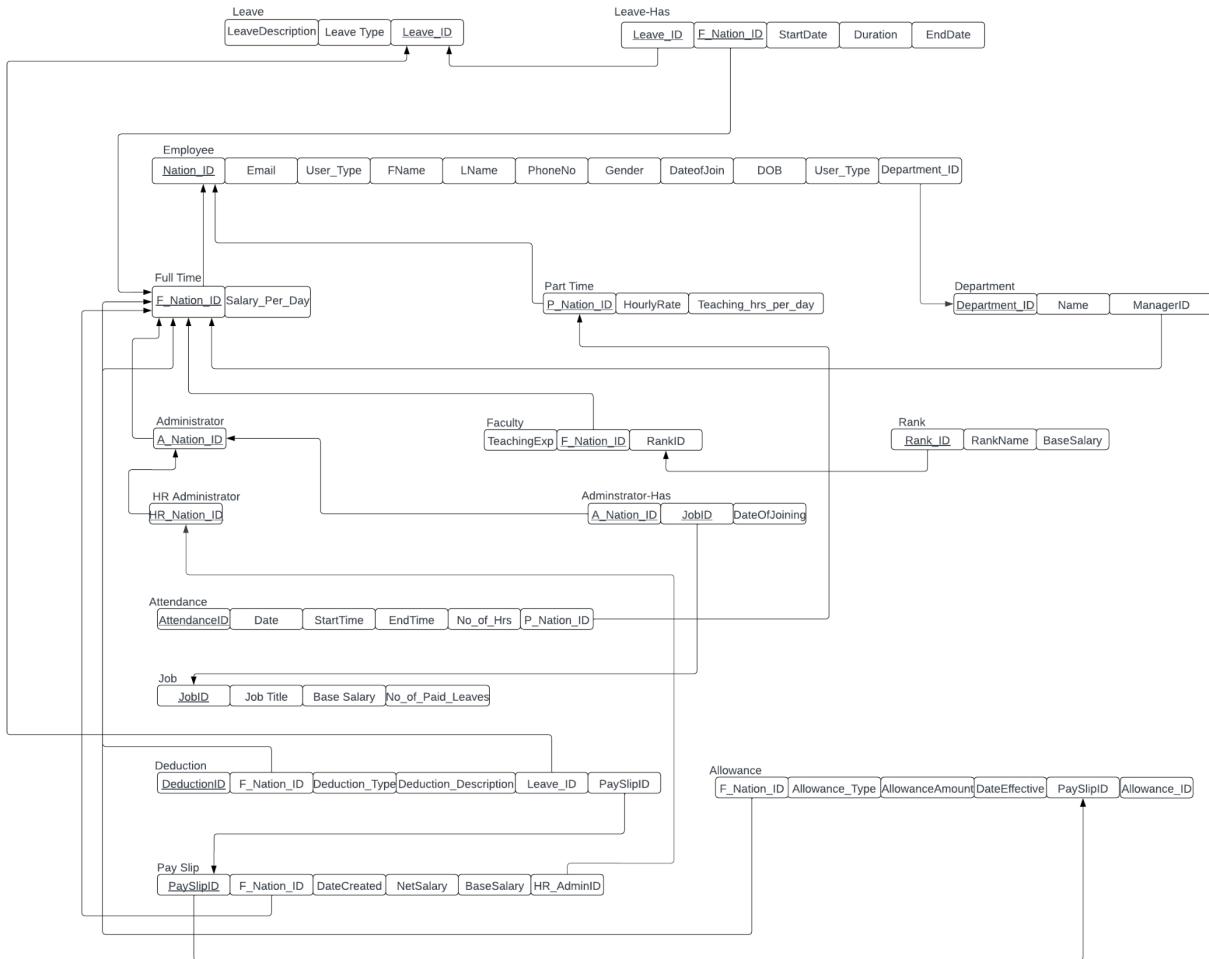
ENTITY Name	Key Type	Constraints	FK Table	FK Column	Data Type	Length
PaySlipID	Unique	Not Null			Varchar	10
EmployeeID		Not Null	Employee	EmployeeID	varchar	10
start_date		Not Null			date	
end_date		Not Null			date	
base_salary		Not Null			Varchar	10

allowance_amount		Not Null			Varchar	10
deduction_amount		Not Null			Varchar	10
gross_salary		Not Null			Varchar	10
net_salary		Not Null			Varchar	10
date_created		Not Null			Date	

Allowance

ENTITY Name	Key Type	Constraints	FK Table	FK Column	Data Type	Length
AllowanceID	PK	Not Null, unique			VARCHAR	10
AllowanceType		Not null			VARCHAR	20
allowance_amount		Not null			VARCHAR	15
effectiveDate		Not null			DATE	

Relational Diagram:



Code for Database Table Creation:

```
create schema PSUPayroll;
USE PSUPayroll;
-----
CREATE TABLE Department (
    departmentID VARCHAR(10) PRIMARY KEY,
    name VARCHAR(15) NOT NULL
);
-----
CREATE TABLE Job(
    jobID VARCHAR(10) PRIMARY KEY,
    job_title VARCHAR(50) NOT NULL,
    base_salary VARCHAR(50),
    num_paid_leaves INT
);
-----
CREATE TABLE JobRank(
    rankID VARCHAR(10) PRIMARY KEY,
    rank_name VARCHAR(20) NOT NULL,
    base_salary VARCHAR(50) NOT NULL
);
-----
CREATE TABLE Employee(
    employeeID VARCHAR(10) PRIMARY KEY,
    fname VARCHAR(15) NOT NULL,
    lname VARCHAR(15) NOT NULL,
    gender VARCHAR(1) NOT NULL,
    dt_birth DATE NOT NULL,
    hire_date DATE NOT NULL,
    phone_number CHAR(15) NOT NULL,
    email_address VARCHAR(50) NOT NULL,
    user_type VARCHAR(20) NOT NULL,
    yrs_of_experience INT,
    departmentID VARCHAR(10),
    FOREIGN KEY (departmentID) REFERENCES Department(departmentID)
    ON delete set null
);
```

```

CREATE TABLE FullTimeEmployee (
F_EmployeeID VARCHAR(10) PRIMARY KEY,
salary_per_day VARCHAR(50) NOT NULL,
FOREIGN KEY (F_EmployeeID) REFERENCES Employee(EmployeeID)
);

-----
ALTER TABLE Department
Add ManagerID VARCHAR(10);
ALTER TABLE department
add FOREIGN KEY (ManagerID) REFERENCES
FullTimeEmployee(F_EmployeeID);

-----
CREATE TABLE Faculty(
FacultyID VARCHAR(10) PRIMARY KEY,
teaching_experience INT NOT NULL,
rankID VARCHAR(10) UNIQUE NOT NULL,
FOREIGN KEY (facultyID) REFERENCES
FulltimeEmployee(F_EmployeeID),
FOREIGN KEY (rankID) REFERENCES JobRank(rankID)
);

-----
-----

CREATE TABLE PartTimeEmployee (
P_EmployeeID VARCHAR(10) PRIMARY KEY,
teaching_hours_per_day VARCHAR(3) NOT NULL,
hourly_rate VARCHAR(10) NOT NULL,
FOREIGN KEY (P_EmployeeID) REFERENCES Employee(employeeID)
);

CREATE TABLE Attendance (
AttendanceID VARCHAR(10) PRIMARY KEY,
AttendanceDate DATE,
StartTime VARCHAR(10),
EndTime VARCHAR(10),
No_of_Hours INT,
P_EmployeeID VARCHAR(10),
FOREIGN KEY (P_EmployeeID) REFERENCES
PartTimeEmployee(P_EmployeeID)
);

```

```

CREATE TABLE Administrator(
AdminID VARCHAR(10) PRIMARY KEY,
FOREIGN KEY (AdminID) REFERENCES FulltimeEmployee(F_EmployeeID)
);

-----
CREATE TABLE Administrator_Job
(
AdminID VARCHAR(10),
jobID VARCHAR(10),
PRIMARY KEY(AdminID,jobID),
FOREIGN KEY (AdminID) REFERENCES Administrator(AdminID),
FOREIGN KEY (JobID) REFERENCES Job(JobID)
);
-----

CREATE TABLE ELeave
(
leaveID VARCHAR(20) PRIMARY KEY,
leave_type VARCHAR(20) NOT NULL,
LeaveDesc VARCHAR(60)
);
-----

CREATE TABLE PaidLeave
(
leaveID VARCHAR(10),
employeeID VARCHAR(10) UNIQUE NOT NULL,
start_date DATE NOT NULL,
end_date DATE NOT NULL,
duration INT NOT NULL,
PRIMARY KEY (leaveID, EmployeeID),
FOREIGN KEY (employeeID) REFERENCES Employee(employeeID),
FOREIGN KEY (LeaveID) REFERENCES ELeave(LeaveID)
);
-----

CREATE TABLE PaySlip (
paySlipID VARCHAR(10) PRIMARY KEY,
F_EmployeeID VARCHAR(10) UNIQUE NOT NULL,
base_salary VARCHAR(10) NOT NULL,

```

```

gross_salary VARCHAR(10) NOT NULL,
net_salary VARCHAR(10) NOT NULL,
date_created DATE NOT NULL,
ApprovedBy VARCHAR(10),
FOREIGN KEY (F_EmployeeID) REFERENCES
FullTimeEmployee(F_EmployeeID),
FOREIGN KEY (ApprovedBy) REFERENCES Administrator(AdminID)
);
CREATE TABLE Allowance (
allowanceID VARCHAR(10) PRIMARY KEY,
allowanceType VARCHAR(20) NOT NULL,
allowance_amount VARCHAR(15) NOT NULL,
effectiveDate DATE NOT NULL,
PaySlipID VARCHAR(10),
FOREIGN KEY (PaySlipID) REFERENCES payslip(payslipID)
);
-----
CREATE TABLE Deduction (
DeductionID VARCHAR(10) PRIMARY KEY,
DeductionType VARCHAR(20) NOT NULL,
DeductionDesc VARCHAR(60) NOT NULL,
PaySlipID VARCHAR(10),
FOREIGN KEY (PaySlipID) REFERENCES payslip(payslipID)
);

```

Code for Record Insertion:

```

-- Inserting records into Department
INSERT INTO Department (departmentID, name) VALUES ('D001',
'HR');
INSERT INTO Department (departmentID, name) VALUES ('D002',
'Finance');
INSERT INTO Department (departmentID, name) VALUES ('D003',
'IT');
INSERT INTO Department (departmentID, name) VALUES ('D004',
'Marketing');
INSERT INTO Department (departmentID, name) VALUES ('D005',
'Sales');

```

```

INSERT INTO Department (departmentID, name) VALUES ('D006',
'Legal');
INSERT INTO Department (departmentID, name) VALUES ('D007',
'Operations');
INSERT INTO Department (departmentID, name) VALUES ('D008',
'Research');
INSERT INTO Department (departmentID, name) VALUES ('D009',
'Development');
INSERT INTO Department (departmentID, name) VALUES ('D010',
'Support');

```

The screenshot shows the MySQL Workbench interface with the 'pupayroll' database selected. The 'department' table is open, displaying its structure and data. The table has three columns: departmentID, name, and ManagerID. The data shows ten departments: HR, Finance, IT, Marketing, Sales, Legal, Operations, Research, Development, and Support.

departmentID	name	ManagerID
D001	HR	
D002	Finance	
D003	IT	
D004	Marketing	
D005	Sales	
D006	Legal	
D007	Operations	
D008	Research	
D009	Development	
D010	Support	

```

-- Inserting records into Job
INSERT INTO Job (jobID, job_title, base_salary, num_paid_leaves)
VALUES ('J001', 'Instructor', '60000', 20);
INSERT INTO Job (jobID, job_title, base_salary, num_paid_leaves)
VALUES ('J002', 'Analyst', '50000', 15);
INSERT INTO Job (jobID, job_title, base_salary, num_paid_leaves)
VALUES ('J003', 'Head of CS', '70000', 20);
INSERT INTO Job (jobID, job_title, base_salary, num_paid_leaves)
VALUES ('J004', 'Administrator', '55000', 15);
INSERT INTO Job (jobID, job_title, base_salary, num_paid_leaves)
VALUES ('J005', 'Assistant', '65000', 20);
INSERT INTO Job (jobID, job_title, base_salary, num_paid_leaves)
VALUES ('J006', 'Instructor', '72000', 18);
INSERT INTO Job (jobID, job_title, base_salary, num_paid_leaves)
VALUES ('J007', 'Instructor', '48000', 12);

```

```
INSERT INTO Job (jobID, job_title, base_salary, num_paid_leaves)
VALUES ('J008', 'Administrator', '62000', 16);
INSERT INTO Job (jobID, job_title, base_salary, num_paid_leaves)
VALUES ('J009', 'Supervisor', '40000', 10);
INSERT INTO Job (jobID, job_title, base_salary, num_paid_leaves)
VALUES ('J010', 'Assistant', '35000', 8);

--  
INSERT INTO JobRank (rankID, rank_name, base_salary) VALUES
('R001', 'Junior', '40000');
INSERT INTO JobRank (rankID, rank_name, base_salary) VALUES
('R002', 'Mid-Level', '60000');
INSERT INTO JobRank (rankID, rank_name, base_salary) VALUES
('R003', 'Senior', '80000');
INSERT INTO JobRank (rankID, rank_name, base_salary) VALUES
('R004', 'Lead', '90000');
INSERT INTO JobRank (rankID, rank_name, base_salary) VALUES
('R005', 'Principal', '100000');
INSERT INTO JobRank (rankID, rank_name, base_salary) VALUES
('R006', 'Chief', '120000');
INSERT INTO JobRank (rankID, rank_name, base_salary) VALUES
('R007', 'Executive', '150000');
INSERT INTO JobRank (rankID, rank_name, base_salary) VALUES
('R008', 'Assistant', '30000');
INSERT INTO JobRank (rankID, rank_name, base_salary) VALUES
('R009', 'Associate', '50000');
INSERT INTO JobRank (rankID, rank_name, base_salary) VALUES
('R010', 'Director', '110000');
INSERT INTO JobRank (rankID, rank_name, base_salary) VALUES
('R011', 'Spot', '10000');
```

MySQL Workbench

Schemas

- PSUPayroll (psupayroll) ×
- psupayroll ×
- PSUPayroll ×

File Edit View Query Database Server Tools Scripting Help

Navigator:

PSUpayroll SQL File 1* fulltimeemployee parttimeemployee administrator department job jobrank

Unit to 1000 rows

1 • `SELECT * FROM psupayroll.jobrank;`

Result Grid

rankID	rank_name	base_salary
R001	Junior	40000
R002	Mid-Level	60000
R003	Senior	80000
R004	Manager	90000
R005	Principal	100000
R006	Chef	120000
R007	Executive	150000
R008	Assistant	30000
R009	Associate	50000
R010	Director	110000
R011	Spot	10000
R009	Other	0000

jobrank ×

Action Output

- 330 16:08:57 SELECT e.fname, e.lname, p.base_salary FROM Employee e JOIN PaySlip p ON employeeID = p.employeeID ORDER BY p.base_salary DESC LIMIT 0, 1000 10 row(s) returned 0.000 sec / 0.000 sec
- 331 16:09:29 SELECT e.employeeID, e.fname, e.lname, pl.* FROM Employee e JOIN PaidLeave pl ON e.employeeID = pl.employeeID WHERE pl.start_date > (SELECT start_date FROM PaidLeave WHERE employeeID = e.employeeID) AND pl.end_date < (SELECT end_date FROM PaidLeave WHERE employeeID = e.employeeID) 10 row(s) returned 0.000 sec / 0.000 sec
- 332 16:10:24 SELECT * FROM psupayroll.administrator LIMIT 0, 1000 10 row(s) returned 0.000 sec / 0.000 sec
- 333 16:11:12 SELECT * FROM psupayroll.department LIMIT 0, 1000 10 row(s) returned 0.000 sec / 0.000 sec
- 334 16:11:29 SELECT * FROM psupayroll.job LIMIT 0, 1000 10 row(s) returned 0.000 sec / 0.000 sec
- 335 16:11:48 SELECT * FROM psupayroll.jobrank LIMIT 0, 1000 11 row(s) returned 0.000 sec / 0.000 sec

--

```

INSERT INTO Employee (employeeID, fname, lname, gender,
dt_birth, hire_date, phone_number, email_address, user_type,
yrs_of_experience, departmentID) VALUES ('E001', 'Abdullah',
'Al-Saud', 'M', '1980-05-15', '2010-04-01', '0501234567',
'abdullah.alsaud@example.com', 'FullTime', 10, 'D001');
INSERT INTO Employee (employeeID, fname, lname, gender,
dt_birth, hire_date, phone_number, email_address, user_type,
yrs_of_experience, departmentID) VALUES ('E002', 'Fatimah',
'Al-Faisal', 'F', '1985-06-20', '2012-05-01', '0502345678',
'fatimah.alfaisal@example.com', 'FullTime', 8, 'D002');
INSERT INTO Employee (employeeID, fname, lname, gender,
dt_birth, hire_date, phone_number, email_address, user_type,
yrs_of_experience, departmentID) VALUES ('E003', 'Mohammed',
'Al-Otaibi', 'M', '1990-07-25', '2014-06-01', '0503456789',
'mohammed.alotaibi@example.com', 'PartTime', 6, 'D003');
INSERT INTO Employee (employeeID, fname, lname, gender,
dt_birth, hire_date, phone_number, email_address, user_type,
yrs_of_experience, departmentID) VALUES ('E004', 'Noura',

```

```
'Al-Rashid', 'F', '1975-08-30', '2008-07-01', '0504567890',  
'noura.alrashid@example.com', 'FullTime', 15, 'D004');  
INSERT INTO Employee (employeeID, fname, lname, gender,  
dt_birth, hire_date, phone_number, email_address, user_type,  
yrs_of_experience, departmentID) VALUES ('E005', 'Yousef',  
'Al-Mutairi', 'M', '1982-09-15', '2011-08-01', '0505678901',  
'yousef.almuitairi@example.com', 'PartTime', 9, 'D005');  
INSERT INTO Employee (employeeID, fname, lname, gender,  
dt_birth, hire_date, phone_number, email_address, user_type,  
yrs_of_experience, departmentID) VALUES ('E006', 'Huda',  
'Al-Qahtani', 'F', '1988-10-20', '2013-09-01', '0506789012',  
'huda.alqahtani@example.com', 'FullTime', 7, 'D006');  
INSERT INTO Employee (employeeID, fname, lname, gender,  
dt_birth, hire_date, phone_number, email_address, user_type,  
yrs_of_experience, departmentID) VALUES ('E007', 'Saud',  
'Al-Ghamdi', 'M', '1978-11-25', '2009-10-01', '0507890123',  
'saud.alghamdi@example.com', 'FullTime', 12, 'D007');  
INSERT INTO Employee (employeeID, fname, lname, gender,  
dt_birth, hire_date, phone_number, email_address, user_type,  
yrs_of_experience, departmentID) VALUES ('E008', 'Mona',  
'Al-Harbi', 'F', '1986-12-30', '2015-11-01', '0508901234',  
'mona.alharbi@example.com', 'PartTime', 5, 'D008');  
INSERT INTO Employee (employeeID, fname, lname, gender,  
dt_birth, hire_date, phone_number, email_address, user_type,  
yrs_of_experience, departmentID) VALUES ('E009', 'Saleh',  
'Al-Subaie', 'M', '1992-01-05', '2016-12-01', '0509012345',  
'saleh.alsubaie@example.com', 'FullTime', 4, 'D009');  
INSERT INTO Employee (employeeID, fname, lname, gender,  
dt_birth, hire_date, phone_number, email_address, user_type,  
yrs_of_experience, departmentID) VALUES ('E010', 'Aisha',  
'Al-Mansour', 'F', '1995-02-10', '2017-01-01', '0500123456',  
'aisha.almansour@example.com', 'PartTime', 3, 'D010');  
INSERT INTO Employee (employeeID, fname, lname, gender,  
dt_birth, hire_date, phone_number, email_address, user_type,  
yrs_of_experience, departmentID) VALUES ('E011', 'Hassan',  
'Al-Juhani', 'M', '1987-03-22', '2018-01-01', '0501234566',  
'hassan.aljuhani@example.com', 'PartTime', 2, 'D001');  
INSERT INTO Employee (employeeID, fname, lname, gender,  
dt_birth, hire_date, phone_number, email_address, user_type,
```

```

yrs_of_experience, departmentID) VALUES ('E012', 'Laila',
'Al-Zahrani', 'F', '1987-04-12', '2011-03-01', '0502233444',
'laila.alzahrani@example.com', 'FullTime', 11, 'D002');

INSERT INTO Employee (employeeID, fname, lname, gender,
dt_birth, hire_date, phone_number, email_address, user_type,
yrs_of_experience, departmentID) VALUES ('E013', 'Ahmed',
'Al-Shammari', 'M', '1989-05-15', '2012-04-01', '0501122334',
'ahmed.alshammari@example.com', 'FullTime', 10, 'D003');

INSERT INTO Employee (employeeID, fname, lname, gender,
dt_birth, hire_date, phone_number, email_address, user_type,
yrs_of_experience, departmentID) VALUES ('E014', 'Rania',
'Al-Shehri', 'F', '1991-06-18', '2013-05-01', '0502233445',
'rania.alshehri@example.com', 'PartTime', 9, 'D004');

INSERT INTO Employee (employeeID, fname, lname, gender,
dt_birth, hire_date, phone_number, email_address, user_type,
yrs_of_experience, departmentID) VALUES ('E015', 'Sami',
'Al-Anazi', 'M', '1980-07-21', '2014-06-01', '0503344556',
'sami.alanazi@example.com', 'FullTime', 8, 'D005');

```

The screenshot shows the MySQL Workbench interface with the following details:

- Top Bar:** File, Edit, View, Query, Database, Server, Tools, Scripting, Help.
- Schemas:** Schemas dropdown, currently set to 'pu payroll'.
- Tables:** Tables dropdown, currently set to 'employee'.
- SQL Editor:** Contains the SQL query: `SELECT * FROM pu payroll.employee;`
- Result Grid:** Displays the data from the 'employee' table. The columns are: employeeID, fname, lname, gender, dt_birth, hire_date, phone_number, email_address, user_type, yrs_of_experience, and departmentID. The data includes rows for employees E001 through E015.
- Output Tab:** Shows the execution history of queries. The last few entries are:
 - 331 16:11:29 SELECT * FROM Employee e JOIN PaidLeave pl ON employeeID = e.employeeID WHERE pl.start_date = (SELECT start_date FROM Employee WHERE employeeID = 1)
 - 332 16:10:24 SELECT * FROM pu payroll.administrator LIMIT 0, 1000
 - 333 16:11:12 SELECT * FROM pu payroll.department LIMIT 0, 1000
 - 334 16:11:29 SELECT * FROM pu payroll.job LIMIT 0, 1000
 - 335 16:11:48 SELECT * FROM pu payroll.jobrank LIMIT 0, 1000
 - 336 16:12:12 SELECT * FROM pu payroll.employee LIMIT 0, 1000
- System Status:** Shows the system status at the bottom right: 41°C, ملخص, 4:12 PM, ENG, 5/20/2024.

```

-- --
INSERT INTO FullTimeEmployee (F_EmployeeID, salary_per_day)
VALUES ('E001', '200');
INSERT INTO FullTimeEmployee (F_EmployeeID, salary_per_day)
VALUES ('E002', '180');
INSERT INTO FullTimeEmployee (F_EmployeeID, salary_per_day)
VALUES ('E004', '210');
INSERT INTO FullTimeEmployee (F_EmployeeID, salary_per_day)
VALUES ('E006', '190');
INSERT INTO FullTimeEmployee (F_EmployeeID, salary_per_day)
VALUES ('E007', '220');
INSERT INTO FullTimeEmployee (F_EmployeeID, salary_per_day)
VALUES ('E009', '160');
INSERT INTO FullTimeEmployee (F_EmployeeID, salary_per_day)
VALUES ('E010', '150');
INSERT INTO FullTimeEmployee (F_EmployeeID, salary_per_day)
VALUES ('E012', '230');
INSERT INTO FullTimeEmployee (F_EmployeeID, salary_per_day)
VALUES ('E013', '175');
INSERT INTO FullTimeEmployee (F_EmployeeID, salary_per_day)
VALUES ('E015', '205');

```

The screenshot shows the MySQL Workbench interface with the following details:

- Schemas:** The 'psupayroll' schema is selected, displaying tables like administrator, department, job, jobrank, and employee.
- Tables:** The 'fulltimeemployee' table is selected, showing its columns: F_EmployeeID, departmentID, ManagerID, and salary_per_day.
- Result Grid:** A grid of data for the 'fulltimeemployee' table, listing 15 rows of employee IDs and their corresponding salaries.
- Action Output:** A log of SQL queries executed, showing the time, action, message, duration, and fetch size for each query.
- System Bar:** Shows the system status (41°C), user (عمر), language (ENG), and date/time (4:12 PM 5/20/2024).

F_EmployeeID	salary_per_day
E001	200
E002	180
E004	210
E006	190
E007	220
E009	160
E010	150
E012	230
E013	175
E015	205
E003	200
E005	200
E008	200
E011	200
E014	200

```

-----
INSERT INTO Faculty (FacultyID, teaching_experience, rankID)
VALUES ('E001', 5, 'R001');
INSERT INTO Faculty (FacultyID, teaching_experience, rankID)
VALUES ('E002', 3, 'R002');
INSERT INTO Faculty (FacultyID, teaching_experience, rankID)
VALUES ('E004', 8, 'R003');
INSERT INTO Faculty (FacultyID, teaching_experience, rankID)
VALUES ('E006', 6, 'R004');
INSERT INTO Faculty (FacultyID, teaching_experience, rankID)
VALUES ('E007', 10, 'R005');
INSERT INTO Faculty (FacultyID, teaching_experience, rankID)
VALUES ('E009', 4, 'R006');
INSERT INTO Faculty (FacultyID, teaching_experience, rankID)
VALUES ('E010', 2, 'R007');
INSERT INTO Faculty (FacultyID, teaching_experience, rankID)
VALUES ('E012', 7, 'R008');
INSERT INTO Faculty (FacultyID, teaching_experience, rankID)
VALUES ('E013', 9, 'R009');
INSERT INTO Faculty (FacultyID, teaching_experience, rankID)
VALUES ('E015', 11, 'R010');

```

The screenshot shows the MySQL Workbench interface with the following details:

- Schemas:** psupayroll
- Tables:** department, faculty, fulltimeemployee, parttimeemployee, administrator, job, jobrank, employee.
- Information:** The department table is selected. It has columns: departmentID (PK), departmentName (varchar(10)), ManagerID (varchar(10)). The data grid shows 15 rows of data:

departmentID	departmentName	ManagerID
E001	Mathematics	
E002	Physics	
E004	Chemistry	
E006	Biology	
E007	Computer Science	
E009	Engineering	
E012	Business Administration	
E013	History	
E015	Geography	
E003	Physics	
E005	Mathematics	
E008	Chemistry	
E010	Biology	
E011	Computer Science	
E014	Engineering	
E016	Business Administration	

- Action Output:** Shows the history of queries run on the department table.

```

-----
INSERT INTO PartTimeEmployee (P_EmployeeID,
teaching_hours_per_day, hourly_rate) VALUES ('E003', '6', '25');
INSERT INTO PartTimeEmployee (P_EmployeeID,
teaching_hours_per_day, hourly_rate) VALUES ('E005', '4', '18');
INSERT INTO PartTimeEmployee (P_EmployeeID,
teaching_hours_per_day, hourly_rate) VALUES ('E008', '7', '22');
INSERT INTO PartTimeEmployee (P_EmployeeID,
teaching_hours_per_day, hourly_rate) VALUES ('E010', '5', '30');
INSERT INTO PartTimeEmployee (P_EmployeeID,
teaching_hours_per_day, hourly_rate) VALUES ('E011', '6', '28');
INSERT INTO PartTimeEmployee (P_EmployeeID,
teaching_hours_per_day, hourly_rate) VALUES ('E014', '4', '26');

```

The screenshot shows the MySQL Workbench interface with the following details:

- Schemas:** The left sidebar shows the database structure with tables like parttimeemployee, fulltimeemployee, administrator, department, job, jobrank, employee, fulltimeemployee, faculty, and parttimeemployee.
- Table: parttimeemployee:** The main pane displays the data for the parttimeemployee table. The columns are P_EmployeeID, teaching_hours_per_day, and hourly_rate. The data is as follows:

P_EmployeeID	teaching_hours_per_day	hourly_rate
E003	6	25
E005	4	18
E008	7	22
E010	5	30
E011	6	28
E014	4	26

- History:** Below the table, the history pane shows the execution of several SELECT statements against various tables (job, jobrank, employee, fulltimeemployee, faculty, parttimeemployee) with their respective execution times and message details.
- System Bar:** The bottom bar includes the Windows taskbar with icons for File Explorer, Task View, Start, and others, along with system status information like temperature (41°C), language (ENG), and date/time (4/13 PM 5/20/2024).

```

INSERT INTO Attendance (AttendanceID, AttendanceDate, StartTime,
EndTime, No_of_Hours, P_EmployeeID) VALUES ('A001',
'2023-01-01', '09:00:00', '13:00:00', 4, 'E003');
INSERT INTO Attendance (AttendanceID, AttendanceDate, StartTime,
EndTime, No_of_Hours, P_EmployeeID) VALUES ('A002',
'2023-01-02', '10:00:00', '15:00:00', 5, 'E005');

```

```

INSERT INTO Attendance (AttendanceID, AttendanceDate, StartTime,
EndTime, No_of_Hours, P_EmployeeID) VALUES ('A006',
'2023-01-06', '09:00:00', '13:00:00', 4, 'E008');
INSERT INTO Attendance (AttendanceID, AttendanceDate, StartTime,
EndTime, No_of_Hours, P_EmployeeID) VALUES ('A007',
'2023-01-07', '10:00:00', '15:00:00', 5, 'E010');
INSERT INTO Attendance (AttendanceID, AttendanceDate, StartTime,
EndTime, No_of_Hours, P_EmployeeID) VALUES ('A008',
'2023-01-08', '08:00:00', '11:00:00', 3, 'E011');
INSERT INTO Attendance (AttendanceID, AttendanceDate, StartTime,
EndTime, No_of_Hours, P_EmployeeID) VALUES ('A009',
'2023-01-09', '09:00:00', '13:00:00', 4, 'E014');

```

The screenshot shows the MySQL Workbench interface with the following details:

- Schemas:** The left sidebar shows the database structure with tables like parttimeemployee, fulltimeemployee, administrator, department, job, jobrank, employee, fulltimeemployee, faculty, parttimeemployee, and attendance.
- Tables:** The main area displays the data from the `parttimeemployee` table in a grid format. The columns are `P_EmployeeID`, `Teaching_hours_per_day`, and `hourly_rate`. The data includes rows for employees A003 through A014.
- Query History:** The bottom section shows the history of executed queries, including SELECT statements for various tables like `jobrank`, `fulltimeemployee`, `faculty`, and `attendance`.
- System Status:** At the bottom right, it shows the system temperature (41°C), language (ENG), and date/time (6:13 PM, 5/20/2024).

```

INSERT INTO Administrator (AdminID) VALUES ('E001');
INSERT INTO Administrator (AdminID) VALUES ('E002');
INSERT INTO Administrator (AdminID) VALUES ('E004');
INSERT INTO Administrator (AdminID) VALUES ('E006');
INSERT INTO Administrator (AdminID) VALUES ('E007');
INSERT INTO Administrator (AdminID) VALUES ('E009');
INSERT INTO Administrator (AdminID) VALUES ('E010');
INSERT INTO Administrator (AdminID) VALUES ('E012');
INSERT INTO Administrator (AdminID) VALUES ('E013');
INSERT INTO Administrator (AdminID) VALUES ('E015');

```

The screenshot shows the MySQL Workbench interface. The top navigation bar includes File, Edit, View, Query, Database, Server, Tools, Scripting, and Help. The main window displays the 'psupayroll' schema with its tables: fulltimeemployee, parttimeemployee, administrator, department, job, jobrank, employee, fulltimeemployee, faculty, parttimeemployee, and attendance. The 'administrator' table is selected, showing 15 rows with AdminID values from E001 to E015. The 'Output' pane at the bottom shows the history of SQL queries run on the session, including SELECT statements for various tables like employee, faculty, and attendance.

```
--  
INSERT INTO Administrator_Job (AdminID, jobID) VALUES ('E001',  
'J001');  
INSERT INTO Administrator_Job (AdminID, jobID) VALUES ('E002',  
'J002');  
INSERT INTO Administrator_Job (AdminID, jobID) VALUES ('E004',  
'J003');  
INSERT INTO Administrator_Job (AdminID, jobID) VALUES ('E006',  
'J004');  
INSERT INTO Administrator_Job (AdminID, jobID) VALUES ('E007',  
'J005');  
INSERT INTO Administrator_Job (AdminID, jobID) VALUES ('E009',  
'J006');  
INSERT INTO Administrator_Job (AdminID, jobID) VALUES ('E010',  
'J007');  
INSERT INTO Administrator_Job (AdminID, jobID) VALUES ('E012',  
'J008');  
INSERT INTO Administrator_Job (AdminID, jobID) VALUES ('E013',  
'J009');  
INSERT INTO Administrator_Job (AdminID, jobID) VALUES ('E015',  
'J010');
```

The screenshot shows the MySQL Workbench interface with the PSUPayroll database selected. The left pane displays the schema browser for the psupayroll schema, showing tables like fulltimeemployee, parttimeemployee, administrator, department, job, jobrank, employee, and fullt. The right pane shows a query editor with the following SQL query:

```
1 • SELECT * FROM psupayroll.administrator_job;
```

The result grid for the administrator_job table contains the following data:

AdminID	JobID
E001	J001
E002	J002
E003	J003
E004	J004
E005	J005
E006	J006
E007	J007
E012	J008
E013	J009
E015	J010

```
--  
INSERT INTO ELeave (leaveID, leave_type, LeaveDesc) VALUES  
('L001', 'Sick', 'Sick Leave');  
INSERT INTO ELeave (leaveID, leave_type, LeaveDesc) VALUES  
('L002', 'Casual', 'Casual Leave');  
INSERT INTO ELeave (leaveID, leave_type, LeaveDesc) VALUES  
('L003', 'Maternity', 'Maternity Leave');  
INSERT INTO ELeave (leaveID, leave_type, LeaveDesc) VALUES  
('L004', 'Paternity', 'Paternity Leave');  
INSERT INTO ELeave (leaveID, leave_type, LeaveDesc) VALUES  
('L005', 'Bereavement', 'Bereavement Leave');  
INSERT INTO ELeave (leaveID, leave_type, LeaveDesc) VALUES  
('L006', 'Unpaid', 'Unpaid Leave');  
INSERT INTO ELeave (leaveID, leave_type, LeaveDesc) VALUES  
('L007', 'Annual', 'Annual Leave');  
INSERT INTO ELeave (leaveID, leave_type, LeaveDesc) VALUES  
('L008', 'Sabbatical', 'Sabbatical Leave');  
INSERT INTO ELeave (leaveID, leave_type, LeaveDesc) VALUES  
('L009', 'Study', 'Study Leave');  
INSERT INTO ELeave (leaveID, leave_type, LeaveDesc) VALUES  
('L010', 'Special', 'Special Leave');
```

The screenshot shows the MySQL Workbench interface with the PSUPayroll database selected. The left pane displays the schema browser with tables like fulltimeemployee, parttimeemployee, and administrator_job. The right pane shows a query editor with the following SQL code:

```
1 • SELECT * FROM psupayroll.leave;
```

The results grid displays the following data:

leaveID	leave_type	LeaveDesc
L001	SickLeave	Sick Leave
L002	Casual	Casual Leave
L003	Maternity	Maternity Leave
L004	Paternity	Paternity Leave
L005	Bereavement	Bereavement Leave
L006	Unpaid	Unpaid Leave
L007	Annual	Annual Leave
L008	Study	Study Leave
L009	Special	Special Leave
L010	Holiday	Holiday Leave

```
INSERT INTO PaidLeave (leaveID, employeeID, start_date,
end_date, duration) VALUES ('L001', 'E001', '2023-03-01',
'2023-03-10', 10);
INSERT INTO PaidLeave (leaveID, employeeID, start_date,
end_date, duration) VALUES ('L002', 'E002', '2023-04-01',
'2023-04-05', 5);
INSERT INTO PaidLeave (leaveID, employeeID, start_date,
end_date, duration) VALUES ('L003', 'E004', '2023-05-01',
'2023-05-15', 15);
INSERT INTO PaidLeave (leaveID, employeeID, start_date,
end_date, duration) VALUES ('L004', 'E006', '2023-06-01',
'2023-06-07', 7);
INSERT INTO PaidLeave (leaveID, employeeID, start_date,
end_date, duration) VALUES ('L006', 'E007', '2023-07-01',
'2023-07-02', 2);
INSERT INTO PaidLeave (leaveID, employeeID, start_date,
end_date, duration) VALUES ('L006', 'E009', '2023-08-01',
'2023-08-12', 12);
INSERT INTO PaidLeave (leaveID, employeeID, start_date,
end_date, duration) VALUES ('L007', 'E010', '2023-09-01',
'2023-09-08', 8);
INSERT INTO PaidLeave (leaveID, employeeID, start_date,
end_date, duration) VALUES ('L008', 'E012', '2023-10-01',
'2023-10-10', 10);
```

```

INSERT INTO PaidLeave (leaveID, employeeID, start_date,
end_date, duration) VALUES ('L009', 'E013', '2023-11-01',
'2023-11-05', 5);
INSERT INTO PaidLeave (leaveID, employeeID, start_date,
end_date, duration) VALUES ('L010', 'E015', '2023-12-01',
'2023-12-12', 12);

```

The screenshot shows the MySQL Workbench interface with the 'PSUpayroll' database selected. The 'Tables' tab is open, displaying the 'administrator_job' table. The table structure is as follows:

Column	Type	Comments
leaveID	varchar(10)	
employeeID	PK	
start_date	date	
end_date	date	
duration	int	

The data inserted is:

leaveID	employeeID	start_date	end_date	duration
L001	E001	2023-03-01	2023-03-10	10
L002	E002	2023-04-01	2023-04-15	5
L003	E004	2023-05-01	2023-05-15	15
L004	E006	2023-06-01	2023-06-07	7
L005	E007	2023-07-01	2023-07-02	2
L006	E009	2023-08-01	2023-08-12	12
L007	E010	2023-09-01	2023-09-10	10
L008	E012	2023-10-01	2023-10-10	10
L009	E013	2023-11-01	2023-11-05	5
L010	E015	2023-12-01	2023-12-12	12
L011	E018	2023-01-01	2023-01-05	5

```

---  

INSERT INTO PaySlip (paySlipID, F_EmployeeID, base_salary,
gross_salary, net_salary, date_created, ApprovedBy) VALUES
('P001', 'E001', '60000', '65000', '58000', '2023-01-01',
'E006');  

INSERT INTO PaySlip (paySlipID, F_EmployeeID, base_salary,
gross_salary, net_salary, date_created, ApprovedBy) VALUES
('P002', 'E002', '50000', '55000', '48000', '2023-02-01',
'E007');  

INSERT INTO PaySlip (paySlipID, F_EmployeeID, base_salary,
gross_salary, net_salary, date_created, ApprovedBy) VALUES
('P003', 'E004', '70000', '75000', '68000', '2023-03-01',
'E009');  

INSERT INTO PaySlip (paySlipID, F_EmployeeID, base_salary,
gross_salary, net_salary, date_created, ApprovedBy) VALUES
('P004', 'E006', '65000', '70000', '63000', '2023-04-01',
'E010');  

INSERT INTO PaySlip (paySlipID, F_EmployeeID, base_salary,
gross_salary, net_salary, date_created, ApprovedBy) VALUES
('P005', 'E007', '72000', '77000', '70000', '2023-05-01',
'E001');

```

```

INSERT INTO PaySlip (paySlipID, F_EmployeeID, base_salary,
gross_salary, net_salary, date_created, ApprovedBy) VALUES
('P006', 'E009', '48000', '53000', '45000', '2023-06-01',
'E002');

INSERT INTO PaySlip (paySlipID, F_EmployeeID, base_salary,
gross_salary, net_salary, date_created, ApprovedBy) VALUES
('P007', 'E010', '35000', '40000', '32000', '2023-07-01',
'E004');

INSERT INTO PaySlip (paySlipID, F_EmployeeID, base_salary,
gross_salary, net_salary, date_created, ApprovedBy) VALUES
('P008', 'E012', '80000', '85000', '78000', '2023-08-01',
'E015');

INSERT INTO PaySlip (paySlipID, F_EmployeeID, base_salary,
gross_salary, net_salary, date_created, ApprovedBy) VALUES
('P009', 'E013', '70000', '75000', '68000', '2023-09-01',
'E006');

INSERT INTO PaySlip (paySlipID, F_EmployeeID, base_salary,
gross_salary, net_salary, date_created, ApprovedBy) VALUES
('P010', 'E015', '90000', '95000', '88000', '2023-10-01',
'E007');

```

The screenshot shows the MySQL Workbench interface with the 'psupayroll' database selected. The 'Schemas' tree on the left lists various tables and objects. The 'paylip' table is currently selected in the main workspace. The 'Result Grid' tab displays 10 rows of data from the 'paylip' table, with columns including paySlipID, F_EmployeeID, base_salary, gross_salary, net_salary, date_created, and ApprovedBy. The 'Actions' tab at the bottom shows the history of SQL queries executed, such as SELECT statements for 'attendance', 'administrator', and 'paylip' tables.

paySlipID	F_EmployeeID	base_salary	gross_salary	net_salary	date_created	ApprovedBy
P001	E001	60000	65000	58000	2023-01-01	E006
P002	E002	55000	58000	53000	2023-02-01	E009
P003	E004	70000	75000	68000	2023-03-01	E009
P004	E006	65000	70000	63000	2023-04-01	E010
P005	E007	72000	77000	70000	2023-05-01	E001
P006	E009	45000	50000	48000	2023-06-01	E002
P007	E010	25000	30000	28000	2023-07-01	E004
P008	E012	80000	85000	78000	2023-08-01	E015
P009	E013	70000	75000	68000	2023-09-01	E006
P010	E015	90000	95000	88000	2023-10-01	E007

```
INSERT INTO Allowance (allowanceID, allowanceType,
allowance_amount, effectiveDate, PaySlipID) VALUES ('A001',
'Travel', '1000', '2023-01-01', 'P001');
INSERT INTO Allowance (allowanceID, allowanceType,
allowance_amount, effectiveDate, PaySlipID) VALUES ('A002',
'Housing', '2000', '2023-02-01', 'P002');
INSERT INTO Allowance (allowanceID, allowanceType,
allowance_amount, effectiveDate, PaySlipID) VALUES ('A003',
'Medical', '1500', '2023-03-01', 'P003');
INSERT INTO Allowance (allowanceID, allowanceType,
allowance_amount, effectiveDate, PaySlipID) VALUES ('A004',
'Food', '800', '2023-04-01', 'P004');
INSERT INTO Allowance (allowanceID, allowanceType,
allowance_amount, effectiveDate, PaySlipID) VALUES ('A005',
'Transport', '900', '2023-05-01', 'P005');
INSERT INTO Allowance (allowanceID, allowanceType,
allowance_amount, effectiveDate, PaySlipID) VALUES ('A006',
'Education', '1100', '2023-06-01', 'P006');
INSERT INTO Allowance (allowanceID, allowanceType,
allowance_amount, effectiveDate, PaySlipID) VALUES ('A007',
'Childcare', '1200', '2023-07-01', 'P007');
INSERT INTO Allowance (allowanceID, allowanceType,
allowance_amount, effectiveDate, PaySlipID) VALUES ('A008',
'Internet', '1300', '2023-08-01', 'P008');
INSERT INTO Allowance (allowanceID, allowanceType,
allowance_amount, effectiveDate, PaySlipID) VALUES ('A009',
'Utilities', '1400', '2023-09-01', 'P009');
INSERT INTO Allowance (allowanceID, allowanceType,
allowance_amount, effectiveDate, PaySlipID) VALUES ('A010',
'Gym', '1500', '2023-10-01', 'P010');
```

The screenshot shows the MySQL Workbench interface with the following details:

- Schemas:** PSUPayroll (selected), psupayroll, PSUPayroll
- Tables:** allowance, administrator, department, job, jobrank, employee, fulltimeemployee, faculty, parttimeemployee, attendance, administrator_job, leave, paidleave, paylip
- Current Query:** SELECT * FROM psupayroll.allowance;
- Result Grid:** Shows the allowance table with columns: allowanceID, allowanceType, allowance_amount, effectiveDate, PaySlipID. Data rows include A001 (Travel, 1000, 2023-01-01, P001), A002 (Housing, 2000, 2023-01-01, P002), A003 (Medical, 1000, 2023-01-01, P003), A004 (Food, 800, 2023-01-01, P004), A005 (Transport, 900, 2023-01-01, P005), A006 (Education, 1100, 2023-01-01, P006), A007 (Childcare, 1200, 2023-01-01, P007), A008 (Rent, 1300, 2023-01-01, P008), A009 (Utilities, 1400, 2023-01-01, P009), A010 (Gym, 1500, 2023-01-01, P010).
- Action Output:** Shows a history of actions with timestamps, queries, messages, and duration/fetch times.

```
-----
INSERT INTO Deduction (DeductionID, DeductionType,
DeductionDesc, PaySlipID) VALUES ('D001', 'Tax', 'Income Tax',
'P001');
INSERT INTO Deduction (DeductionID, DeductionType,
DeductionDesc, PaySlipID) VALUES ('D002', 'Pension', 'Pension Fund',
'P002') ;
INSERT INTO Deduction (DeductionID, DeductionType,
DeductionDesc, PaySlipID) VALUES ('D003', 'Insurance', 'Health Insurance',
'P003') ;
INSERT INTO Deduction (DeductionID, DeductionType,
DeductionDesc, PaySlipID) VALUES ('D004', 'Loan', 'Student Loan',
'P004') ;
INSERT INTO Deduction (DeductionID, DeductionType,
DeductionDesc, PaySlipID) VALUES ('D005', 'Savings', 'Savings Plan',
'P005') ;
INSERT INTO Deduction (DeductionID, DeductionType,
DeductionDesc, PaySlipID) VALUES ('D006', 'Union', 'Union Dues',
'P006') ;
INSERT INTO Deduction (DeductionID, DeductionType,
DeductionDesc, PaySlipID) VALUES ('D007', 'Parking', 'Parking Fee',
'P007') ;
```

```

INSERT INTO Deduction (DeductionID, DeductionType,
DeductionDesc, PaySlipID) VALUES ('D008', 'Healthcare',
'Healthcare Contribution', 'P008') ;
INSERT INTO Deduction (DeductionID, DeductionType,
DeductionDesc, PaySlipID) VALUES ('D009', 'Charity', 'Charity
Donation', 'P009') ;
INSERT INTO Deduction (DeductionID, DeductionType,
DeductionDesc, PaySlipID) VALUES ('D010', 'Subscription',
'Magazine Subscription', 'P010') ;

```

The screenshot shows the MySQL Workbench interface with the following details:

- Schemas:** Shows the database structure with tables like fulltimeemployee, parttimeemployee, administrator, department, job, jobrank, employee, faculty, parttimeemployee, attendance, administrator, administrator_job, leave, paidleave, payslip, allowance, and deduction.
- Table: payslip** (selected):

Column	Type	DeductionID	DeductionType	PayslipID
paySlipID	varchar(10)	D001	Tax	P001
F_EmployeeID	PK varchar(10)	D002	Pension	P002
gross_salary	varchar(10)	D003	Insurance	P003
net_salary	varchar(10)	D004	Loan	P004
date_created	date	D005	Summies	P005
ApprovedBy	varchar(10)	D006	Union	P006
		D007	Parking	P007
		D008	Healthcare	P008
		D009	Charity	P009
		D010	Subscription	P010
- deduction 1** (selected):

Action	Time	Action	Time	Message	Duration / Fetch
342	16:13:58	SELECT * FROM psupayroll.administrator_job LIMIT 0, 1000		10 rows(s) returned	0.000 sec / 0.000 sec
343	16:14:21	SELECT * FROM psupayroll.leave LIMIT 0, 1000		10 rows(s) returned	0.015 sec / 0.000 sec
344	16:14:46	SELECT * FROM psupayroll.paidleave LIMIT 0, 1000		10 rows(s) returned	0.000 sec / 0.000 sec
345	16:15:14	SELECT * FROM psupayroll.payslip LIMIT 0, 1000		10 rows(s) returned	0.000 sec / 0.000 sec
346	16:15:30	SELECT * FROM psupayroll.allowance LIMIT 0, 1000		10 rows(s) returned	0.000 sec / 0.000 sec
347	16:15:48	SELECT * FROM psupayroll.deduction LIMIT 0, 1000		10 rows(s) returned	0.000 sec / 0.000 sec

- Basic SQL Queries:

- Get the department with the highest number of employees.

```

SELECT SUM(net_salary) AS total_salary_paid
FROM PaySlip
INNER JOIN FullTimeEmployee ON PaySlip.F_EmployeeID =
FullTimeEmployee.F_EmployeeID;

```

```
426 • SELECT SUM(net_salary) AS total_salary_paid
427   FROM PaySlip
428   INNER JOIN FullTimeEmployee ON PaySlip.F_EmployeeID = FullTimeEmployee.F_EmployeeID
429
430
```

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

total_salary_paid
618000

Result 25 x			
Output			
Action Output			Message
#	Time	Action	
1	19:59:47	SELECT SUM(net_salary) AS total_salary_paid FROM PaySlip INNER JOIN FullTimeEmployee ...	1 row(s) returned

2. Find the average number of teaching hours per day for part-time employees.

```
SELECT AVG(teaching_hours_per_day) AS average_teaching_hours  
FROM PartTimeEmployee;
```

```
429 • SELECT AVG(teaching_hours_per_day) AS average_teaching_hours  
430 FROM PartTimeEmployee;  
431  
432  
433  
< Result Grid | Filter Rows:  | Export:  | Wrap Cell Content:   


| average_teaching_hours |
|------------------------|
| 5.333333333333333      |


```

#	Time	Action	Message
1	20:02:21	SELECT AVG(teaching_hours_per_day) AS average_teaching_hours FROM PartTimeEmployee...	1 row(s) returned

3. List all employees who have not taken any paid leaves:

```
SELECT Employee.employeeID, Employee.fname, Employee.lname  
FROM Employee  
LEFT JOIN PaidLeave ON Employee.employeeID =  
PaidLeave.employeeID  
WHERE PaidLeave.leaveID IS NULL;
```

```

370 •   SELECT Employee.employeeID, Employee.fname, Employee.lname
371   FROM Employee
372   LEFT JOIN PaidLeave ON Employee.employeeID = PaidLeave.employeeID
373   WHERE PaidLeave.leaveID IS NULL;
374
375
376
< Result Grid | Filter Rows: [ ] | Export: [ ] | Wrap Cell Content: [ ]


| employeeID | fname    | lname      |
|------------|----------|------------|
| E003       | Mohammed | Al-Olaibi  |
| E005       | Yousef   | Al-Mutairi |
| E008       | Mona     | Al-Harbi   |
| E011       | Hassan   | Al-Juhani  |
| E014       | Rania    | Al-Shehri  |


```

Result 14 ×

Output

Action Output

#	Time	Action	Message
1	19:38:40	SELECT Employee.employeeID, Employee.fname, Employee.lname FROM Employee ...	5 row(s) returned

4. Find employees' IDs and their corresponding department names:

```

SELECT Employee.employeeID, Department.name
FROM Employee INNER JOIN Department
ON Employee.departmentID = Department.departmentID;

```

```

314 •   SELECT Employee.employeeID, Department.name
315   FROM Employee INNER JOIN Department
316   ON Employee.departmentID = Department.departmentID;
317
318
319
320
321
< Result Grid | Filter Rows: [ ] | Export: [ ] | Wrap Cell Content: [ ]


| employeeID | name       |
|------------|------------|
| E001       | HR         |
| E011       | HR         |
| E002       | Finance    |
| E012       | Finance    |
| E003       | IT         |
| E013       | IT         |
| E004       | Marketing  |
| E014       | Marketing  |
| E005       | Sales      |
| E015       | Sales      |
| E006       | Legal      |
| E007       | Operations |
| E008       | Research   |


```

Result 1 ×

Output

Action Output

#	Time	Action	Message
1	19:18:44	SELECT Employee.employeeID, Department.name FROM Employee INNER JOIN Department. ON Employee.departmentID... 15 row(s) returned	

5. Count the number of employees in each department.

```

SELECT departmentID, COUNT(*)
FROM Employee
NATURAL JOIN Department
GROUP BY departmentID;

```

```

312 - INSERT INTO Deduction (DeductionID, Dev
313
314 • SELECT departmentID, COUNT(*)
315   FROM Employee
316   NATURAL JOIN Department
317   GROUP BY departmentID;
318
319
320
??

```

Result Grid | Filter Rows: [] | Export: []

departmentID	COUNT(*)
D001	2
D002	2
D003	2
D004	2
D005	2
D006	1
D007	1
D008	1
D009	1
D010	1

6. Get the total number of paid leaves for each job title.

```

SELECT job_title, SUM(num_paid_leaves)
FROM Job
GROUP BY job_title;

```

```

318
319 • SELECT job_title, SUM(num_paid_leaves)
320   FROM Job
321   GROUP BY job_title;
322
??

```

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

job_title	SUM(num_paid_leaves)
Instructor	50
Analyst	15
Head of CS	20
Administrator	15
Assistant	28
Administrator	16
Supervisor	10

7. Retrieve the top 3 highest earning full-time employees based on salary per day.

```

SELECT F_EmployeeID, salary_per_day
FROM FullTimeEmployee
ORDER BY salary_per_day DESC
LIMIT 3 ;

```

```

MySQL Workbench
File Edit View Query Database Server Tools Scripting Help
Navigator
Schemas
PSUPayroll (psupayroll) x payroll x PSUPayroll x
PSUPayroll* SQL File 1* parttimeemployee
331    SELECT p.EmployeeID, NoDay, rate FROM ParttimeEmployee;
332    SELECT a.EmployeeID, j.JobTitle FROM Administrator a JOIN Administrator_Job aj ON a.EmployeeID = aj.EmployeeID JOIN Job j ON aj.JobID = j.JobID;
333    SELECT leave_type, leave_days FROM Leave;
334
335    SELECT SUM(base_salary * rate) AS salary_per_day FROM Employee;
336    SELECT AVG(CAST(base_salary AS DECIMAL)) FROM JobRank;
337    SELECT * FROM FulltimeEmployee ORDER BY CAST(salary_per_day AS DECIMAL) DESC FETCH FIRST 3 ROW ONLY;
338
339    SELECT SUM(base_salary * rate) AS salary_per_day FROM Employee;
340    SELECT AVG(CAST(base_salary AS DECIMAL)) FROM JobRank;
341    SELECT * FROM FulltimeEmployee ORDER BY CAST(salary_per_day AS DECIMAL) DESC FETCH FIRST 3 ROW ONLY;
342

```

Result Grid | Filter Rows | Export | Wrap Cell Content | Fetch rows |

P.EmployeeID	NoDay	salary_per_day
1012	220	220
1009	220	220
1004	210	210
1008	200	200

8. Find the total number of hours worked by part-time employees.

```
SELECT SUM(No_of_Hours) FROM Attendance;
```

```

MySQL Workbench
File Edit View Query Database Server Tools Scripting Help
Navigator
Schemas
PSUPayroll (psupayroll) x payroll x PSUPayroll x
PSUPayroll* SQL File 1* fulltimeemployee parttimeemployee
347
348    --
349
350    SELECT fname, lname
351    FROM Employee
352    WHERE hire_date > '2015-01-01';
353    -
354    SELECT SUM(No_of_Hours) FROM Attendance;
355
356

```

Result Grid | Filter Rows | Export | Wrap Cell Content | Fetch rows |

SUM(No_of_Hours)
25

9. Retrieves employee information with their respective department names, ensuring all employees are included.

```
SELECT Employee.employeeID, Employee.fname, Employee.lname,
Department.name
FROM Employee
LEFT JOIN Department ON Employee.departmentID =
Department.departmentID;
```

```

442
443 •   SELECT Employee.employeeID, Employee.fname, Employee.lname, Department.name
444     FROM Employee
445     LEFT JOIN Department ON Employee.departmentID = Department.departmentID;
446
447

```

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

employeeID	fname	lname	name
E001	Abdullah	Al-Saud	HR
E002	Fatimah	Al-Faisal	Finance
E003	Mohammed	Al-Otaibi	IT
E004	Noura	Al-Rashid	Marketing
E005	Yousef	Al-Mutairi	Sales
E006	Huda	Al-Qahtani	Legal
E007	Saud	Al-Ghamdi	Operations
E008	Mona	Al-Harbi	Research
E009	Saleh	Al-Subaei	Development
E010	Aisha	Al-Mansour	Support
E011	Hassan	Al-Juhani	HR
E012	Lala	Al-Zahrani	Finance
E013	Ahmed	Al-Shammary	IT
E014	Rania	Al-Shehri	Marketing
E015	Sami	Al-Anazi	Sales

Result 30 × Read Only Cor

Action Output

#	Time	Action
1	20:21:26	SELECT Employee.employeeID, Employee.fname, Employee.lname, Department.name FROM E... 15 row(s) returned

10. Retrieve employees hired after January 1, 2015:

```

SELECT fname, lname
FROM Employee
WHERE hire_date > '2015-01-01';

```

MySQL Workbench

File Edit View Query Database Server Tools Scripting Help

Navigator: PSUPayroll Schemas Administration

SQL File 1* fulltimeemployee parttimeemployee

```

354
355     SELECT SUM(No_of_Hours) FROM Attendance;
356
357     SELECT allowanceID, allowanceType, effectiveDate
358     FROM Allowance;
359
360     SELECT fname, lname
361     FROM Employee
362     WHERE hire_date > '2015-01-01';
363

```

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

fname	lname
Mona	Al-Harbi
Saleh	Al-Subaei
Aisha	Al-Mansour
Hassan	Al-Juhani

11. Retrieve attendance records for employee E005:

```

SELECT A.date, A.attendance_status, E.fname, E.lname
FROM Attendance A
JOIN Employee E ON A.employee_id = E.employee_id
WHERE A.employee_id = 'E005';

```

The screenshot shows the MySQL Workbench interface. In the top navigation bar, there are tabs for 'PSUPayroll' (selected), 'psupayroll', 'psupayroll', and 'PSUPayroll'. The main area has a 'SQL File 1' tab with the following SQL code:

```

307 • INSERT INTO Deduction (DeductionID, DeductionType, DeductionDesc, PaySlipID) VALUES ('D000', 'Union', 'Union Dues', 'P000');
308 • INSERT INTO Deduction (DeductionID, DeductionType, DeductionDesc, PaySlipID) VALUES ('D001', 'Parking', 'Parking Fee', 'P001');
309 • INSERT INTO Deduction (DeductionID, DeductionType, DeductionDesc, PaySlipID) VALUES ('D002', 'Healthcare', 'Healthcare Contribution', 'P002');
310 • INSERT INTO Deduction (DeductionID, DeductionType, DeductionDesc, PaySlipID) VALUES ('D003', 'Charity', 'Charity Donation', 'P003');
311 • INSERT INTO Deduction (DeductionID, DeductionType, DeductionDesc, PaySlipID) VALUES ('D010', 'Subscription', 'Magazine Subscription', 'P010');
312
313
314 • SELECT *
315 FROM Attendance

```

Below the code is a result grid titled 'Attendance' with the following data:

AttendanceID	AttendanceDate	StartTime	EndTime	No_of_Hours	P_EmployeeID
A002	2023-01-02	10:00:00	15:00:00	5	E005

12. Maximum Years of Experience per Department:

```

SELECT Department.name, MAX(Employee.yrs_of_experience) AS max_experience
FROM Department
LEFT JOIN Employee ON Department.departmentID =
Employee.departmentID
GROUP BY Department.name;

```

The screenshot shows the MySQL Workbench interface with the following SQL code in the editor:

```

452 • SELECT Department.name, MAX(Employee.yrs_of_experience) AS max_experience
453 FROM Department
454 LEFT JOIN Employee ON Department.departmentID = Employee.departmentID
455 GROUP BY Department.name;
456

```

Below the code is a result grid titled 'Result Grid' with the following data:

name	max_experience
HR	10
Finance	11
IT	10
Marketing	15
Sales	9
Legal	7
Operations	12
Research	5
Development	4
Support	3

The screenshot shows the MySQL Workbench interface with the following message in the 'Output' section:

Action Output

#	Time	Action	Message
1	20:23:44	SELECT Department.name, MAX(Employee.yrs_of_experience) AS max_experience FROM De...	10 row(s) returned

- Advanced SQL Queries:

1. Retrieve the total salary paid to all full-time employees:

```

SELECT SUM(net_salary) AS total_salary_paid
FROM PaySlip
WHERE F_EmployeeID IN

```

```
(SELECT F_EmployeeID FROM FullTimeEmployee);
```

```
410  -- 1. Retrieve the total salary paid to all full-time employees:  
411 • SELECT SUM(net_salary) AS total_salary_paid  
412   FROM PaySlip  
413   WHERE F_EmployeeID IN  
414     (SELECT F_EmployeeID FROM FullTimeEmployee)  
415  
416  
417
```

Result Grid | Filter Rows: [] | Export: [] | Wrap Cell Content: []
total_salary_paid
1 518000

```
Result 24 x  
Output  
Action Output  
# Time Action Message  
1 19:50:23 2024-05-01 19:50:23 SELECT SUM(net_salary) AS total_salary_paid FROM PaySlip WHERE F_EmployeeID IN (SELECT F_EmployeeID FROM FullTimeEmployee) returned
```

2. Find employees who worked on specific dates:

```
SELECT e.FName, e.LName, a.AttendanceDate  
FROM Employee e JOIN Attendance a  
ON e.employeeID = a.AttendanceID  
WHERE a.AttendanceDate IN ('2024-05-01', '2024-05-02',  
'2024-05-03')  
ORDER BY a.AttendanceDate, e.LName, e.FName;
```

```
436  
437 • SELECT e.FName, e.LName, a.AttendanceDate  
438   FROM Employee e JOIN Attendance a  
439   ON e.employeeID = a.AttendanceID  
440   WHERE a.AttendanceDate IN ('2024-05-01', '2024-05-02', '2024-05-03')  
441   ORDER BY a.AttendanceDate, e.LName, e.FName  
442  
443  
444  
445
```

Result Grid | Filter Rows: [] | Export: [] | Wrap Cell Content: []
FName LName AttendanceDate

```
Result 29 x  
Output  
Action Output  
# Time Action Message  
1 20:17:09 2024-05-01 20:17:09 SELECT e.FName, e.LName, a.AttendanceDate FROM Employee e JOIN Attendance a ON e.Emplo... 0 row(s) returned
```

3. Retrieve the employees who have not taken any leaves.

```
SELECT * FROM Employee e WHERE NOT EXISTS (SELECT * FROM  
PaidLeave pl WHERE e.employeeID = pl.employeeID);
```

MySQL Workbench

File Edit View Query Database Server Tools Scripting Help

Navigator: Schemas

Filter objects

PSUpayroll SQL File 1* fulltimeemployee parttimeemployee

Find: Q:-

Line numbers: 335, 336, 337, 338, 339, 340, 341, 342, 343, 344, 345, 346, 347, 348, 349.

335 FROM Employee

336 LEFT JOIN PaidLeave ON Employee.employeeID = PaidLeave.employeeID

337 WHERE PaidLeave.leaveID IS NULL;

338 ---

339

340 • SELECT d.name AS department_name, COUNT(pl.leaveID) AS total_leaves_taken

341 FROM Department d

342 LEFT JOIN Employee e ON d.departmentID = e.departmentID

343 LEFT JOIN PaidLeave pl ON e.employeeID = pl.employeeID

344 GROUP BY d.name;

345

346 -----

347

348 • SELECT * FROM Employee e WHERE NOT EXISTS (SELECT * FROM PaidLeave pl WHERE e.employeeID = pl.employeeID);

349

Administration Schemas Information

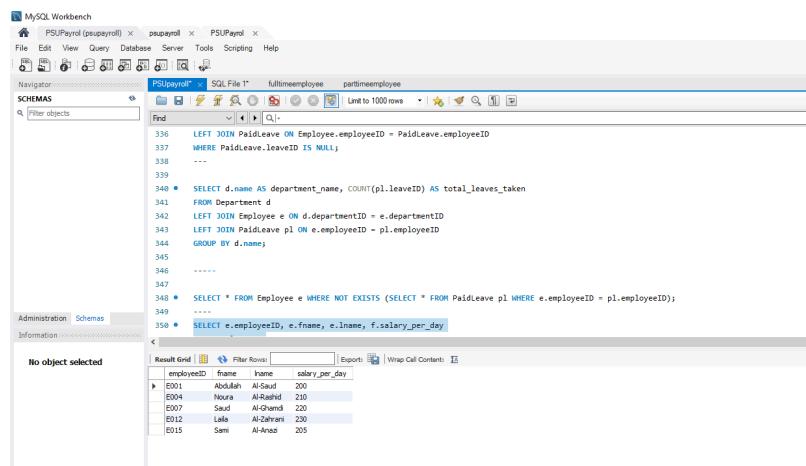
No object selected

Result Grid Filter Rows: [] Edit: [] Export/Import: [] Wrap Cell Contents: []

employeeID	fname	lname	gender	dt_birth	hire_dt	phone_number	email_address	user_type	yr_of_experience	departmentID
E003	Mohammed	Al-Otabi	M	1990-07-25	2014-06-01	0503456789	mohamed.alotabi@example.com	PartTime	6	D003
E005	Yousef	Al-Hutani	M	1982-09-15	2011-08-01	0505678910	yousef.alhutani@example.com	PartTime	9	D005
E007	Anees	Al-Saadi	M	1985-03-20	2013-07-15	0507890123	anees.alsaadi@example.com	PartTime	5	D008
E011	Hassan	Al-Juhani	M	1987-03-22	2018-01-01	0501234566	hassan.aljuhani@example.com	PartTime	2	D001
E014	Rana	Al-shehri	F	1991-06-18	2013-04-01	0502233445	rana.alshehri@example.com	PartTime	9	D004

4. Get the full-time employees who have a salary per day higher than the average.

```
SELECT e.employeeID, e.fname, e.lname, f.salary_per_day
FROM Employee e
JOIN FullTimeEmployee f ON e.employeeID = f.F_EmployeeID
WHERE f.salary_per_day > (
    SELECT AVG(salary_per_day)
    FROM FullTimeEmployee
)
```



5. List the employees who are managers of their departments.

```
SELECT e.* FROM Employee e JOIN Department d ON e.employeeID = d.ManagerID;
```

```

MySQL Workbench
File Edit View Query Database Server Tools Scripting Help
Navigator: PSUPayroll (psupayroll) Schema: psupayroll
Find: SQL File: fulltimeemployee partimeemployee Unit to 1000 rows | | | | | | | |
348 • SELECT * FROM Employee e WHERE NOT EXISTS (SELECT * FROM PaidLeave pl WHERE e.employeeID = pl.employeeID);
349 ----
350 • SELECT e.employeeID, e.fname, e.lname, f.salary_per_day
351   FROM Employee e
352   JOIN FulltimeEmployee f ON e.employeeID = f.F_EmployeeID
353   WHERE f.salary_per_day > (
354     SELECT AVG(salary_per_day)
355       FROM FulltimeEmployee
356   )
357   ...
358   ...
359 • SELECT * FROM FullTimeEmployee
360   WHERE salary_per_day > (SELECT AVG(salary_per_day) FROM FullTimeEmployee)
361   ...
362 • SELECT e.*
```

Result Grid:

employeeID	fname	lname	gender	dt_birth	hire_date	phone_number	email_address	user_type	yrs_of_experience	departmentID
E001	Abdullah	Al-Saud	M	1980-05-15	2010-04-01	0501234567	abdullah.al-saud@example.com	FullTime	10	D001
E002	Fatmeh	Al-Hassid	F	1985-06-20	2012-01-01	0502345678	fatmeh.al-hassid@example.com	FullTime	8	D002
E003	Noura	Al-Hassid	F	1975-07-15	2008-07-01	0503456789	noura.al-hassid@example.com	FullTime	15	D004
E004	Huda	Al-Zahrani	F	1988-09-20	2011-03-01	0504567890	huda.al-zahrani@example.com	FullTime	7	D006
E007	Saud	Al-Qharni	M	1979-11-25	2009-10-01	0507990123	saud.al-qharni@example.com	FullTime	12	D007
E009	Saleh	Al-Suhaimi	M	1992-01-05	2016-10-01	0509012345	saleh.al-suhaimi@example.com	FullTime	4	D009
E012	Lala	Al-Zahrani	F	1987-04-12	2011-03-01	0502233444	lala.al-zahrani@example.com	FullTime	11	D002
E013	Ahmed	Al-Shammar	M	1989-05-15	2012-04-01	0501122344	ahmed.al-shammar@example.com	FullTime	10	D003
E015	Sam	Al-Arekat	M	1980-07-21	2014-06-01	0503344566	sam.al-arekat@example.com	FullTime	8	D005

6. Get the total gross salary paid to employees by each department.

```

SELECT d.name AS department_name, SUM(ps.gross_salary) AS
total_gross_salary
FROM Employee e
JOIN Department d ON e.departmentID = d.departmentID
JOIN PaySlip ps ON e.employeeID = ps.F_EmployeeID
GROUP BY d.name;
```

```

MySQL Workbench
File Edit View Query Database Server Tools Scripting Help
Navigator: PSUPayroll (psupayroll) Schema: psupayroll
Find: SQL File: fulltimeemployee partimeemployee Unit to 1000 rows | | | | | | | |
365 • WHERE e.user_type = 'fulltime'
366
367   ...
368
369 • SELECT * FROM Department WHERE ManagerID NOT IN (SELECT F_EmployeeID FROM FulltimeEmployee);
370   ...
371
372 • SELECT MAX(salary_per_day) AS highest_salary_per_day
373   FROM FulltimeEmployee;
374   ...
375 • SELECT d.name AS department_name, SUM(ps.gross_salary) AS total_gross_salary
376   FROM Employee e
377   JOIN Department d ON e.departmentID = d.departmentID
378   JOIN PaySlip ps ON e.employeeID = ps.F_EmployeeID
379   GROUP BY d.name;
```

Result Grid:

department_name	total_gross_salary
HR	65000
Finance	44000
IT	78000
Marketing	75000
Sales	95000
Legal	70000
Operations	77000
Development	55000
Support	40000

7. List the job ranks and the number of faculty members in each rank.

```

SELECT jr.rank_name, COUNT(*) AS num_faculty_members
FROM JobRank jr
```

```
JOIN Faculty f ON jr.rankID = f.rankID  
GROUP BY jr.rank_name;
```

The screenshot shows the MySQL Workbench interface with the following details:

- File Menu:** File, Edit, View, Query, Database, Server, Tools, Scripting, Help.
- Schemas:** Schemas dropdown.
- Query Editor:** Contains a multi-line SQL query:

```
342 + SELECT e.*,
343     min(DeployID) AS min_deployID,
344     max(DeployID) AS max_deployID
345     WHERE e.user_type = 'fulltime'
346
347 ...
348
349     * FROM Department WHERE ManagerID NOT IN (SELECT p_Deployed FROM nullDeploy);
350
351 ...
352
353     SELECT MAX(salary_per_day) AS highest_salary_per_day
354     FROM fulltimeemployees;
355
356     SELECT jn.rank_name, COUNT(*) AS num_faculty_members;
```
- Results Grid:** Shows the results of the query:

Faculty Members	Count
Professor	1
Associate Professor	1
Senior Lecturer	1
Lecturer	1
Principal	1
Other	1
Executive	1
Administrative	1
Associate	1
Teaching Assistant	1
- Logs:** Displays the execution logs for the query:

```
10:51:45 [1] 10543 SELECT jn.rank_name, COUNT(*) AS num_faculty_members FROM faculty JOIN Adminstration ON faculty.jrn_id = Adminstration.jrn_id WHERE ManagerID IS NOT NULL AND ManagerID NOT IN (SELECT p_Deployed FROM nullDeploy);
10:51:45 [1] 10543 FROM Department WHERE ManagerID IS NOT NULL AND ManagerID NOT IN (SELECT p_Deployed FROM nullDeploy);
10:51:45 [1] 10543 SELECT ManagerID AS Manager, salary_per_day FROM fulltimeemployees ORDER BY salary_per_day DESC LIMIT 1;
10:51:45 [1] 10543 FROM Department WHERE ManagerID IS NOT NULL AND ManagerID NOT IN (SELECT p_Deployed FROM nullDeploy);
10:51:45 [1] 10543 SELECT MAX(salary_per_day) AS highest_salary_per_day FROM fulltimeemployees;
10:51:45 [1] 10543 FROM fulltimeemployees;
10:51:45 [1] 10543 SELECT jn.rank_name, COUNT(*) AS num_faculty_members FROM faculty JOIN Adminstration ON faculty.jrn_id = Adminstration.jrn_id WHERE ManagerID IS NOT NULL AND ManagerID NOT IN (SELECT p_Deployed FROM nullDeploy);
10:51:45 [1] 10543 FROM Department WHERE ManagerID IS NOT NULL AND ManagerID NOT IN (SELECT p_Deployed FROM nullDeploy);
10:51:45 [1] 10543 SELECT e.*,
10:51:45 [1] 10543     min(DeployID) AS min_deployID,
10:51:45 [1] 10543     max(DeployID) AS max_deployID
10:51:45 [1] 10543     WHERE e.user_type = 'fulltime'
10:51:45 [1] 10543
10:51:45 [1] 10543 * FROM Department WHERE ManagerID NOT IN (SELECT p_Deployed FROM nullDeploy);
```
- Object Info:** Shows the current object being edited.
- Session:** Shows the session status.
- Query Compiler:** Shows the type here to search.

8. Find the total number of paid leaves taken by each employee:

```
SELECT e.employeeID, e.fname, e.lname, COUNT(pl.leaveID) AS  
total_paid_leaves  
FROM Employee e  
LEFT JOIN PaidLeave pl ON e.employeeID = pl.employeeID  
GROUP BY e.employeeID, e.fname, e.lname;
```

```
ucapoply x PSUProjx
Server Tools Scripting Help
File Edit View Insert Query SQL File T-SQL FullTextSearch List to 1000 rows
Find < | > C:\|_
347
348 • SELECT * FROM Employee e WHERE NOT EXISTS (SELECT * FROM PaidLeave pl WHERE e.employeeID = pl.employeeID);
349 ----
350 • SELECT e.employeeID, e.fname, e.lname, f.salary_per_day
351   FROM Employee e
352   JOIN FulltimeEmployee f ON e.employeeID = f.F_EmployeeID
353   WHERE f.salary_per_day > (
354     SELECT AVG(salary_per_day)
355     FROM FulltimeEmployee
356   )
357
358 • SELECT e.employeeID, e.fname, e.lname, COUNT(pl.leaveID) AS total_paid_leaves
359   FROM Employee e
360   LEFT JOIN PaidLeave pl ON e.employeeID = pl.employeeID
361   GROUP BY e.employeeID, e.fname, e.lname;
362

Result Grid [ ] Filter Rows [ ] Export [ ] Wrap Cell Content: [ ]
employeeID fname lname total_paid_leaves
E001 Abdullah Al Saad 1
E002 Fahdah Al Faseel 1
E003 Mohamed Al Rashed 0
E004 Noura Al Raheeb 1
E005 Yousef Al Mutari 0
E006 Huda Al Qatnani 1
E007 Zaid Al Khatib 0
E008 Muna Al Hawali 0
E009 Saleh Al Subail 1
E010 Adha Al Marssouf 1
E011 Hanan Al Ghurabi 0
E012 Leila Al Zahrani 1
total 16

```

9. Get the list of employees who have never been approved for a payslip by any administrator.

```
SELECT * FROM Employee e WHERE NOT EXISTS (SELECT * FROM PaySlip ps WHERE e.employeeID = ps.F_EmployeeID AND ps.ApprovedBy IS NOT NULL);
```

The screenshot shows the MySQL Workbench interface with the following details:

- File Menu:** File, Edit, View, Query, Database, Server, Tools, Scripting, Help.
- Navigator:** SCHEMAS (Filter objects), showing PSUPayroll.
- SQL Editor:** SQL File 1*, titled "fulltimeemployee". The query is as follows:

```
344 GROUP BY d.name;
345
346 -----
347
348 • SELECT * FROM Employee e WHERE NOT EXISTS (SELECT * FROM PaidLeave pl WHERE e.employeeID = pl.employeeID);
349 -----
350 • SELECT e.employeeID, e.fname, e.lname, f.salary_per_day
351 FROM Employee e
352 JOIN FulltimeEmployee f ON e.employeeID = f.F_EmployeeID
353 WHERE f.salary_per_day > (
354     SELECT AVG(salary_per_day)
355     FROM FulltimeEmployee
356 )
357 • SELECT * FROM Employee e WHERE NOT EXISTS (SELECT * FROM PaySlip ps WHERE e.employeeID = ps.F_EmployeeID AND ps.ApprovedBy IS NOT NULL);
358
```

Administration Schemas Information pane is visible on the left.

Results Grid: Shows the output of the query with columns: employeeID, fname, lname, gender, dt_birth, hre_date, phone_number, email_address, user_type, yrs_of_experience, departmentID. The data is as follows:

employeeID	fname	lname	gender	dt_birth	hre_date	phone_number	email_address	user_type	yrs_of_experience	departmentID
E003	Mohammed	Al-Otaibi	M	1990-07-25	2014-06-01	0503456799	mohammed.alotaibi@example.com	PartTime	6	D003
E005	Yousef	Al-Mutairi	M	1982-09-15	2014-06-01	0505789801	yousef.almutairi@example.com	PartTime	9	D005
E008	Mona	Al-Harbi	F	1984-12-30	2015-11-01	0508911201	mona.alharbi@example.com	PartTime	5	D008
E010	Rashed	Al-Shahrani	M	1987-03-22	2014-06-01	0501234566	rashed.alshahrani@example.com	PartTime	2	D001
E014	Rami	Al-Shethri	F	1991-06-18	2013-05-01	0502134545	rami.alshethri@example.com	PartTime	9	D004
E018	Abdullah	Al-Shehri	M	1992-01-01	2014-06-01	0502134545	abdullah.alshehri@example.com	PartTime	5	D002

10. Retrieve the details of the most recent leave taken by each employee.

```
SELECT e.employeeID, e.fname, e.lname, pl.* FROM Employee e JOIN PaidLeave pl ON e.employeeID = pl.employeeID WHERE pl.start_date = (SELECT MAX(start_date) FROM PaidLeave pl2 WHERE pl2.employeeID = e.employeeID);
```

The screenshot shows the MySQL Workbench interface. The top navigation bar includes tabs for 'PSU Payroll (psupayroll)', 'psupayroll', and 'PSU Payroll'. The main menu has options like File, Edit, View, Query, Database, Server, Tools, Scripting, and Help. Below the menu is a toolbar with icons for various database operations. The left sidebar has sections for 'Navigator' (schemas, tables, objects), 'SCHEMAS' (selected 'psupayroll'), and a search bar ('Filter objects'). The central area contains a query editor with the following SQL code:

```
350 • SELECT e.employeeID, e.fname, e.lname, f.salary_per_day
351   FROM Employee e
352  JOIN FulltimeEmployee f ON e.employeeID = f.F_EmployeeID
353  WHERE f.salary_per_day > (
354      SELECT AVG(salary_per_day)
355        FROM FulltimeEmployee
356    )
357 • SELECT e.employeeID, e.fname, e.lname, pl.* FROM Employee e JOIN PaidLeave pl
358   ON e.employeeID = pl.employeeID WHERE pl.start_date = (SELECT MAX(start_date)
359   FROM PaidLeave pl2 WHERE pl2.employeeID = e.employeeID)
360
361
362
363
364
```

The bottom part of the interface shows a results grid titled 'Result Grid' with columns: employeeID, fname, lname, leaveID, employeeID, start_date, end_date, duration. The data is as follows:

employeeID	fname	lname	leaveID	employeeID	start_date	end_date	duration
E001	Abdullah	Al-Saud	L001	E001	2023-03-01	2023-03-10	10
E002	Fatmeh	Al-Faisal	L002	E002	2023-04-01	2023-04-05	5
E004	Noura	Al-Zebedi	L003	E004	2023-05-01	2023-05-15	15
E005	Huda	Al-Qahtani	L004	E005	2023-06-01	2023-06-07	7
E007	Saud	Al-Shamdi	L006	E007	2023-07-01	2023-07-02	2
E009	Saleh	Al-Sabaa	L006	E009	2023-08-01	2023-08-12	12
E010	Aisha	Al-Mansour	L007	E010	2023-09-01	2023-09-08	8
E012	Laila	Al-Zahrani	L008	E012	2023-10-01	2023-10-10	10
E013	Ahmed	Al-Shammary	L009	E013	2023-11-01	2023-11-05	5
E015	Sani	Al-Azaaz	L010	E015	2023-12-01	2023-12-12	12

