By:
Reema Binfoiz
Sarah Alojayan
Aljohara Alzaage
Tala Alhazmi
Dana Almashrafy

Payroll Management System Project Proposal

A database that helps sort PSU's payroll system

CS340 Project
For: Ms. Roohi Jan

Phases

Phase 1

Describing the application to be developed

Phase 2

ER/EER

Phase 3

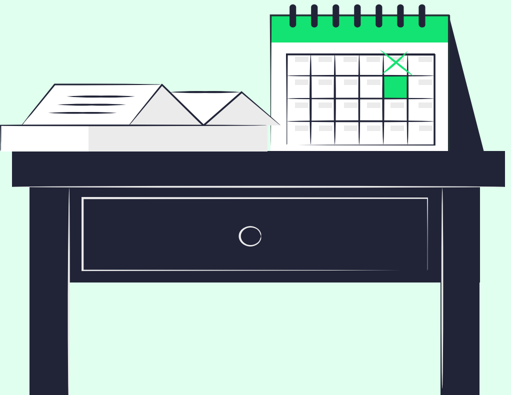
Designing the database

Phase 4

Creating tables

Phase 5

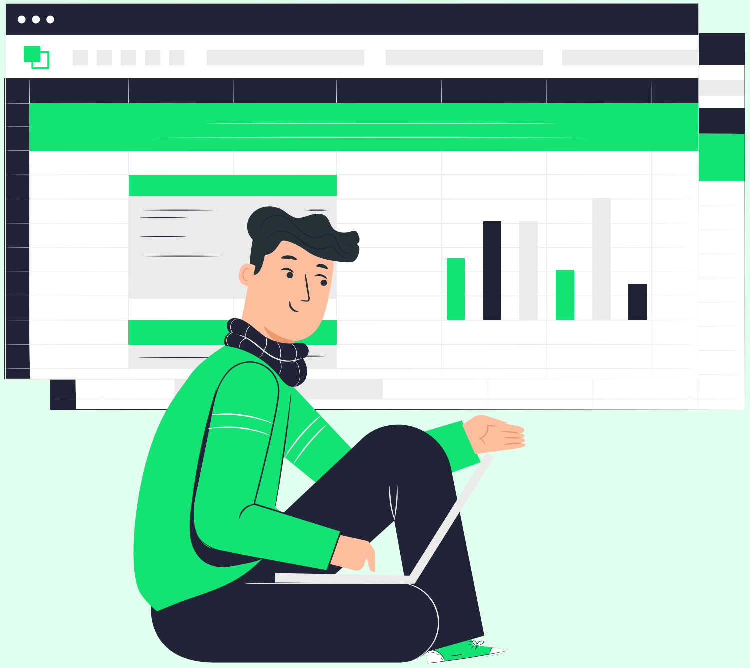
SQL queries



Contents of Phase 01

Introduction	What’s the point of this database, what purpose does it serve?
Background	Let's talk a bit more about what inspired this Project
Purpose and scope	What were the targets of this system and its scope
Functional & Non-functional	What requirements were needed for this Database?
Summary of Phase 1	Everything that was needed to be in Phase 1

Introduction



Our PSU payroll management system is concerning the data of all employees that work at PSU where it can provide the institute with a good way to handle how employees are paid and ease the job of managing names of all faculty and employees that work there.

Background

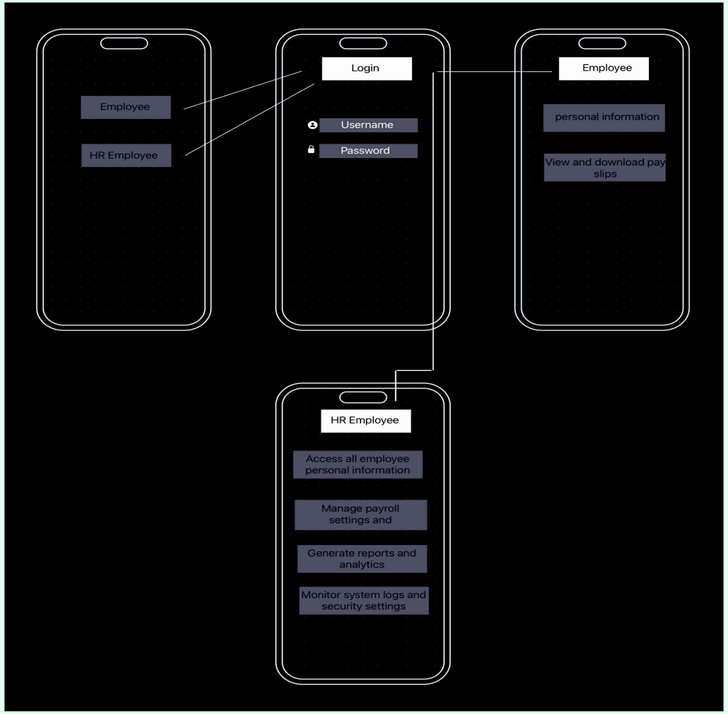
Prince Sultan University is an academic institution that values excellence and development above all else. As stakeholders and end-users, PSU is highly invested in the development and implementation of this system.

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.

Purpose and scope

Our aim is to automate PSU's salary process, streamlining payroll and HR tasks to boost efficiency, accuracy, and transparency in managing employee compensation. For a better understanding we've came up with a prototype for the system.

SUCH AS:


Employee
HR Employee
Login
Username
Password
Employee
personal information
View and download pay slips
HR Employee
Access all employee personal information
Manage payroll settings and
Generate reports and analytics
Monitor system logs and security settings

Functional & Non-Functional Requirements

Functional

User:
- New Account using PSU email and a safe password.
-The system shall handle deductions, bonuses, and allowances.
- The system shall allow employees to update their personal information.

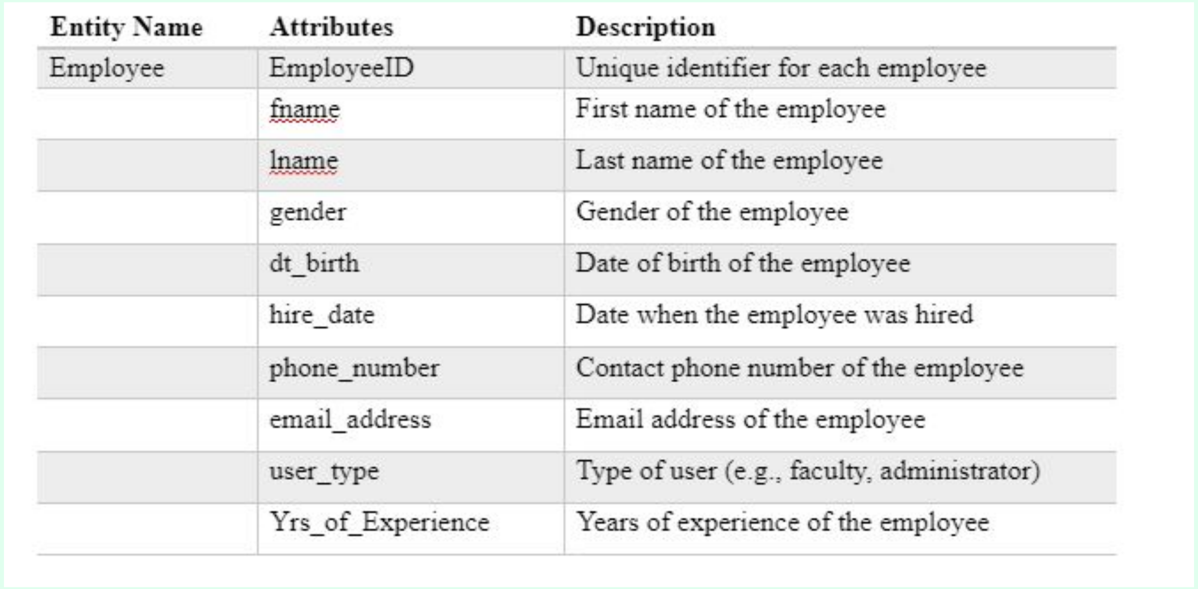
Administrator:
-The system shall allow the administrator to create a new account using the university email and password.
-The system shall allow access to employee information.
-The system shall generate reports on payroll, attendance, and employee information.

Non-Functional

- **Performance**
- **Scalability**
- **Reliability**
- **Security**
- **Usability**

Entities

This is a sample of what the entity table looks like in phase 1:

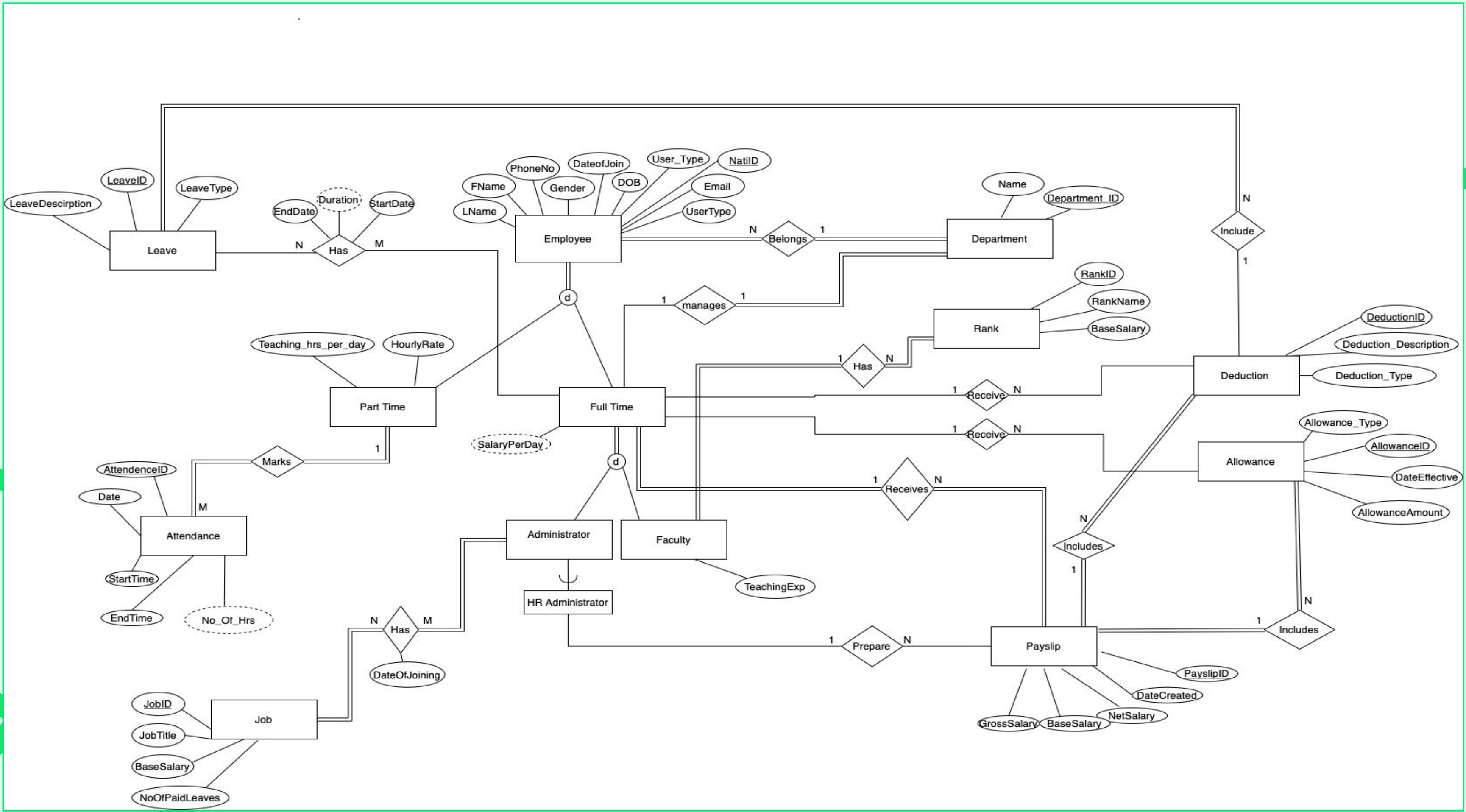


Entity Name	Attributes	Description
Employee	EmployeeID	Unique identifier for each employee
	fname	First name of the employee
	lname	Last name of the employee
	gender	Gender of the employee
	dt_birth	Date of birth of the employee
	hire_date	Date when the employee was hired
	phone_number	Contact phone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

Contents of Phase 02

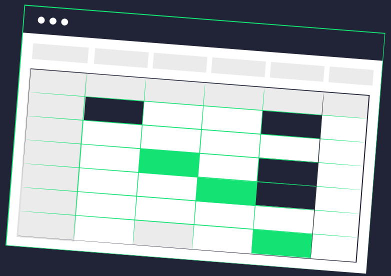
EER Diagram	visually represents a database's entities, relationships, and constraints.
Entities and attributes	Shows each entity and its attributes
Constraint	specify the rules and conditions that govern the relationships and interactions between entities, ensuring that the database accurately reflects policies and processes.

EER Diagram


Leave
LeaveID
LeaveType
LeaveDescirption
Has
N
M
EndDate
Duration
StartDate
Employee
PhoneNo
DateofJoin
User_Type
NatiID
FName
Gender
DOB
Email
LName
UserType
Belongs
N
1
Department
Name
Department_ID
Include
N
1
d
manages
1
1
Rank
RankID
RankName
BaseSalary
Teaching_hrs_per_day
HourlyRate
Part Time
Full Time
SalaryPerDay
Has
1
N
Deduction
DeductionID
Deduction_Description
Deduction_Type
Receive
1
N
Receive
1
N
Allowance
Allowance_Type
AllowanceID
DateEffective
AllowanceAmount
Marks
1
M
AttendenceID
Date
Attendance
StartTime
EndTime
No_Of_Hrs
d
Receives
1
N
Administrator
Faculty
TeachingExp
HR Administrator
Includes
N
1
Includes
N
1
Has
N
M
DateOfJoining
Prepare
1
N
Payslip
PayslipID
DateCreated
NetSalary
GrossSalary
BaseSalary
Job
JobID
JobTitle
BaseSalary
NoOfPaidLeaves

Entities and Attributes:

- **Employee**: Attributes include PhoneNo, Gender, Name, Email, BirthDate, User_Type, etc.
- **Department**: Attributes include Name and Department_ID.
- **Rank**: Attributes include RankID, RankName, BaseSalary.
- **Leave**: Attributes include LeaveDescription and LeaveType.
- **Attendance**: Attributes include Date, StartTime, EndTime, and No_Of_Hrs.
- **Job**: Attributes include JobID, JobTitle, BaseSalary, NoOfPaidLeaves.
- **Allowance**: Attributes include AllowanceType, AllowanceAmount, DateEffective.
- **Deduction**: Attributes include DeductionType, Deduction_Description.
- **Payslip**: Attributes include GrossSalary, BaseSalary, NetSalary, DateCreated, PayslipID.



Business Constraints

A part-timer get paid only by hours worked, unlike Full-timers

Deduction is done if an employee misses their work day

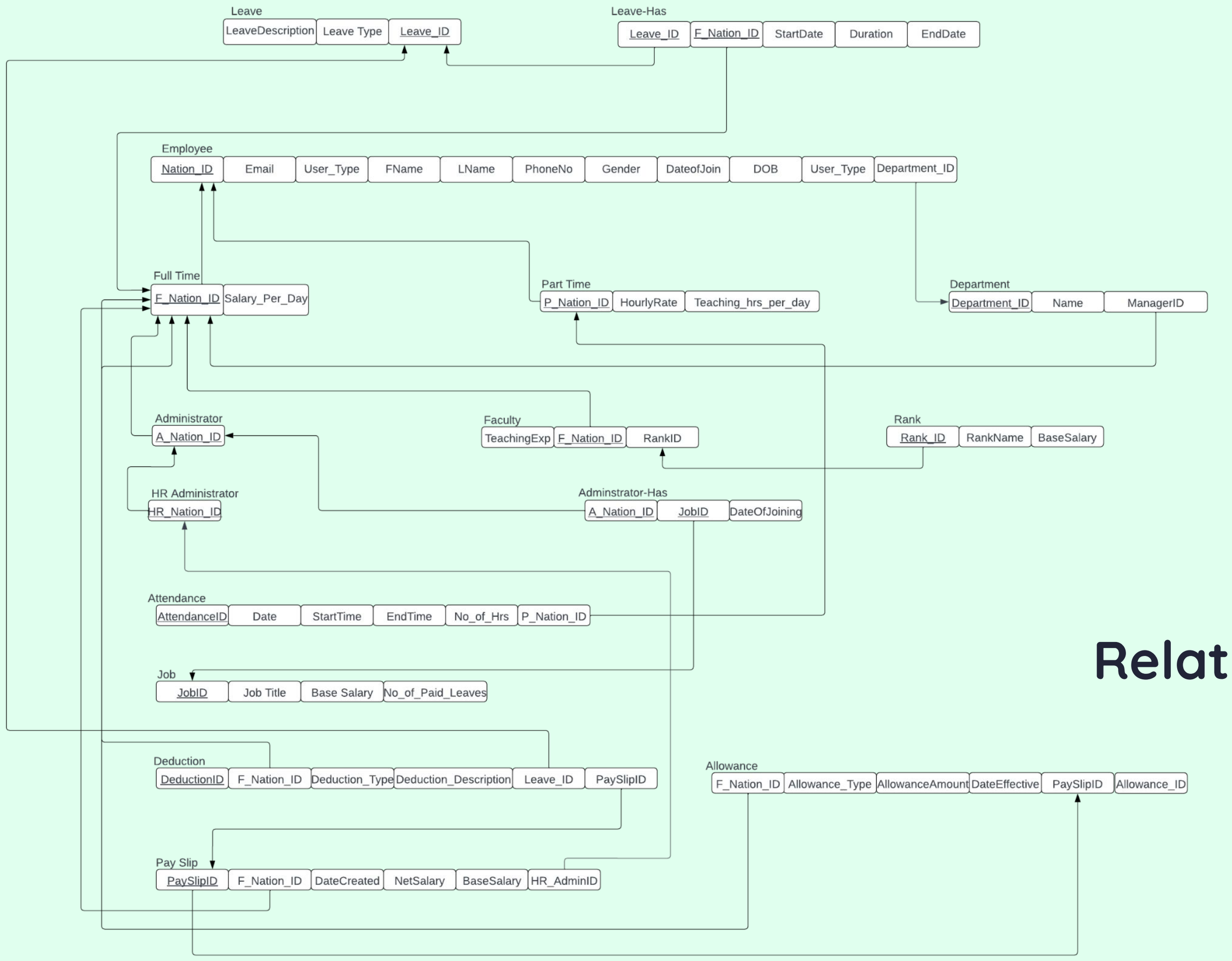
Only full-timers get monthly allowance

High turnover rates can increase the workload for payroll administrators, requiring constant updates to employee records and payroll calculations.

Contents of Phase 03

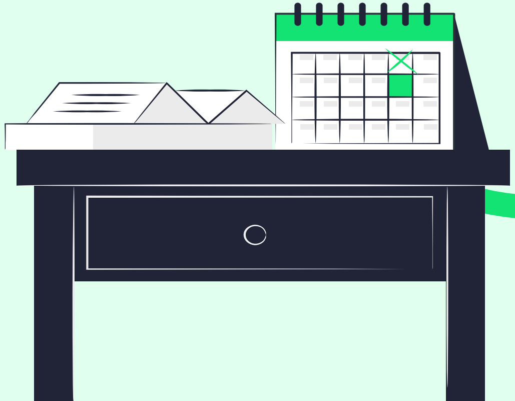
RELATIONAL MODEL

Relational Model


Leave
LeaveDescription | Leave Type | Leave_ID
Leave-Has
Leave_ID | F_Nation_ID | StartDate | Duration | EndDate
Employee
Nation_ID | Email | User_Type | FName | LName | PhoneNo | Gender | DateofJoin | DOB | User_Type | Department_ID
Full Time
F_Nation_ID | Salary_Per_Day
Part Time
P_Nation_ID | HourlyRate | Teaching_hrs_per_day
Department
Department_ID | Name | ManagerID
Administrator
A_Nation_ID
Faculty
TeachingExp | F_Nation_ID | RankID
Rank
Rank_ID | RankName | BaseSalary
HR Administrator
HR_Nation_ID
Adminstrator-Has
A_Nation_ID | JobID | DateOfJoining
Attendance
AttendanceID | Date | StartTime | EndTime | No_of_Hrs | P_Nation_ID
Job
JobID | Job Title | Base Salary | No_of_Paid_Leaves
Deduction
DeductionID | F_Nation_ID | Deduction_Type | Deduction_Description | Leave_ID | PaySlipID
Allowance
F_Nation_ID | Allowance_Type | AllowanceAmount | DateEffective | PaySlipID | Allowance_ID
Pay Slip
PaySlipID | F_Nation_ID | DateCreated | NetSalary | BaseSalary | HR_AdminID

Contents of Phase 04

MySQL Table Creation	Implement EER Diagram into a DBMS, creating tables for each entity.



Database and Table Creation in MySQL

- PSU payroll management system was implemented MySQL Workspace 6.0

- MySQL Code:

```
CREATE DATABASE PSUPayroll;
USE PSUPayroll;
```

- Department's Table:

```
CREATE TABLE Department(
departmentID VARCHAR(10) PRIMARY KEY,
name VARCHAR(15) NOT NULL
);
```

- Job's Table:

```
CREATE TABLE Job(
jobID VARCHAR(10) PRIMARY KEY,
job_title VARCHAR(50) NOT NULL,
base_salary VARCHAR(50),
num_paid_leaves INT
);
```

Table Creation in MySQL

- JobRank's Table:

```
CREATE TABLE JobRank(
rankID VARCHAR(10) PRIMARY KEY,
rank_name VARCHAR(20) NOT NULL,
base_salary VARCHAR(50) NOT NULL
);
```

- Faculty's Table:

```
CREATE TABLE Faculty(
employeeID VARCHAR(10) PRIMARY KEY,
teaching_experience VARCHAR(3) NOT
NULL,
rankID VARCHAR(10) UNIQUE NOT NULL,
FOREIGN KEY (employeeID) REFERENCES
Employee(employeeID),
FOREIGN KEY (rankID) REFERENCES
JobRank(rankID)
);
```

- Employee's Table:

```
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,
natID VARCHAR(20)
);
```

Table Creation in MySQL

- Administrator's Table:

```
CREATE TABLE Administrator(
employeeID VARCHAR(10) PRIMARY KEY,
is_HR_administrator VARCHAR(3) NOT NULL,
salary_per_day VARCHAR(50) NOT NULL,
FOREIGN KEY (employeeID) REFERENCES
Employee(employeeID)
);
```

- Allowance's Table:

```
CREATE TABLE Allowance (
    allowanceID VARCHAR(10) PRIMARY KEY,
    allowanceType VARCHAR(20) NOT NULL,
    allowance_amount VARCHAR(15) NOT
NULL,
    effectiveDate DATE NOT NULL
);
```

- PartTimeFaculty's Table:

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

- FullTimeFaculty's Table:

```
CREATE TABLE FullTimeFaculty (
employeeID VARCHAR(10) PRIMARY KEY,
 academic_rankID VARCHAR(10),
 FOREIGN KEY (employeeID) REFERENCES
Employee(employeeID),
 FOREIGN KEY (academic_rankID) REFERENCES
JobRank(rankID)
);
```

Table Creation in MySQL

- PaySlip's Table:

```
CREATE TABLE PaySlip (
    paySlipID VARCHAR(10) PRIMARY KEY,
    employeeID VARCHAR(10) UNIQUE NOT
NULL,
    base_salary VARCHAR(10) NOT NULL,
    allowance_amount VARCHAR(10) NOT NULL,
    deduction_amount VARCHAR(10) NOT NULL,
    gross_salary VARCHAR(10) NOT NULL,
    net_salary VARCHAR(10)  NOT NULL,
    date_created DATE NOT NULL,
    FOREIGN KEY (employeeID) REFERENCES
Employee(employeeID),
);
```

- PaidLeave's Table:

```
CREATE TABLE PaidLeave (
    leaveID VARCHAR(10) PRIMARY KEY,
    employeeID VARCHAR(10) UNIQUE NOT NULL,
    leave_type VARCHAR(10) NOT NULL,
    duration INT NOT NULL,
    FOREIGN KEY (employeeID) REFERENCES
Employee(employeeID),
);
```

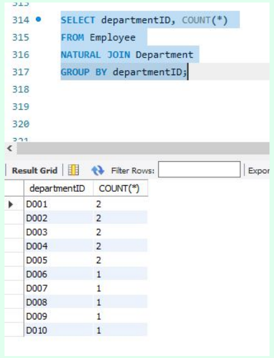
Contents of Phase 05

SQL Queries	Basic Queries
	Advanced Queries
	Netbeans INSERT
	Netbeans UPDATE
	Netbeans DELETE
	Netbeans DISPLAY

Basic Queries

Count the number of employees in each department.

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

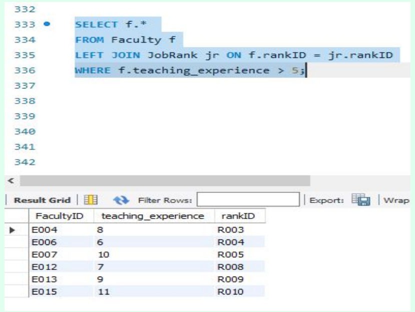


```
SELECT departmentID, COUNT(*)
FROM Employee
NATURAL JOIN Department
GROUP BY departmentID;
```

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

Retrieve all faculty members with more than 5 years of teaching experience.

```
SELECT f.*
FROM Faculty f
LEFT JOIN JobRank jr ON f.rankID = jr.rankID
WHERE f.teaching_experience > 5;
```

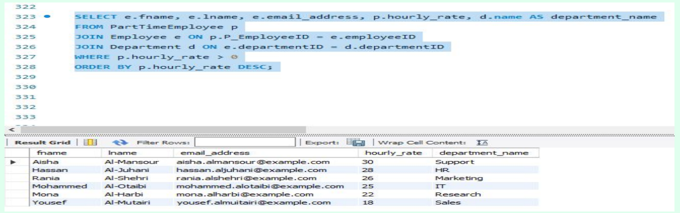


```
SELECT f.*
FROM Faculty f
LEFT JOIN JobRank jr ON f.rankID = jr.rankID
WHERE f.teaching_experience > 5;
```

FacultyID	teaching_experience	rankID
E004	8	R003
E006	6	R004
E007	10	R005
E012	7	R008
E013	9	R009
E015	11	R010

List all part-time employees and their hourly rates.

```
SELECT e.fname, e.lname, e.email_address, p.hourly_rate, d.name AS department_name
FROM PartTimeEmployee p
JOIN Employee e ON p.P_EmployeeID = e.employeeID
JOIN Department d ON e.departmentID = d.departmentID
WHERE p.hourly_rate > 0
ORDER BY p.hourly_rate DESC;
```



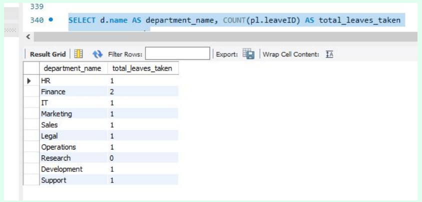
```
SELECT e.fname, e.lname, e.email_address, p.hourly_rate, d.name AS department_name
FROM PartTimeEmployee p
JOIN Employee e ON p.P_EmployeeID = e.employeeID
JOIN Department d ON e.departmentID = d.departmentID
WHERE p.hourly_rate > 0
ORDER BY p.hourly_rate DESC;
```

fname	lname	email_address	hourly_rate	department_name
Aisha	Al-Mansour	aisha.almansour@example.com	30	Support
Hassan	Al-Juhani	hassan.aljuhani@example.com	28	HR
Rania	Al-Shehri	rania.alshehri@example.com	26	Marketing
Mohammed	Al-Otaibi	mohammed.alotaibi@example.com	25	IT
Mona	Al-Harbi	mona.alharbi@example.com	22	Research
Yousef	Al-Mutairi	yousef.almuitairi@example.com	18	Sales

Advanced Queries

Find employees who have not received any allowances:

```
SELECT e.FName, e.LName
FROM Employee e
WHERE NOT EXISTS (
    SELECT 1
    FROM Pay_Slip p
    JOIN Allowance a ON p.PaySlipID = a.PaySlipID
    WHERE p.F_Nation_ID = e.Nation_ID
);
```

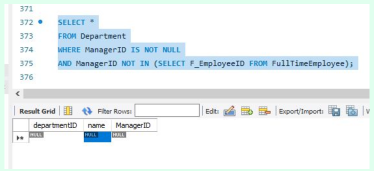


```
SELECT d.name AS department_name, COUNT(pl.leaveID) AS total_leaves_taken
```

department_name	total_leaves_taken
HR	1
Finance	2
IT	1
Marketing	1
Sales	1
Legal	1
Operations	1
Research	0
Development	1
Support	1

Show the departments that do not have any full-time employees as managers.

```
SELECT * FROM Department WHERE ManagerID NOT
IN (SELECT F_EmployeeID FROM
FullTimeEmployee);
```

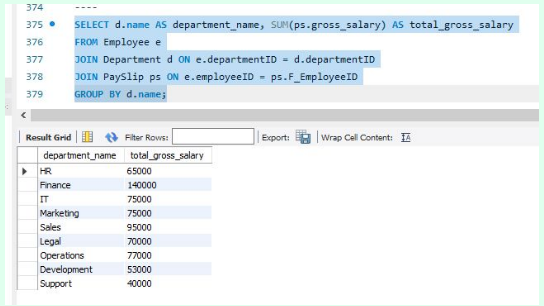


```
SELECT *
FROM Department
WHERE ManagerID IS NOT NULL
AND ManagerID NOT IN (SELECT F_EmployeeID FROM FullTimeEmployee);
```

departmentID	name	ManagerID
NULL	NULL	NULL

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;
```

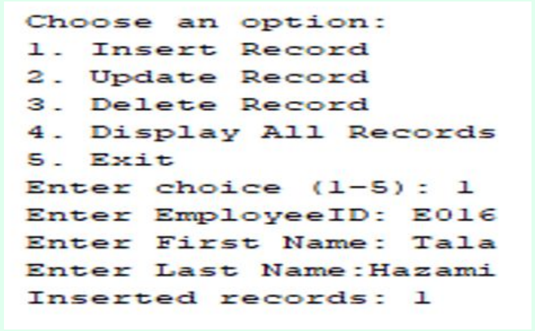


```
----
SELECT d.name AS department_name, SUM(ps.gross_salary) AS total_gross_salary
FROM Employee e
JOIN Department d ON e.departmentID = d.departmentID
JOIN PaySlip ps ON e.employeeID = ps.F_EmployeeID
GROUP BY d.name;
```

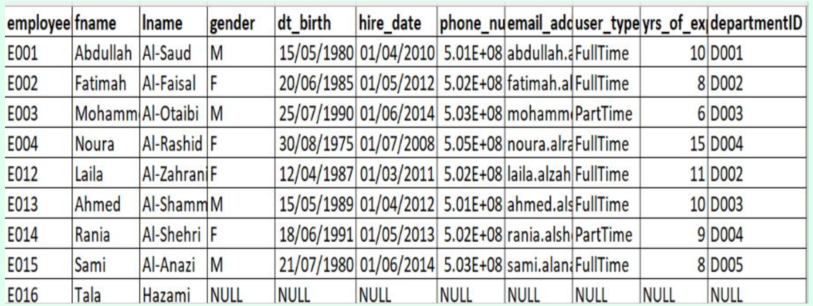
department_name	total_gross_salary
HR	65000
Finance	140000
IT	75000
Marketing	75000
Sales	95000
Legal	70000
Operations	77000
Development	53000
Support	40000

Netbeans(INSERT, UPDATE)

INSERT RECORDS

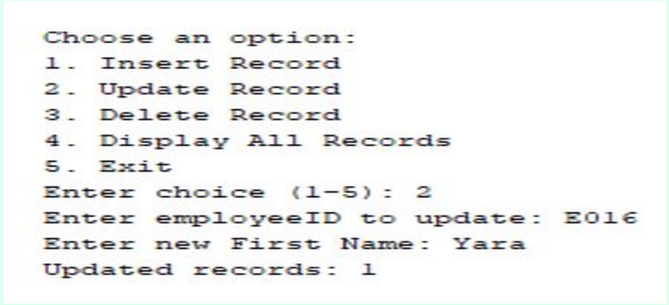


```
Choose an option:
1. Insert Record
2. Update Record
3. Delete Record
4. Display All Records
5. Exit
Enter choice (1-5): 1
Enter EmployeeID: E016
Enter First Name: Tala
Enter Last Name:Hazami
Inserted records: 1
```

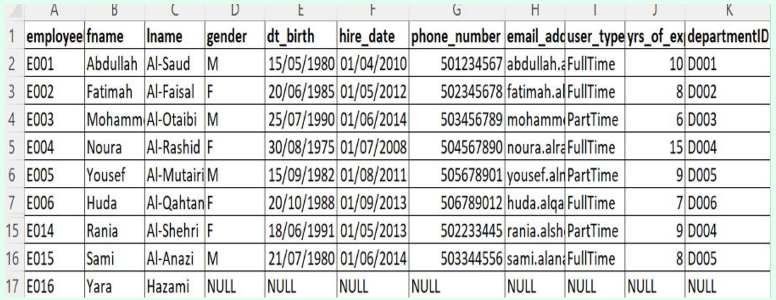


employee	fname	lname	gender	dt_birth	hire_date	phone_nu	email_adc	user_type	yrs_of_ex	departmentID
E001	Abdullah	Al-Saud	M	15/05/1980	01/04/2010	5.01E+08	abdullah.a	FullTime	10	D001
E002	Fatimah	Al-Faisal	F	20/06/1985	01/05/2012	5.02E+08	fatimah.al	FullTime	8	D002
E003	Mohamm	Al-Otaibi	M	25/07/1990	01/06/2014	5.03E+08	mohamm	PartTime	6	D003
E004	Noura	Al-Rashid	F	30/08/1975	01/07/2008	5.05E+08	noura.alra	FullTime	15	D004
E012	Laila	Al-Zahrani	F	12/04/1987	01/03/2011	5.02E+08	laila.alzah	FullTime	11	D002
E013	Ahmed	Al-Shamm	M	15/05/1989	01/04/2012	5.01E+08	ahmed.als	FullTime	10	D003
E014	Rania	Al-Shehri	F	18/06/1991	01/05/2013	5.02E+08	rania.alsh	PartTime	9	D004
E015	Sami	Al-Anazi	M	21/07/1980	01/06/2014	5.03E+08	sami.alana	FullTime	8	D005
E016	Tala	Hazami	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

UPDATE RECORDS



```
Choose an option:
1. Insert Record
2. Update Record
3. Delete Record
4. Display All Records
5. Exit
Enter choice (1-5): 2
Enter employeeID to update: E016
Enter new First Name: Yara
Updated records: 1
```

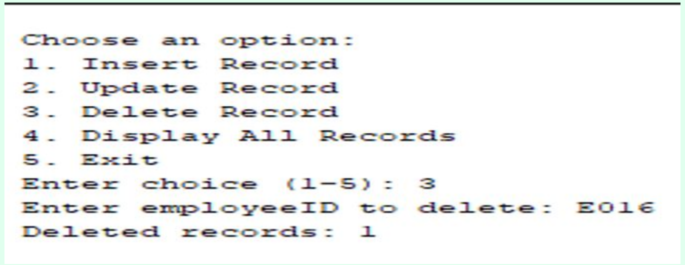


	A	B	C	D	E	F	G	H	I	J	K
1	employee	fname	lname	gender	dt_birth	hire_date	phone_number	email_adc	user_type	yrs_of_ex	departmentID
2	E001	Abdullah	Al-Saud	M	15/05/1980	01/04/2010	501234567	abdullah.a	FullTime	10	D001
3	E002	Fatimah	Al-Faisal	F	20/06/1985	01/05/2012	502345678	fatimah.al	FullTime	8	D002
4	E003	Mohamm	Al-Otaibi	M	25/07/1990	01/06/2014	503456789	mohamm	PartTime	6	D003
5	E004	Noura	Al-Rashid	F	30/08/1975	01/07/2008	504567890	noura.alra	FullTime	15	D004
6	E005	Yousef	Al-Mutairi	M	15/09/1982	01/08/2011	505678901	yousef.aln	PartTime	9	D005
7	E006	Huda	Al-Qahtan	F	20/10/1988	01/09/2013	506789012	huda.alqa	FullTime	7	D006
15	E014	Rania	Al-Shehri	F	18/06/1991	01/05/2013	502233445	rania.alsh	PartTime	9	D004
16	E015	Sami	Al-Anazi	M	21/07/1980	01/06/2014	503344556	sami.alana	FullTime	8	D005
17	E016	Yara	Hazami	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

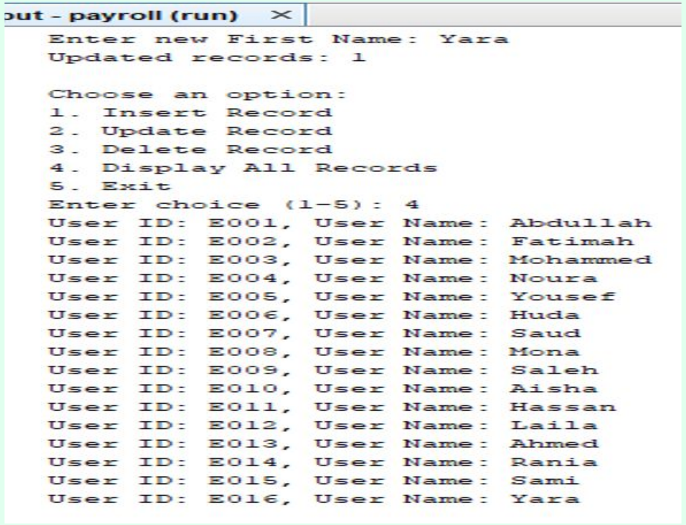
Netbeans(DISPLAY, DELETE)

DISPLAY RECORDS

DELETE RECORDS

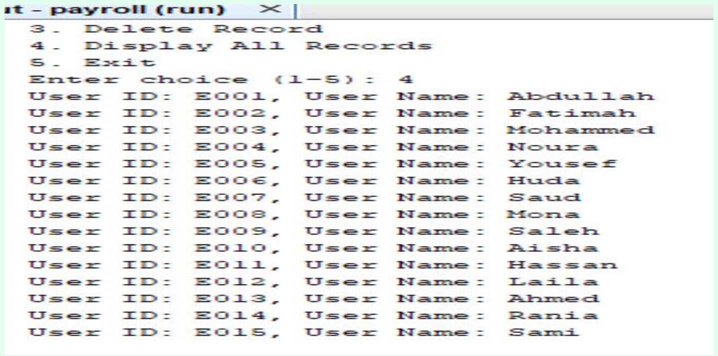


```
Choose an option:
1. Insert Record
2. Update Record
3. Delete Record
4. Display All Records
5. Exit
Enter choice (1-5): 3
Enter employeeID to delete: E016
Deleted records: 1
```



```
Enter new First Name: Yara
Updated records: 1

Choose an option:
1. Insert Record
2. Update Record
3. Delete Record
4. Display All Records
5. Exit
Enter choice (1-5): 4
User ID: E001, User Name: Abdullah
User ID: E002, User Name: Fatimah
User ID: E003, User Name: Mohammed
User ID: E004, User Name: Noura
User ID: E005, User Name: Yousef
User ID: E006, User Name: Huda
User ID: E007, User Name: Saud
User ID: E008, User Name: Mona
User ID: E009, User Name: Saleh
User ID: E010, User Name: Aisha
User ID: E011, User Name: Hassan
User ID: E012, User Name: Laila
User ID: E013, User Name: Ahmed
User ID: E014, User Name: Rania
User ID: E015, User Name: Sami
User ID: E016, User Name: Yara
```



```
3. Delete Record
4. Display All Records
5. Exit
Enter choice (1-5): 4
User ID: E001, User Name: Abdullah
User ID: E002, User Name: Fatimah
User ID: E003, User Name: Mohammed
User ID: E004, User Name: Noura
User ID: E005, User Name: Yousef
User ID: E006, User Name: Huda
User ID: E007, User Name: Saud
User ID: E008, User Name: Mona
User ID: E009, User Name: Saleh
User ID: E010, User Name: Aisha
User ID: E011, User Name: Hassan
User ID: E012, User Name: Laila
User ID: E013, User Name: Ahmed
User ID: E014, User Name: Rania
User ID: E015, User Name: Sami
```

Thank You!

Do you have any questions?