جامعة الأمير سلطان
PRINCE SULTAN UNIVERSITY

مُجاب

‘Mojab’

COMPLAINT MANAGEMENT SYSTEM

Introduction to Database Systems
CS340

Done by :
Lama Alghzzi 220410092
Dana Alamri 220410341
Sultanah Alsubaie 219410354

Under the supervision of :
Ms. Roohi Jan

Section: 796

Phase 1
Describing the Information System to Be Developed

Table of Contents

Introduction

This document is about the "Mojab" system project. This application is a complaint management system. In this document, we cover 5 phases, Describing the Information System to be developed, ER/EER Modelling, Designing Database and Normalization, Creating Tables using Oracle 10g and Meeting the Specification.

Description of the Organization

Mojab in Arabic means answered, respondent and considered. This system was created for Prince Sultan University in Riyadh, which is a private, non-profit university that provides study in many important programs. Based on the university's goals to respond to students and employees and raise the quality of the educational level and the satisfaction of students and employees with the services provided to them, this system was created. The system allows students and employees to raise complaints related to the university, direct them to departments, respond to users as quickly as possible, and evaluate their satisfaction. The need for this application came with the recent increase in the number of students and employees, which makes the traditional way -the manual way- to address complaints more difficult and takes a longer time. Also, the student or employee may not know who is responsible for addressing this complaint or which department it relates to. However, with an application that assigns complaints to the departments according to their responsibilities, this will save effort and time instead of the traditional way of filing and tracking the complaints. In addition, this application provides a greater advantage and meets

the university's need to know the most prominent difficulties that students and employees face. Using this application makes it easier for the university to gather statistics for complaints, classify them, and review them to develop the level of service provided, raise the quality of the university, and achieve part of the university's mission, which aims to integrate modern technologies to enhance productivity and serve beneficiaries.

Description of the purpose and scope of the IS

This application is limited to students and employees of Prince Sultan University in Riyadh. The purpose of this application is to increase the quality and speed of response to the problems and requests of employees and students. This application enhances the service automation system. Instead of the difficulty of physical visits to department heads and officials, this system came to speed up and organize the way to raise complaints and respond to them. A variety of interfaces is provided for each of these categories:

The User view:

- Login and Signup page.
- Personal Information page.
- Page with 3 options (Previous complaints, Add a new complaint and Track a complaint).
- Page to add a new complaint which included (Write your complaint, Type of complaint, Complaint level and Upload an attachments).
- Page to see the complaint state and the expected time to solve the problem.
- The satisfaction page (when the complaint is closed).

The Administrator view:

- **Login and Signup page.**
- **A dashboard that shows all the complaints.**
- **A page to control and manage the complaints states and due dates.**
- **A page to assigns the complaint to the appropriate department and to show who assigns the complaints to the departments**

The Department view:

- **Login and Signup page.**
- **A dashboard to see all complaints and solve them.**

Functional System Requirements:

User :

- **The system will allow the users to register with PSU ID, email, phone number and their own password.**
- **The system then will verify the registration and each login process by sending an OTP to the user phone number(It will not be implemented in this project).**
- **The system will provide each user a personal profile.**
- **The system will allow the user to write the complaint.**
- **The system will make the student choose the desired department.**
- **The system will notify the user that the complaint was received.**
- **The user will be able to check the complaint state all the time.**
- **After receiving the reply from the department, the system will provide a page to rate the service.**

Administrator :

- **The system will provide the administrator with the ticket.**
- **The administrator will analyze the complaint.**

- **The administrator will send the ticket to the desired department.**

Department:

- **The department head will receive the ticket from the administrator.**
- **The department head will assess the complaint and solve the problem.**
- **Then the department will notify the administrator that the problem is solved.**
- **If the user is not satisfied the department will receive the problem again and check it.**

Non-Functional System Requirements :

1- Performance and Scalability:

- **The system shall function normally no matter how many users are accessing the system at the same time.**
- **The response time of any operation the user requests or the page loads should not take more than 5 seconds.**
- **The system should have a strong recovery plan if there are any lagging or blocking problems.**

2- Usability:

- **From the first time, the system should be easy to use and remember.**
- **Accessing all the program functions within the interface should be obvious to the user.**

3- Security:

- **The system should allow access to the authorized people only.**
- **A verification number with each login or signup process.**
- **Using VPN which include strong encryption algorithms to ensure the highest level of security within the connection and network.**

- **The passwords must be strong and unguessable.**

4- Extensibility and maintainability:

- The mean time to restore the system "MTTRS" after any failure should not take more than 10 minutes.
- Adding new features to the existing system should not affect the whole system.

5- Reliability:

- During support access the downtime of the system should be no more than 15 minutes per 30 days.

Data Requirements

a. Scenario:

1. The user can be either a student or employee, which all inherit from the User. Users are distinguished by their User_ID. Users are also identified by Email, Password, Phone and Name. The name is composed of first name and last name.

2. Students are distinguished by User_ID. Students are also identified by Student_ID, Student_major and Academic_level.

3. Employees are distinguished by User_ID. Employees are also identified by Employee_ID, Job_position and Rank.

4. Administrators are distinguished by Admin_ID, Email and Password.

5. Departments are distinguished by Departments_ID. Departments are identified by Department_Name, Email and Password.

6. An employee works in one department. A department has many employees working in it.

7. An employee may manage one department and one department must have one employee who manages the department.

8. Complaints are distinguished by Complaint_ID. Complaints are identified by Complaint_Type, Complaint_Text, Status, Complaint_level.

9. A complaint may be related to one or many locations. A Location is described by the Location_ID, BuildingNum, RoomNo, FloorNumber, Type of location such as classroom, computer labs, offices, student lounge, library, meeting rooms etc.. A location may be shown in many complaints.

10. The user files a complaint. The user can file any number of complaints. A complaint is filed by one user. The date of a complaint is recorded.

11. The administrator assigns the complaint to the department depending on the type of the complaint. The date of assignment is recorded, and the Department will add the solved date .

12. The department will address the complaint. The department will add date_solved and status of the complaint is updated. It can be solved, cannot be solved, in progress.

13. After the complaint is closed, the user can fill out the complaint satisfaction rate and comments.

b. Entity Definition Table

Entity	Description		Identifier	Attribute
User	Basic entity contains information about all PSU users that are using the application (Student and employee)		User_ID	User_ID Email Password Phone Name (F_name, L_name)
Student	Inherited entity, it will contain information from the user in addition to student ID, major, academic level		User_ID	User_ID Student_ID Student_major Academic_level
Employee	Inherited entity that will receive information from the user in addition to employee ID, job position and rank		User_ID	User_ID Employee_ID Job_position Rank Email Password
Administrator	An entity that controls complaints filed by the		Admin_ID	Admin_ID Email Password

Entity	Description		Identifier	Attribute
	user and assigns them to the departments			
Department	An entity representing the departments in the PSU that address complaints		Department_ID	Department_ID Department_Name
Complaint	An entity that represents the complaint filed by the user		Complaint_ID	Complaint_ID Status Complaint_TypeComplaint_Text Complaint_level
Location	An entity that represents a location for the complaint		Location_ID	Location_ID BuildingNum RoomNo FloorNumber Location_Type

c. **Relation Definition Table:**

Relationship Name	Type	Attributes	Entities	Description
IS-A	1:1 Parent-child single inheritance relationship Disjoint, TP		User, Student Employee	The user can be either a student or employee or an administrator
Files	Binary, 1:N, TP:PP	Complaint_Date	User and Complaint	This relationship allows "user" to be linked with "Complaint" to file a complaint, complaint date is recorded when the user files it
Assigns	Tertiary, 1:N:1, TP,TP,PP	Assignment_Date Date_Solved	Administrator, Complaint and Department	This relationship allows "Department" to be linked with "Complaint" by the "Administrator", the date of assignment is recorded and

Relationship Name	Type	Attributes	Entities	Description
				department will add the solved date
Fill out	Binary, 1:N,PP:PP	Comments Satisfaction_ Rate	User and Complaint	This relationship allows "Complaint" to be evaluated by the "User", user can fill out the complaint satisfaction rate and comments for the complaint
Works	Binary, N:1,TP:TP		Employee and Department	This relationship allows "Employee" to be linked with "Department"
Managed by	Binary, 1:1,PP:TP		Employee and Department	This relationship allows "Employee" to manage the "Department"
Related	Binary, N-1,PP:TP		Complaint and Location	This relationship allows "Complaint"

Relationship Name	Type	Attributes	Entities	Description
				to be related to “Location”

d. **Attribute Definition Table:**

Entity Name	Attribute	Type	Description
User	User_ID	Simple Key Attribute	The identification number assigned to the user
	Email	Simple Attribute	The email assigned to the user
	Password	Simple Attribute	The password chosen by the user to log in
	Name (F_name, L_name)	Composite Attribute	The full name for the user
	Phone	Simple Attribute	Phone number that is registered with the user account
Student	Student_ID	Simple Key Attribute	The identification number assigned to PSU students
	Student_major	Simple Attribute	The major of the student
	Academic_level	Simple Attribute	Academic level for the student

Entity Name	Attribute	Type	Description
Employee	Employee_ID	Simple Key Attribute	The identification number assigned to PSU employees
	Job_position	Simple Attribute	Job position of the employee
	Rank	Simple Attribute	Rank of the employee in PSU
Department	Department_ID	Simple Key Attribute	The identification number assigned to PSU departments
	Department_Name	Simple Attribute	Department name in psu
	Email	Simple Attribute	The email assigned to the department
	Password	Simple Attribute	The password chosen by the department to log in
Administrator	Admin_ID	Simple Key Attribute	The identification number assigned to Mogab Administrator
	Email	Simple Attribute	The email assigned to the administrator
	Password	Simple Attribute	The password chosen by the administrator to log in
	Complaint_ID	Simple Key	The identification

Entity Name	Attribute	Type	Description
Complaint		Attribute	number assigned for each complaint
	Status	Simple Attribute	The status of the complaint
	Complaint_Type	Simple Attribute	The type of the complaint (financial, academic, resources, facilities)
	Complaint_Text	Simple Attribute	Text of the complaint and attachments
	Complaint_level	Simple Attribute	Classification of the complaint as whether it is important and urgent or not
Location	Location_ID	Simple Key Attribute	The identification number assigned for each location
	BuildingNum	Simple Attribute	Building Number at PSU
	RoomNo	Simple Attribute	Room Number at PSU
	FloorNumber	Simple Attribute	Floor Number at PSU

Entity Name	Attribute	Type	Description
	Location_Type	Simple Attribute	Location type such as; classroom, computer labs, offices, student lounge, library, meeting rooms

Identity relevant clauses from the IEEE and ACM Professional Code of Ethics in database analysis :

IEEE code of ethics:

- To avoid unlawful conduct in professional activities.
- To treat all persons fairly and with respect.
- To support colleagues and co-workers in following this code of ethics.
- To maintain and improve our technical competence.
- To seek, accept, and offer honest criticism of technical work.
- To prioritize the safety, health, and welfare of the public.
- To Avoid deceptive acts.

ACM code of ethics:

1. General Ethical Principles

1.1 Contribute to society and to human well-being, acknowledging that all people are stakeholders in computing.

1.2 Avoid harm.

1.3 Be honest and trustworthy.

1.4 Be fair and take action not to discriminate.

1.6 Respect privacy.

1.7 Honor confidentiality.

2. Professional Responsibilities

2.1 Strive to achieve high quality in both the processes and products of professional work.

2.2 Maintain high standards of professional competence, conduct, and ethical practice.

2.3 Know and respect existing rules pertaining to professional work.

2.8 Access computing and communication resources only when authorized or when compelled by the public good.

2.9 Design and implement systems that are robustly and usably secure.

3. Professional Leadership Principles

3.2 Articulate, encourage acceptance of, and evaluate fulfillment of social responsibilities by members of the organization or group.

3.3 Manage personnel and resources to enhance the quality of working life.

3.4 Articulate, apply, and support policies and processes that reflect the principles of the Code.

3.5 Create opportunities for members of the organization or group to grow as professionals.

4. Compliance with The Code

4.1 Uphold, promote, and respect the principles of the Code.

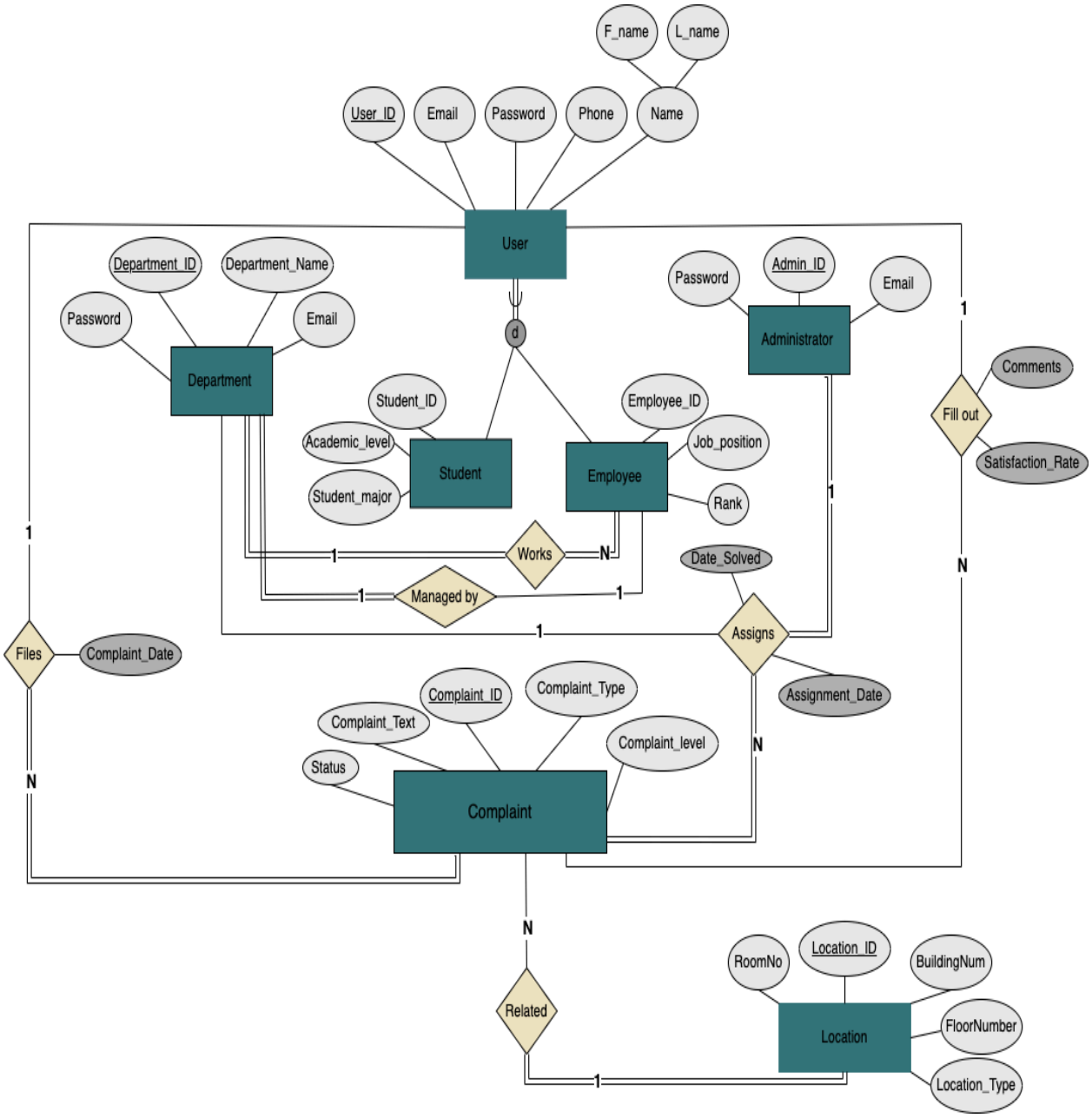
4.2 Treat violations of the Code as inconsistent with membership in the ACM.

Contribution of the team members to phase 1 and strategy used to demonstrate teamwork :

Dana	**Functional System Requirement** **Non-Functional** **System Requirements** **Identity relevant clauses from the IEEE and ACM Professional Code of Ethics in database analysis**
Lama	**Description of the Organization** **Description of the purpose and scope of the IS** **Data Requirements**

Phase 2
ER/EER Modeling

EER Diagram:


F_name
L_name
User_ID
Email
Password
Phone
Name
User
Department_ID
Department_Name
Password
Email
Department
d
Password
Admin_ID
Email
Administrator
Comments
Fill out
Satisfaction_Rate
Student_ID
Academic_level
Student_major
Student
Employee_ID
Job_position
Rank
Employee
Works
Managed by
Date_Solved
Assigns
Assignment_Date
Files
Complaint_Date
Complaint_ID
Complaint_Type
Complaint_Text
Complaint_level
Status
Complaint
Related
RoomNo
Location_ID
BuildingNum
FloorNumber
Location_Type
Location
1
N

Business Rules:

1- The user password must be at most 10 characters including a lowercase letter, uppercase letter, and numbers.

2- The user id must be at most 10 characters.

3- One user can upload complaints an infinite number of times.

4- Many complaints will be assigned by one administrator to one or many departments.

5- One department will have many complaints.

6- One department has many workers.

7- One user can evaluate many complaints.

8- The user must see the complaint state.

9- One complaint could have many locations or not.

Identity relevant clauses from the IEEE and ACM Professional Code of Ethics in database analysis :

IEEE code of ethics:

- To avoid unlawful conduct in professional activities.
- To treat all persons fairly and with respect.
- To support colleagues and co-workers in following this code of ethics.
- To maintain and improve our technical competence.
- To seek, accept, and offer honest criticism of technical work.
- To prioritize the safety, health, and welfare of the public.
- To Avoid deceptive acts.

ACM code of ethics:

1. General Ethical Principles

1.1 Contribute to society and to human well-being, acknowledging that all people are stakeholders in computing.

1.2 Avoid harm.

1.3 Be honest and trustworthy.

1.4 Be fair and take action not to discriminate.

1.6 Respect privacy.

1.7 Honor confidentiality.

2. Professional Responsibilities

2.1 Strive to achieve high quality in both the processes and products of professional work.

2.2 Maintain high standards of professional competence, conduct, and ethical practice.

2.3 Know and respect existing rules pertaining to professional work.

2.8 Access computing and communication resources only when authorized or when compelled by the public good.

2.9 Design and implement systems that are robustly and usably secure.

3. Professional Leadership Principles

3.2 Articulate, encourage acceptance of, and evaluate fulfillment of social responsibilities by members of the organization or group.

3.3 Manage personnel and resources to enhance the quality of working life.

3.4 Articulate, apply, and support policies and processes that reflect the principles of the Code.

3.5 Create opportunities for members of the organization or group to grow as professionals.

4. Compliance with The Code

4.1 Uphold, promote, and respect the principles of the Code.

4.2 Treat violations of the Code as inconsistent with membership in the ACM.

Contribution of the team members to phase 2 and strategy used to demonstrate teamwork :

Dana	**Business Rules**
Lama	**EER Diagram**

Phase 3
Designing Database & Normalization

Data Dictionary:

User

Column Name	Key Type	Constraints	FK Table	FK Column	Data Type	Length
User_ID	PK				Varchar	10
Email		Not null, unique			Varchar	62
Password		Not null			Varchar	10
Phone		Not null, unique			Char	10
F_name		Not null			Varchar	32
L_name		Not null			Varchar	21

Student

Column Name	Key Type	Constraints	FK Table	FK Column	Data Type	Length
Student_ID	PK,FK		User	User_ID	Varchar	10
major		Not null			Varchar	20
Academic_Level		Not null			Varchar	20

Employee

Column Name	Key Type	Constraints	FK Table	FK Column	Data Type	Length
Employee_ID	PK,FK		User	User_ID	Varchar	10
Job_Position		Not null			Varchar	10
Rank		Not null			Varchar	20
Department_ ID	FK		Departm ent	Departm ent_ID	Varchar	20

Administrator

Column Name	Key Type	Constraints	FK Table	FK Column	Data Type	Length
Admin_ID	PK				Varchar	10
Email		Not null, unique			Varchar	62
Password		Not null			Varchar	10

Department

Column Name	Key Type	Constraints	FK Table	FK Column	Data Type	Length
Department_ID	PK				Varchar	25
Dpartment_Name		Not null			Varchar	30
Password		Not null			Varchar	10

Column Name	Key Type	Constraints	FK Table	FK Column	Data Type	Length
Email		Not null			Varchar	62
Managed_by	FK		Employee	User_ID	Varchar	10

Complaint

Column Name	Key Type	Constraints	FK Table	FK Column	Data Type	Length
Complaint_ID	PK				Varchar	10
Status					Varchar	30
Complaint_Type		Not null			Varchar	30
Complaint_Text		Not null			Varchar	3000
Complaint_Level		Not null			Varchar	30
Assign_to		Not null	Department	Department_ID	Varchar	25
Solved_Date					Date	15
Requester		Not null	User	User_ID	Varchar	10
Assigned_by		Not null	Administrator	Admin_ID	Varchar	10
Location_ID			Location	Location_ID	Varchar	10
Complaint_Date		Not null			Date	12

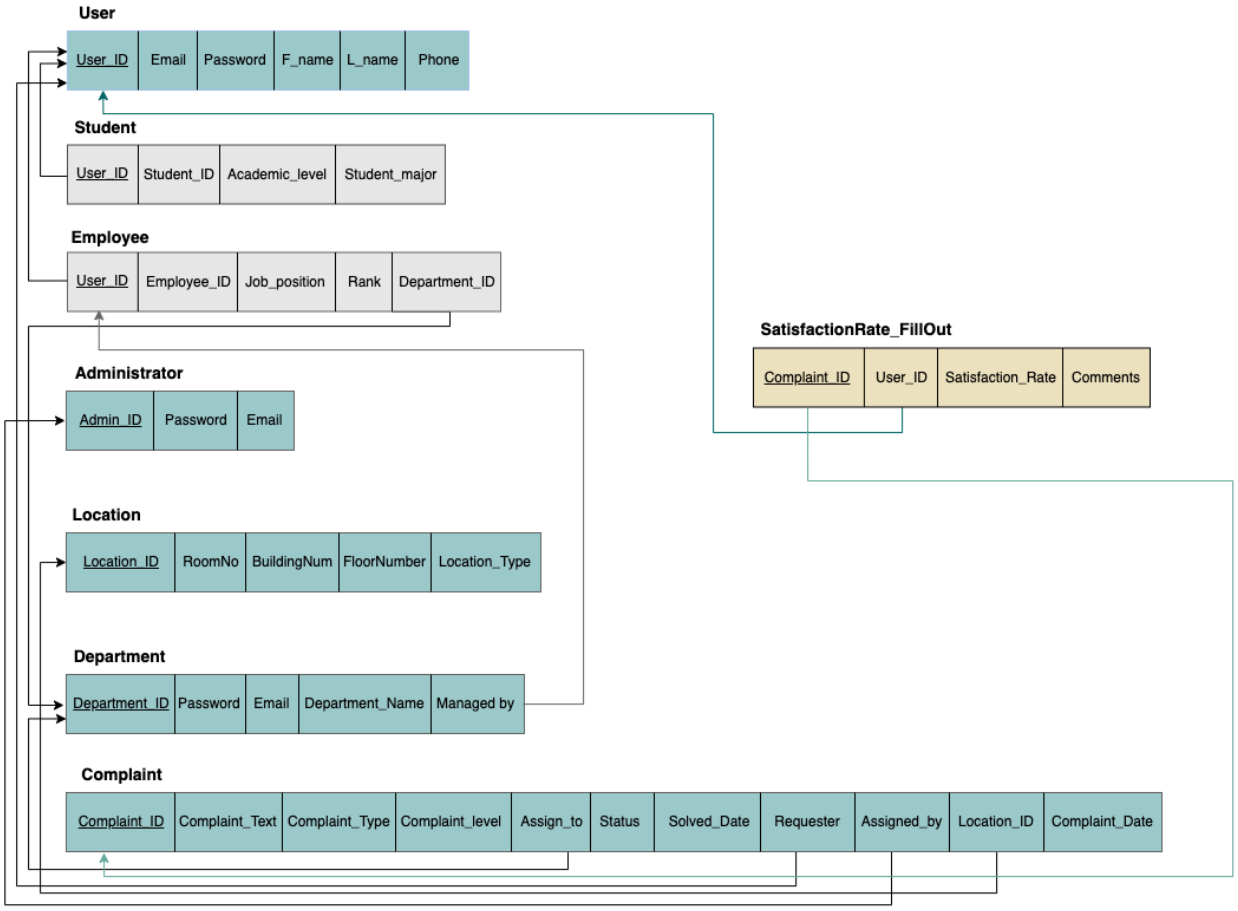
Location

Column Name	Key Type	Constraints	FK Table	FK Column	Data Type	Length
Location_ID	PK				Varchar	10
BuldingNum		Not null			Varchar	30
RoomNo					Varchar	20
FloorNumber		Not null			Number	20
Location_Type		Not null			Varchar	30

SatisfactionRate_FillOut

Column Name	Key Type	Constraints	FK Table	FK Column	Data Type	Length
Complaint_ID	PK,Fk		Complaint	Complaint_ID	Varchar	10
User_ID	FK		User	User_ID	Varchar	10
Satisfaction_Rate					Varchar	100
Comments					Varchar	1000

Relational Model:



Identity relevant clauses from the IEEE and ACM Professional Code of Ethics in database design:

IEEE code of ethics:

- To avoid unlawful conduct in professional activities.
- To treat all persons fairly and with respect.
- To support colleagues and co-workers in following this code of ethics.
- To maintain and improve our technical competence.
- To seek, accept, and offer honest criticism of technical work.
- To prioritize the safety, health, and welfare of the public.
- To Avoid deceptive acts.

ACM code of ethics :

Principle 1: PUBLIC

1.01. Accept full responsibility for their own work.

1.02. Moderate the interests of the software engineer, the employer, the client and the users with the public good.

1.03. Approve software only if they have a well-founded belief that it is safe, meets specifications, passes appropriate tests, and does not diminish quality of life, diminish privacy or harm the environment. The ultimate effect of the work should be to the public good.

1.06. Be fair and avoid deception in all statements, particularly public ones, concerning software or related documents, methods and tools.

1.07. Consider issues of physical disabilities, allocation of resources, economic disadvantage and other factors that can diminish access to the benefits of software.

1.08. Be encouraged to volunteer professional skills to good causes and contribute to public education concerning the discipline.

Principle 2: CLIENT AND EMPLOYER

2.03. Use the property of a client or employer only in ways properly authorized, and with the client's or employer's knowledge and consent.

2.04. Ensure that any document upon which they rely has been approved, when required, by someone authorized to approve it.

2.07. Identify, document, and report significant issues of social concern, of which they are aware, in software or related documents, to the employer or the client.

2.08. Accept no outside work detrimental to the work they perform for their primary employer.

2.09. Promote no interest adverse to their employer or client, unless a higher ethical concern is being compromised; in that case, inform the employer or another appropriate authority of the ethical concern.

Principle 3: PRODUCT

3.01. Strive for high quality, acceptable cost and a reasonable schedule, ensuring significant tradeoffs are clear to and accepted by the employer and the client, and are available for consideration by the user and the public.

3.02. Ensure proper and achievable goals and objectives for any project on which they work or propose.

3.03. Identify, define and address ethical, economic, cultural, legal and environmental issues related to work projects.

3.07. Strive to fully understand the specifications for software on which they work.

3.08. Ensure that specifications for software on which they work have been well documented, satisfy the users' requirements and have the appropriate approvals.

3.13.Be careful to use only accurate data derived by ethical and lawful means, and use it only in ways properly authorized.

3.14. Maintain the integrity of data, being sensitive to outdated or flawed occurrences.

3.15 Treat all forms of software maintenance with the same professionalism as new development.

Principle 4: JUDGMENT

4.01. Temper all technical judgments by the need to support and maintain human values.

4.02 Only endorse documents either prepared under their supervision or within their areas of competence and with which they are in agreement.

4.03. Maintain professional objectivity with respect to any software or related documents they are asked to evaluate.

4.04. Not engage in deceptive financial practices such as bribery, double billing, or other improper financial practices.

4.05. Disclose to all concerned parties those conflicts of interest that cannot reasonably be avoided or escaped.

Principle 5: MANAGEMENT

5.02. Ensure that software engineers are informed of standards before being held to them.

5.03. Ensure that software engineers know the employer's policies and procedures for protecting passwords, files and information that is confidential to the employer or confidential to others.

5.04. Assign work only after taking into account appropriate contributions of education and experience tempered with a desire to further that education and experience.

5.07. Offer fair and just remuneration.

5.08. Not unjustly prevent someone from taking a position for which that person is suitably qualified.

5.11. Not ask a software engineer to do anything inconsistent with this Code.

5.12. Not punish anyone for expressing ethical concerns about a project.

Principle 6: PROFESSION

6.01. Help develop an organizational environment favorable to acting ethically.

6.04. Support, as members of a profession, other software engineers striving to follow this Code

6.06. Obey all laws governing their work, unless, in exceptional circumstances, such compliance is inconsistent with the public interest.
6.07. Be accurate in stating the characteristics of software on which they work, avoiding not only false claims but also claims that might reasonably be supposed to be speculative, vacuous, deceptive, misleading, or doubtful.
6.10. Avoid associations with businesses and organizations which are in conflict with this code.

Principle 7: COLLEAGUES

7.01. Encourage colleagues to adhere to this Code.
7.02. Assist colleagues in professional development.
7.03. Credit fully the work of others and refrain from taking undue credit.
7.08. In situations outside of their own areas of competence, call upon the opinions of other professionals who have competence in that area.

Principle 8: SELF

8.02. Improve their ability to create safe, reliable, and useful quality software at reasonable cost and within a reasonable time.
8.03. Improve their ability to produce accurate, informative, and well-written documentation.
8.04. Improve their understanding of the software and related documents on which they work and of the environment in which they will be used.
8.05. Improve their knowledge of relevant standards and the law governing the software and related documents on which they work.
8.07 Not give unfair treatment to anyone because of any irrelevant prejudices.
8.08. Not influence others to undertake any action that involves a breach of this Code.
8.09. Recognize that personal violations of this Code are inconsistent with being a professional software engineer.

Contribution of the team members to phase 3 and strategy used to demonstrate teamwork :

Dana	Data Dictionary
Lama	Relational Model
Sultanah	ACM Code

Phase 4
Creating Tables

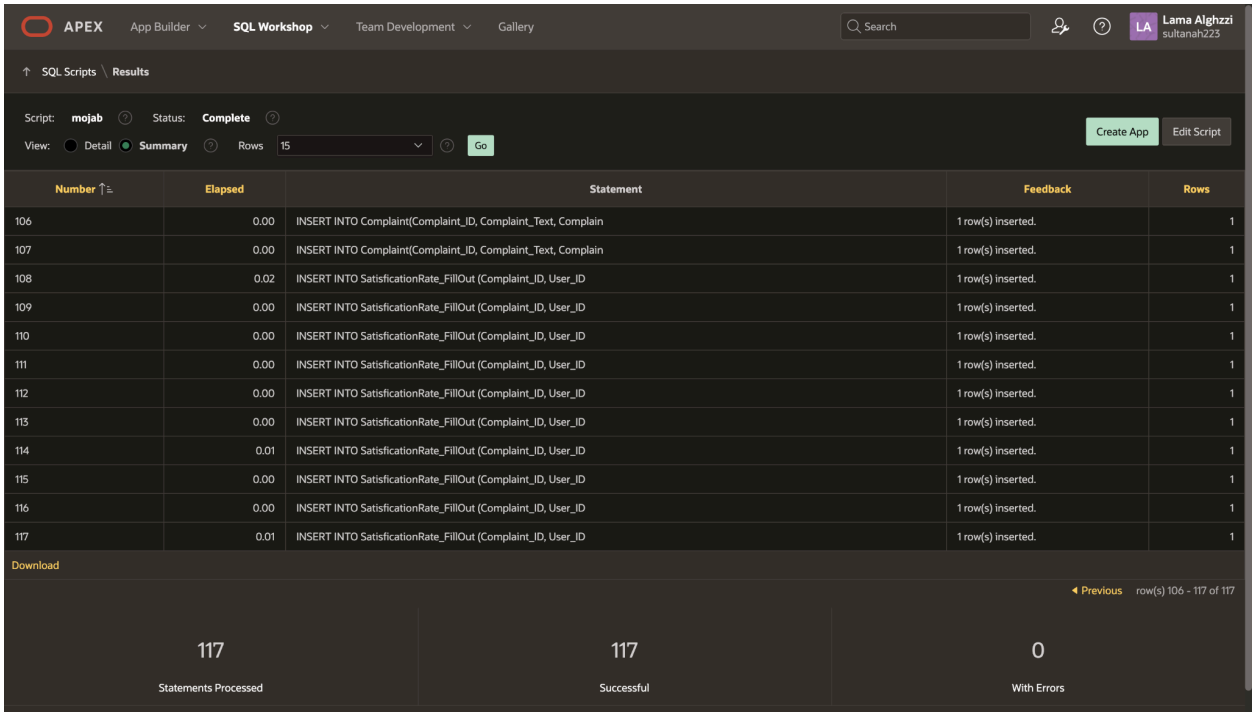
Creating Tables

```
Create table Users(
User_ID Varchar (10) CONSTRAINT user_pk PRIMARY KEY,
Email Varchar (62)  Not Null,
Password Varchar (10) Not Null,
Phone Varchar (10) Not Null,
F_name Varchar (32) Not Null,
L_name Varchar (21) Not Null,

unique (Phone),
unique (Email)
);



Create table Student(
STUDENT_ID Varchar (10) CONSTRAINT student_pk PRIMARY KEY,
Academic_level Varchar (20) Not Null,
Student_major Varchar (20) Not Null,
CONSTRAINT USER_ST_FK FOREIGN KEY (STUDENT_ID) REFERENCES
USERS(USER_ID) ON DELETE CASCADE
);

Create table Employee (
Employee_ID Varchar (10) CONSTRAINT EMP_PK PRIMARY KEY,
```

```
Job_position Varchar (10) Not Null,

Rank Varchar (20) Not Null,

Department_ID Varchar (20),

CONSTRAINT     EMPL_FK   Foreign   key   (EMPLOYEE_ID)   references
Users(USER_ID) ON DELETE CASCADE


);


Create table Department(

Department_ID Varchar (25),

Email Varchar (62)  Not Null,

Password Varchar (10)  Not Null,

Department_Name Varchar (30)  Not Null,

Managed_by Varchar (10),

Primary key (Department_ID),

CONSTRAINT   MAN_BU_FK   FOREIGN   KEY   (MANAGED_BY)   REFERENCES
EMPLOYEE(EMPLOYEE_ID) ON DELETE CASCADE

);



ALTER Table EMPLOYEE

ADD CONSTRAINT fk_Department_ID

FOREIGN KEY (Department_ID) REFERENCES Department(Department_ID)
ON DELETE CASCADE;
```

```
Create table Administrator(
Admin_ID Varchar (10),
Email Varchar (62) Not Null,
Password VARCHAR (10) Not Null,
unique (Email),
Primary key (Admin_ID)
);
Create table Location(
Location_ID Varchar (10),
RoomNo Varchar (20),
BuildingNum Varchar (30) Not Null,
FloorNumber Varchar (20) Not Null,
Location_Type Varchar (30)  Not Null,
Primary key (Location_ID)
);

Create table Complaint(
Complaint_ID Varchar (10),
Complaint_Text Varchar (3000) Not Null,
Complaint_Type Varchar (30) Not Null,
Complaint_Level Varchar (30) Not Null,
Complaint_Date Varchar (12) Not Null,
Assign_to Varchar (25) Not Null,
```

```
Status Varchar (30) ,

Solved_Date Varchar (15),

Requester Varchar (10) Not Null,

Assign_by Varchar (10) Not Null,

Location_ID Varchar (10) ,

Primary key (Complaint_ID),

Foreign key (Assign_to) references Department (Department_ID) ON
DELETE CASCADE,

Foreign key (Requester) references Users (User_ID)  ON DELETE
CASCADE,

Foreign key (Assign_by) references Administrator (Admin_ID) ON
DELETE CASCADE,

Foreign key (Location_ID) references Location (Location_ID)  ON
DELETE CASCADE

);


Create table SatisficationRate_FillOut(

Complaint_ID Varchar (10),

User_ID Varchar (10),

Satisfication_Rate Varchar (100),

Comments Varchar (1000),

Primary key (Complaint_ID),

Foreign key (Complaint_ID) references Complaint (Complaint_ID)
ON DELETE CASCADE,

Foreign key (User_ID) references Users (User_ID)  ON DELETE
CASCADE

);
```



https://apex.oracle.com/pls/apex/r/apex/sql-workshop/script-editor?p60_file_id=31549683598038273543&session=14894670595107

Table: **USERS**

Column	Data Type	Length	Precision	Scale	Nullable
USER_ID	VARCHAR2	10	-	-	No
EMAIL	VARCHAR2	62	-	-	No
PASSWORD	VARCHAR2	10	-	-	No
PHONE	VARCHAR2	10	-	-	No
F_NAME	VARCHAR2	32	-	-	No
L_NAME	VARCHAR2	21	-	-	No

1 - 6

Table: **STUDENT**

Column	Data Type	Length	Precision	Scale	Nullable
STUDENT_ID	VARCHAR2	10	-	-	No
ACADEMIC_LEVEL	VARCHAR2	20	-	-	No
STUDENT_MAJOR	VARCHAR2	20	-	-	No

1 - 3

Table: **EMPLOYEE**

Column	Data Type	Length	Precision	Scale	Nullable
EMPLOYEE_ID	VARCHAR2	10	-	-	No
JOB_POSITION	VARCHAR2	10	-	-	No
RANK	VARCHAR2	20	-	-	No
DEPARTMENT_ID	VARCHAR2	20	-	-	Yes

1 - 4

Table: **ADMINISTRATOR**

Column	Data Type	Length	Precision	Scale	Nullable
ADMIN_ID	VARCHAR2	10	-	-	No
EMAIL	VARCHAR2	62	-	-	No
PASSWORD	VARCHAR2	10	-	-	No

1 - 3

Table: **LOCATION**

Column	Data Type	Length	Precision	Scale	Nullable
LOCATION_ID	VARCHAR2	10	-	-	No
ROOMNO	VARCHAR2	20	-	-	Yes
BUILDINGNUM	VARCHAR2	30	-	-	No
FLOORNUMBER	VARCHAR2	20	-	-	No
LOCATION_TYPE	VARCHAR2	30	-	-	No

1 - 5

Table: **DEPARTMENT**

Column	Data Type	Length	Precision	Scale	Nullable
DEPARTMENT_ID	VARCHAR2	25	-	-	No
EMAIL	VARCHAR2	62	-	-	No
PASSWORD	VARCHAR2	10	-	-	No
DEPARTMENT_NAME	VARCHAR2	30	-	-	No
MANAGED_BY	VARCHAR2	10	-	-	Yes

1 - 5

Table: **COMPLAINT**

Column	Data Type	Length	Precision	Scale	Nullable
COMPLAINT_ID	VARCHAR2	10	-	-	No
COMPLAINT_TEXT	VARCHAR2	3000	-	-	No
COMPLAINT_TYPE	VARCHAR2	30	-	-	No
COMPLAINT_LEVEL	VARCHAR2	30	-	-	No
COMPLAINT_DATE	VARCHAR2	12	-	-	No
ASSIGN_TO	VARCHAR2	25	-	-	No
STATUS	VARCHAR2	30	-	-	Yes
SOLVED_DATE	VARCHAR2	15	-	-	Yes
REQUESTER	VARCHAR2	10	-	-	No
ASSIGN_BY	VARCHAR2	10	-	-	No
LOCATION_ID	VARCHAR2	10	-	-	Yes

1 - 11

Table: **SATISFICATIONRATE_FILLOUT**

Column	Data Type	Length	Precision	Scale	Nullable
COMPLAINT_ID	VARCHAR2	10	-	-	No
USER_ID	VARCHAR2	10	-	-	Yes
SATISFICATION_RATE	VARCHAR2	100	-	-	Yes
COMMENTS	VARCHAR2	1000	-	-	Yes

1 - 4

Inserting 10 Records For Each Entity

User

```
INSERT INTO Users values

('2204103410', '220410341@psu.edu.sa', 'Aa56788', '0554679233',
'Dana', 'Amri');

INSERT INTO Users values

('2205103410', '220510341@psu.edu.sa', 'Ea523448', '0505379223',
'Sara', 'Mohd');

INSERT INTO Users values

('2208103210', '220810321@psu.edu.sa', 'Rt913648', '0505381233',
'Asma', 'Rajhi');

INSERT INTO Users values

('2207104210',        '2207104210@psu.edu.sa',        'Gq911678',
'0505811733', 'Nora', 'Ali');

INSERT INTO Users values

('2201104230',        '2201104230@psu.edu.sa',        'Ww512688',
'0505811933', 'Lina', 'Jamal');

INSERT INTO Users values

('2201303230',        '2201303230@psu.edu.sa',        'Ea112618',
'0505214934', 'Noor', 'Abduallah');

INSERT INTO Users values

('2204382230',        '2204382230@psu.edu.sa',        'Ss432638',
'0505514762', 'Jomana', 'Marwan');

INSERT INTO Users values

('2204552630',        '2204552630@psu.edu.sa',        'Cv442239',
'0505224964', 'Suad', 'Khaled');

INSERT INTO Users values
```

```
('2204452930',        '2204452930@psu.edu.sa',        'Cy222299',
'0505923904', 'Aljohara', 'Saad');

INSERT INTO Users values

('2214492630',        '2214492630@psu.edu.sa',        'Ca281179',
'0505523104', 'Raghad', 'Ibrahim');

INSERT INTO Users values

('1115566780',        '1115566780@psu.edu.sa',        'Ce291979',
'0506523114', 'Sara', 'Jihad');

INSERT INTO Users values

('1122566980',        '1122566980@psu.edu.sa',        'Ar291449',
'0506543614', 'Lamia', 'Faisal');

INSERT INTO Users values

('1132593980',        '1132593980@psu.edu.sa',        'Wq196599',
'0506141614', 'Sultanah', 'Ahmad');

INSERT INTO Users values

('1157597780',        '1157597780@psu.edu.sa',        'Wi292589',
'0506271614', 'Haya', 'Osama');

INSERT INTO Users values

('1159599980',        '1159599980@psu.edu.sa',        'Ip231149',
'0516271614', 'Hana', 'Emad');

INSERT INTO Users values

('1109393180',        '1109393180@psu.edu.sa',        'It631949',
'0519271624', 'Shahad', 'Yusuf');

INSERT INTO Users values

('1109222100',        '1109222100@psu.edu.sa',        'Yi101245',
'0519288914', 'Sawsan', 'Kamal');

INSERT INTO Users values
```

```
('1118242100',        '1118242100@psu.edu.sa',        'Oa121549',
'0509118994', 'Haifa', 'Yazan');

INSERT INTO Users values

('1158849120',        '1158849120@psu.edu.sa',        'Ef922589',
'0509918991', 'Jood', 'Saud');

INSERT INTO Users values

('1143879420',        '1143879420@psu.edu.sa',        'Pk821089',
'0508926911', 'Esra', 'Saleh');
```

Student

```
INSERT INTO Student values

('2204103410', 'Junior', 'Computer Science');

INSERT INTO Student values

('2205103410', 'Sophomore', 'Computer Science');

INSERT INTO Student values

('2208103210', 'Junior', 'Software engineering');

INSERT INTO Student values

('2207104210', 'Sophomore', 'Software engineering');

INSERT INTO Student values

('2204382230', 'Freshman', 'Information systems');

INSERT INTO Student values

('2204552630', 'Junior', 'Finance');

INSERT INTO Student values

('2204452930', 'Senior', 'Finance');

INSERT INTO Student values

('2214492630', 'Sophomore', 'Marketing');
```

```
INSERT INTO Student values
('2201303230', 'Junior', 'Internal Design');
INSERT INTO Student values
('2201104230', 'Sophomore', 'Internal Design');
```

Employee

```
INSERT INTO Employee (Employee_ID, Job_position, Rank)
VALUES ('1115566780', 'Manager', '1');
INSERT INTO Employee (Employee_ID, Job_position, Rank)
VALUES ('1122566980', 'Manager', '3');
INSERT INTO Employee (Employee_ID, Job_position, Rank)
VALUES ('1132593980', 'Assistant', '2');
INSERT INTO Employee (Employee_ID, Job_position, Rank)
VALUES ('1157597780', 'Assistant', '1');
INSERT INTO Employee (Employee_ID, Job_position, Rank)
VALUES ('1159599980', 'Assistant', '2');
INSERT INTO Employee (Employee_ID, Job_position, Rank)
VALUES ('1109393180', 'Manager', '3');
INSERT INTO Employee (Employee_ID, Job_position, Rank)
VALUES ('1109222100', 'Manager', '2');
INSERT INTO Employee (Employee_ID, Job_position, Rank)
VALUES ('1118242100', 'Manager', '1');
INSERT INTO Employee (Employee_ID, Job_position, Rank)
VALUES ('1158849120', 'Assistant', '1');
```

```
INSERT INTO Employee (Employee_ID, Job_position, Rank)

VALUES ('1143879420', 'Assistant', '3');
```

Department

```
INSERT INTO Department (Department_ID, Email, Password, Department_Name, Managed_by)

VALUES ('DEP002', 'cad@example.com', 'cad1234', 'CAD Department', '1115566780');

INSERT INTO Department (Department_ID, Email, Password, Department_Name, Managed_by)

VALUES ('DEP001', 'ccis@example.com', 'ccis1234', 'CCIS Department', '1122566980');

INSERT INTO Department (Department_ID, Email, Password, Department_Name, Managed_by)

VALUES ('DEP004', 'chs@psu.edu.sa', 'chs1234', 'CHS Department', '1109393180');

INSERT INTO Department (Department_ID, Email, Password, Department_Name, Managed_by)

VALUES ('DEP006', 'medical@psu.edu.sa', 'med1234', 'Medical Department', '1109222100');

INSERT INTO Department (Department_ID, Email, Password, Department_Name, Managed_by)

VALUES ('DEP007', 'IT@psu.edu.sa', 'IT1234', 'IT Department', '1118242100');

INSERT INTO Department (Department_ID, Email, Password, Department_Name, Managed_by)

VALUES ('DEP008', 'CE@psu.edu.sa', 'CE1234', 'CE Department', '1132593980');
```

```
INSERT INTO Department (Department_ID, Email, Password,
Department_Name, Managed_by)

VALUES ('DEP009', 'quality@psu.edu.sa', 'Qa1234', 'Quality
assurance Department', '1157597780');

INSERT INTO Department (Department_ID, Email, Password,
Department_Name, Managed_by)

VALUES ('DEP010', 'maintenance@psu.edu.sa', 'Qa1234',
'maintenance Department', '1159599980');

INSERT INTO Department (Department_ID, Email, Password,
Department_Name, Managed_by)

VALUES ('DEP005', 'law@psu.edu.sa', 'law1234', 'Law Department',
'1158849120');

INSERT INTO Department (Department_ID, Email, Password,
Department_Name, Managed_by)

VALUES ('DEP003', 'cpa@psu.edu.sa', 'cpa1234', 'CPA Department',
'1143879420');


UPDATE Employee

SET   DEPARTMENT_ID = 'DEP001'

WHERE employee_id = 1122566980 ;


UPDATE Employee

SET   DEPARTMENT_ID = 'DEP002'

WHERE employee_id = 1115566780 ;


UPDATE Employee

SET   DEPARTMENT_ID = 'DEP003'
```

```
WHERE employee_id = 1143879420;

UPDATE Employee
SET   DEPARTMENT_ID = 'DEP004'
WHERE employee_id = 1109393180 ;

UPDATE Employee
SET   DEPARTMENT_ID = 'DEP005'
WHERE employee_id = 1158849120 ;

UPDATE Employee
SET   DEPARTMENT_ID = 'DEP006'
WHERE employee_id = 1109222100 ;

UPDATE Employee
SET   DEPARTMENT_ID = 'DEP007'
WHERE employee_id = 1118242100 ;

UPDATE Employee
SET   DEPARTMENT_ID = 'DEP008'
WHERE employee_id = 1132593980 ;

UPDATE Employee
SET   DEPARTMENT_ID = 'DEP009'
```

```
WHERE employee_id = 1157597780 ;


UPDATE Employee

SET   DEPARTMENT_ID = 'DEP010'

WHERE employee_id = 1159599980 ;
```

Administrator:

```
INSERT INTO Administrator (Admin_ID, Email, Password)
VALUES ('ADM001', 'admin@psu.edu.sa', 'adminPa1');
INSERT INTO Administrator (Admin_ID, Email, Password)
VALUES ('ADM002', 'admin2@psu.edu.sa', 'adminPa2');
INSERT INTO Administrator (Admin_ID, Email, Password)
VALUES ('ADM003', 'admin3@psu.edu.sa', 'adminPa3');
INSERT INTO Administrator (Admin_ID, Email, Password)
VALUES ('ADM004', 'admin4@psu.edu.sa', 'adminPa4');
INSERT INTO Administrator (Admin_ID, Email, Password)
VALUES ('ADM005', 'admin5@psu.edu.sa', 'adminPa5');
INSERT INTO Administrator (Admin_ID, Email, Password)
VALUES ('ADM006', 'admin6@psu.edu.sa', 'adminPa6');
INSERT INTO Administrator (Admin_ID, Email, Password)
VALUES ('ADM007', 'admin7@psu.edu.sa', 'adminPa7');
INSERT INTO Administrator (Admin_ID, Email, Password)
VALUES ('ADM008', 'admin8@psu.edu.sa', 'adminPa8');
INSERT INTO Administrator (Admin_ID, Email, Password)
VALUES ('ADM009', 'admin9@psu.edu.sa', 'adminPa9');
INSERT INTO Administrator (Admin_ID, Email, Password)
VALUES ('ADM010', 'admin10@psu.edu.sa', 'adminPa10');
```

Location:

```
INSERT INTO Location (Location_ID, RoomNo, BuildingNum,
FloorNumber, Location_Type)
VALUES ('LOC001', '105', 'W243', '2', 'classroom');
```

```
INSERT INTO Location (Location_ID, RoomNo, BuildingNum,
FloorNumber, Location_Type)
VALUES ('LOC002', '104', 'Labg', '1', 'computer labs');
INSERT INTO Location (Location_ID, RoomNo, BuildingNum,
FloorNumber, Location_Type)
VALUES ('LOC003', '102', 'N104', '1', 'meetingRoom');
INSERT INTO Location (Location_ID, RoomNo, BuildingNum,
FloorNumber, Location_Type)
VALUES ('LOC004', '105', 'S102', '3', 'library');
INSERT INTO Location (Location_ID, RoomNo, BuildingNum,
FloorNumber, Location_Type)
VALUES ('LOC005', '105', 'A236', '2', 'student lounge');
INSERT INTO Location (Location_ID, RoomNo, BuildingNum,
FloorNumber, Location_Type)
VALUES ('LOC006', '105', 'W350', '3', 'offices');
INSERT INTO Location (Location_ID, RoomNo, BuildingNum,
FloorNumber, Location_Type)
VALUES ('LOC007', '104', 'W123', '0', 'architecture room');
INSERT INTO Location (Location_ID, RoomNo, BuildingNum,
FloorNumber, Location_Type)
VALUES ('LOC008', '101', 'S204', '2', 'pool');
INSERT INTO Location (Location_ID, RoomNo, BuildingNum,
FloorNumber, Location_Type)
VALUES ('LOC009', '102', 'S206', '2', 'gym');
INSERT INTO Location (Location_ID, RoomNo, BuildingNum,
FloorNumber, Location_Type)
VALUES ('LOC010', '101', 'S109', '0', 'court');
```

Complaint:

```
INSERT INTO Complaint(Complaint_ID, Complaint_Text,
Complaint_Type, Complaint_Level, Assign_to, Status, Solved_Date,
Requester, Assign_by, Location_ID, Complaint_Date)
VALUES ('CMP001', 'Internet connectivity issue', 'Technical',
'High', 'DEP002', 'Pending', NULL, '2204103410', 'ADM001',
'LOC001', '2023-11-20');
```

```
INSERT INTO Complaint(Complaint_ID, Complaint_Text,
Complaint_Type, Complaint_Level, Assign_to, Status, Solved_Date,
Requester, Assign_by, Location_ID, Complaint_Date)
VALUES ('CMP002', 'Printer not working', 'Hardware', 'Medium',
'DEP001', 'inProgress', NULL, '2205103410', 'ADM002', 'LOC002',
'2023-11-22');

INSERT INTO Complaint(Complaint_ID, Complaint_Text,
Complaint_Type, Complaint_Level, Assign_to, Status, Solved_Date,
Requester, Assign_by, Location_ID, Complaint_Date)
VALUES ('CMP003', 'Application crashing', 'Software', 'Law',
'DEP003', 'Closed', '2023-11-16', '2208103210', 'ADM003',
'LOC003', '2023-11-23');

INSERT INTO Complaint(Complaint_ID, Complaint_Text,
Complaint_Type, Complaint_Level, Assign_to, Status, Solved_Date,
Requester, Assign_by, Location_ID, Complaint_Date)
VALUES ('CMP004', 'Slow internet speed', 'Network', 'High',
'DEP004', 'Pending', NULL, '2207104210', 'ADM004', 'LOC004',
'2023-11-23');

INSERT INTO Complaint(Complaint_ID, Complaint_Text,
Complaint_Type, Complaint_Level, Assign_to, Status, Solved_Date,
Requester, Assign_by, Location_ID, Complaint_Date)
VALUES ('CMP005', 'Computer not booting', 'Hardware', 'Medium',
'DEP005', 'inProgress', NULL, '2201104230', 'ADM005', 'LOC005',
'2023-11-24');

INSERT INTO Complaint(Complaint_ID, Complaint_Text,
Complaint_Type, Complaint_Level, Assign_to, Status, Solved_Date,
Requester, Assign_by, Location_ID, Complaint_Date)
VALUES ('CMP006', 'Software not responding', 'Software', 'Low',
'DEP006', 'Closed', '2023-11-18', '2201303230', 'ADM006',
'LOC006', '2023-11-25');

INSERT INTO Complaint(Complaint_ID, Complaint_Text,
Complaint_Type, Complaint_Level, Assign_to, Status, Solved_Date,
Requester, Assign_by, Location_ID, Complaint_Date)
```

```
VALUES ('CMP007', 'Server downtime', 'Technical', 'High',
'DEP007', 'Pending', NULL, '2204382230', 'ADM007', 'LOC007',
'2023-11-26');

INSERT INTO Complaint(Complaint_ID, Complaint_Text,
Complaint_Type, Complaint_Level, Assign_to, Status, Solved_Date,
Requester, Assign_by, Location_ID, Complaint_Date)
VALUES ('CMP008', 'Keyboard malfunction', 'Hardware', 'Medium',
'DEP008', 'inProgress', NULL, '2204552630', 'ADM008', 'LOC008',
'2023-11-27');

INSERT INTO Complaint(Complaint_ID, Complaint_Text,
Complaint_Type, Complaint_Level, Assign_to, Status, Solved_Date,
Requester, Assign_by, Location_ID, Complaint_Date)
VALUES ('CMP009', 'Database error', 'Software', 'Low', 'DEP009',
'Closed', '2023-11-19', '2204452930', 'ADM009', 'LOC009',
'2023-11-10');

INSERT INTO Complaint(Complaint_ID, Complaint_Text,
Complaint_Type, Complaint_Level, Assign_to, Status, Solved_Date,
Requester, Assign_by, Location_ID, Complaint_Date)
VALUES ('CMP010', 'Connection drops', 'Network', 'High',
'DEP010', 'Pending', NULL, '2214492630', 'ADM010', 'LOC010',
'2023-11-29');
```

SatisficationRate_FillOut table:

```
INSERT INTO SatisficationRate_FillOut (Complaint_ID, User_ID,
Satisfication_Rate, Comments)

VALUES ('CMP001', '2204103410', '4/5', 'The issue was resolved
efficiently');

INSERT INTO SatisficationRate_FillOut (Complaint_ID, User_ID,
Satisfication_Rate, Comments)

VALUES ('CMP002', '2205103410', '3/5', 'Satisfactory resolution
but took longer than expected');
```

```
INSERT INTO SatisficationRate_FillOut (Complaint_ID, User_ID,
Satisfication_Rate, Comments)

VALUES ('CMP003', '2208103210', '5/5', 'Excellent service! Quick
resolution');

INSERT INTO SatisficationRate_FillOut (Complaint_ID, User_ID,
Satisfication_Rate, Comments)

VALUES ('CMP004', '2207104210', '2/5', 'Not satisfied with the
resolution');

INSERT INTO SatisficationRate_FillOut (Complaint_ID, User_ID,
Satisfication_Rate, Comments)

VALUES ('CMP005', '2201104230', '4/5', 'Resolved the issue, but
communication could be improved');

INSERT INTO SatisficationRate_FillOut (Complaint_ID, User_ID,
Satisfication_Rate, Comments)

VALUES ('CMP006', '2201303230', '5/5', 'Outstanding support');

INSERT INTO SatisficationRate_FillOut (Complaint_ID, User_ID,
Satisfication_Rate, Comments)

VALUES ('CMP007', '2204382230', '3/5', 'Resolution took longer
than anticipated');

INSERT INTO SatisficationRate_FillOut (Complaint_ID, User_ID,
Satisfication_Rate, Comments)

VALUES ('CMP008', '2204552630', '4/5', 'Good service, but room
for improvement');

INSERT INTO SatisficationRate_FillOut (Complaint_ID, User_ID,
Satisfication_Rate, Comments)

VALUES ('CMP009', '2204452930', '5/5', 'Very satisfied with the
quick resolution');

INSERT INTO SatisficationRate_FillOut (Complaint_ID, User_ID,
Satisfication_Rate, Comments)
```

```
VALUES ('CMP010', '2214492630', '2/5', 'Not happy with the support provided');
```

COMPLAINT_ID	COMPLAINT_TEXT	COMPLAINT_TYPE	COMPLAINT_LEVEL	ASSIGN_TO	STATUS	SOLVED_DATE	REQUESTER	ASSIGN_BY	LOCATION_ID	COMPLAIN
CMP010	Connection drops	Network	High	DEP010	Pending	NULL	2214492630	ADM010	LOC010	2023-11-29
CMP004	Slow internet speed	Network	High	DEP004	Pending	NULL	2207104210	ADM004	LOC004	2023-11-23
CMP007	Server downtime	Technical	High	DEP007	Pending	NULL	2204382230	ADM007	LOC007	2023-11-26
CMP001	Internet connectivity issue	Technical	High	DEP002	Pending	NULL	2204103410	ADM001	LOC001	2023-11-20
CMP009	Database erorr	Software	Low	DEP009	Closed	2023-11-19	2204452930	ADM009	LOC009	2023-11-10
CMP003	Application crashing	Software	Law	DEP003	Closed	2023-11-16	2208103210	ADM003	LOC003	2023-11-23
CMP002	Printer not working	Hardware	Medium	DEP001	inProgress	NULL	2205103410	ADM002	LOC002	2023-11-22
CMP005	Computer not booting	Hardware	Medium	DEP005	inProgress	NULL	2201104230	ADM005	LOC005	2023-11-24
CMP006	Software not responding	Software	Low	DEP006	Closed	2023-11-18	2201303230	ADM006	LOC006	2023-11-25
CMP008	Keyboard malfunction	Hardware	Medium	DEP008	inProgress	NULL	2204552630	ADM008	LOC008	2023-11-27

```
1  select *
2  from SatisficationRate_FillOut
```

Results | Explain | Describe | Saved SQL | History

COMPLAINT_ID	USER_ID	SATISFICATION_RATE	COMMENTS
CMP001	2204103410	4/5	The issue was resolved efficiently
CMP003	2208103210	5/5	Excellent service! Quick resolution
CMP010	2214492630	2/5	Not happy with the support provided
CMP005	2201104230	4/5	Resolved the issue, but communication could be improved
CMP002	2205103410	3/5	Satisfactory resolution but took longer than expected
CMP008	2204552630	4/5	Good service, but room for improvement
CMP004	2207104210	2/5	Not satisfied with the resolution
CMP006	2201303230	5/5	Outstanding support
CMP007	2204382230	3/5	Resolution took longer than anticipated
CMP009	2204452930	5/5	Very satisfied with the quick resolution

10 rows returned in 0.02 seconds Download

```
select *
from Location
```

Results | Explain | Describe | Saved SQL | History

LOCATION_ID	ROOMNO	BUILDINGNUM	FLOORNUMBER	LOCATION_TYPE
LOC003	102	N104	1	meetingRoom
LOC005	105	A236	2	student lounge
LOC009	102	S206	2	gym
LOC002	104	Labg	1	computer labs
LOC006	105	W350	3	offices
LOC007	104	W123	0	architecture room
LOC008	101	S204	2	pool
LOC010	101	S109	0	court
LOC001	105	W243	2	classroom
LOC004	105	S102	3	labrary

10 rows returned in 0.03 seconds Download

```
select *
from Administrator
```

Results | Explain | Describe | Saved SQL | History

ADMIN_ID	EMAIL	PASSWORD
ADM004	admin4@psu.edu.sa	adminPa4
ADM008	admin8@psu.edu.sa	adminPa8
ADM002	admin2@psu.edu.sa	adminPa2
ADM010	admin10@psu.edu.sa	adminP10
ADM006	admin6@psu.edu.sa	adminPa
ADM007	admin7@psu.edu.sa	adminPa7
ADM009	admin9@psu.edu.sa	adminPa9
ADM001	admin@psu.edu.sa	adminPa1
ADM003	admin3@psu.edu.sa	adminPa3
ADM005	admin5@psu.edu.sa	adminPa5

10 rows returned in 0.01 seconds Download

```
1  select *
2  from Department
```

Results | Explain | Describe | Saved SQL | History

DEPARTMENT_ID	EMAIL	PASSWORD	DEPARTMENT_NAME	MANAGED_BY
DEP002	cad@example.com	cad1234	CAD Department	1115566780
DEP001	ccis@example.com	ccis1234	CCIS Department	1122566980
DEP004	chs@psu.edu.sa	chs1234	CHS Department	1109393180
DEP007	IT@psu.edu.sa	IT1234	IT Department	1118242100
DEP008	CE@psu.edu.sa	CE1234	CE Department	1132593980
DEP010	maintenance@psu.edu.sa	Qa1234	maintenance Department	1159599980
DEP006	medical@psu.edu.sa	med1234	Medical Department	1109222100
DEP009	quality@psu.edu.sa	Qa1234	Quality assurance Department	1157597780
DEP005	law@psu.edu.sa	law1234	Law Department	1158849120
DEP003	cpa@psu.edu.sa	cpa1234	CPA Department	1143879420

10 rows returned in 0.03 seconds Download

```
1  select *
2  from student
```

Results | Explain | Describe | Saved SQL | History

STUDENT_ID	ACADEMIC_LEVEL	STUDENT_MAJOR
2208103210	Junior	Software engineering
2201303230	Junior	Internal Design
2204103410	Junior	Computer Science
2205103410	Sophomore	Computer Science
2207104210	Sophomore	Software engineering
2204382230	Freshman	Information systems
2204452930	Senior	Finance
2214492630	Sophomore	Marketing
2204552630	Junior	Finance
2201104230	Sophomore	Internal Design

10 rows returned in 0.02 seconds Download

```
1  select *
2  from users
```

Results | Explain | Describe | Saved SQL | History

USER_ID	EMAIL	PASSWORD	PHONE	F_NAME	L_NAME
2208103210	2208103210@psu.edu.sa	Rt913648	0505381233	Asma	Rajhi
2204382230	2204382230@psu.edu.sa	Ss432638	0505514762	Jomana	Marwan
2204552630	2204552630@psu.edu.sa	Cv442239	0505224964	Suad	Khaled
1158849120	1158849120@psu.edu.sa	Ef922589	0509918991	Jood	Saud
2204103410	2204103410@psu.edu.sa	Aa56788	0554679233	Dana	Amri
2201303230	2201303230@psu.edu.sa	Ea112618	0505214934	Noor	Abduallah
1122566980	1122566980@psu.edu.sa	Ar291449	0506543614	Lamia	Faisal
1115566780	1115566780@psu.edu.sa	Ce291979	0506523114	Sara	Jihad
1157597780	1157597780@psu.edu.sa	Wi292589	0506271614	Haya	Osama
1159599980	1159599980@psu.edu.sa	Ip231149	0516271614	Hana	Emad

More than 10 rows available. Increase rows selector to view more rows.

```
1  select *
2  from Employee
```

Results | Explain | Describe | Saved SQL | History

EMPLOYEE_ID	JOB_POSITION	RANK	DEPARTMENT_ID
1132593980	assistant	2	DEP008
1157597780	assistant	1	DEP009
1109393180	manager	3	DEP004
1109222100	manager	2	DEP006
1115566780	Manager	1	DEP002
1122566980	Manager	3	DEP001
1158849120	assistant	1	DEP005
1159599980	assistant	2	DEP010
1118242100	manager	1	DEP007
1143879420	assistant	3	DEP003

10 rows returned in 0.02 seconds Download

Contribution of the team members to phase 4 and strategy used to demonstrate teamwork :

Dana	**INSERTING RECORDS**
Lama	**Creating Tables**
Sultanah	**INSERTING RECORDS**

Phase 5

Creating simple and advanced queries using Oracle

1- Simple queries

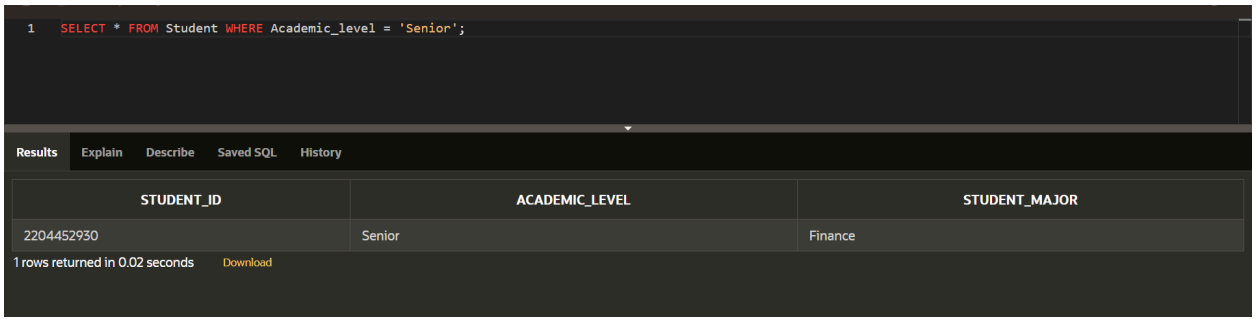
1

-Display the employee_id, job_positon, rank of employee who work as assistants

```
select Employee_ID, Job_position, Rank
from employee
where Job_position = 'assistant'
```

-List of all the students who is senior

```
SELECT *
FROM Student WHERE Academic_level = 'Senior';
```

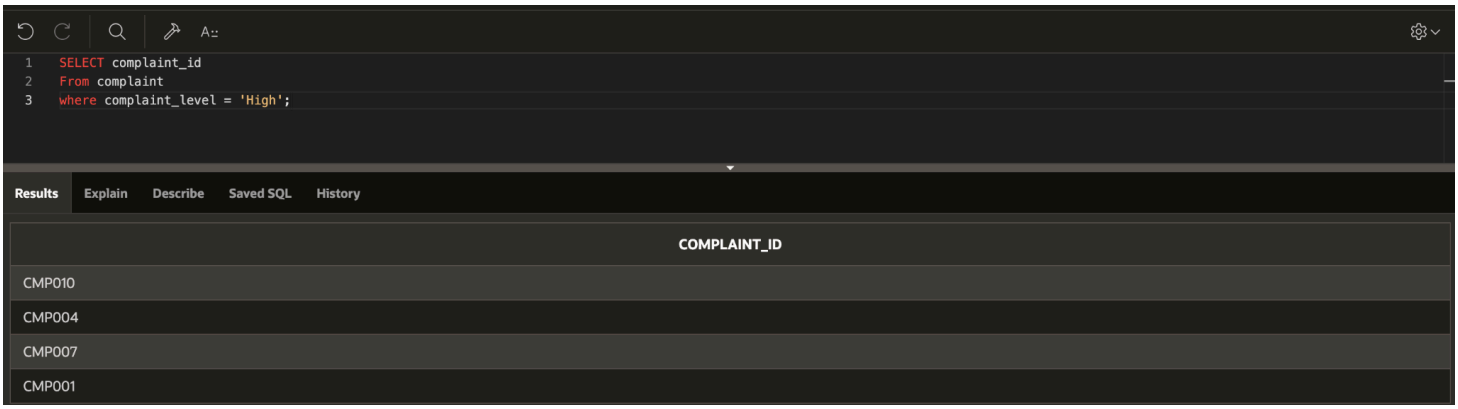


-Display the record for a specific user

```
SELECT*
FROM SatisficationRate_FillOut
WHERE user_ID = '22144926305';
```

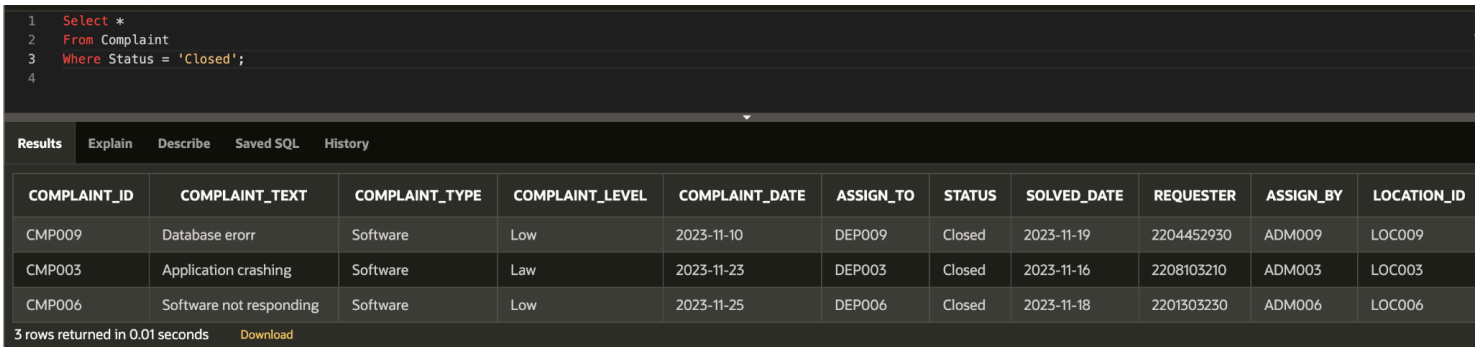
-Display the complaint ID for complaints with a complaint level High

```
SELECT complaint_id
From complaint
where complaint_level = 'High';
```



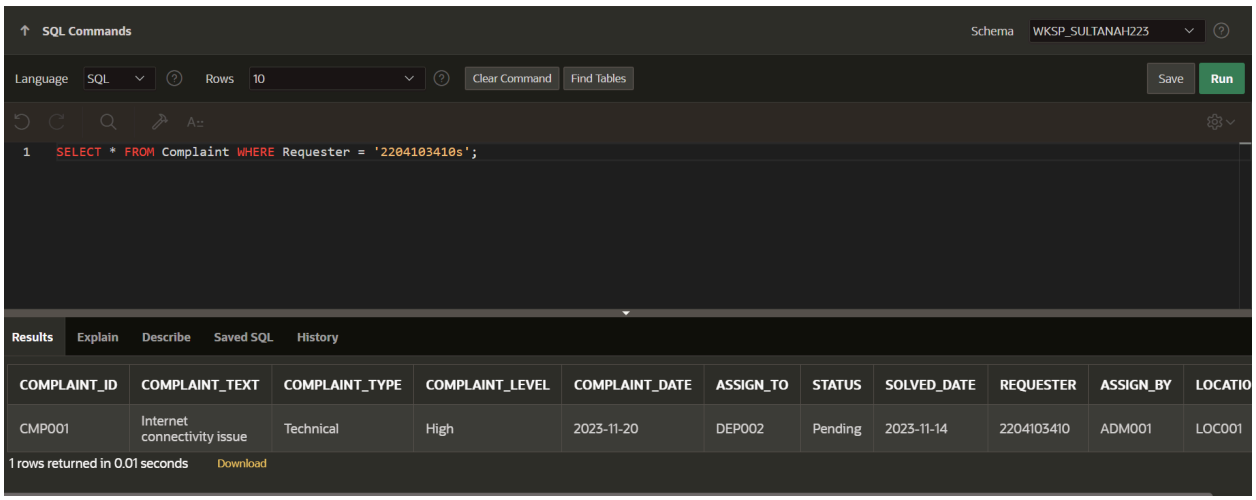
-Display all columns from the Complaint that its status is Closed

```
SELECT *
FROM Complaint
WHERE Status = 'Closed';
```



-Count the number of complaints

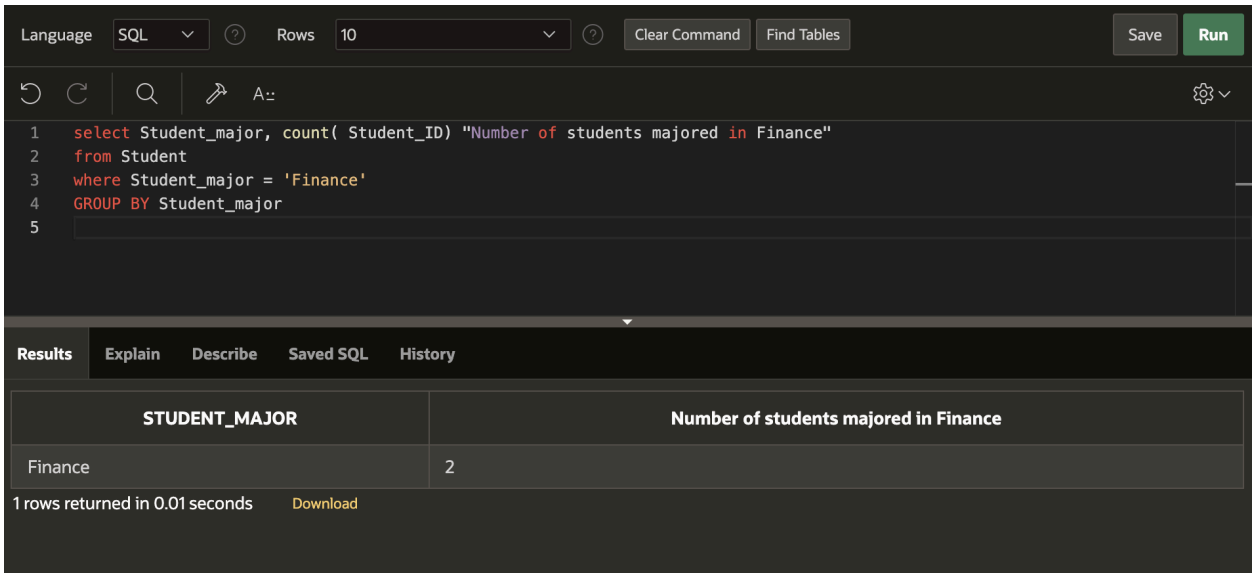
```
SELECT COUNT(*) AS complaint_count FROM Complaint;
```



2

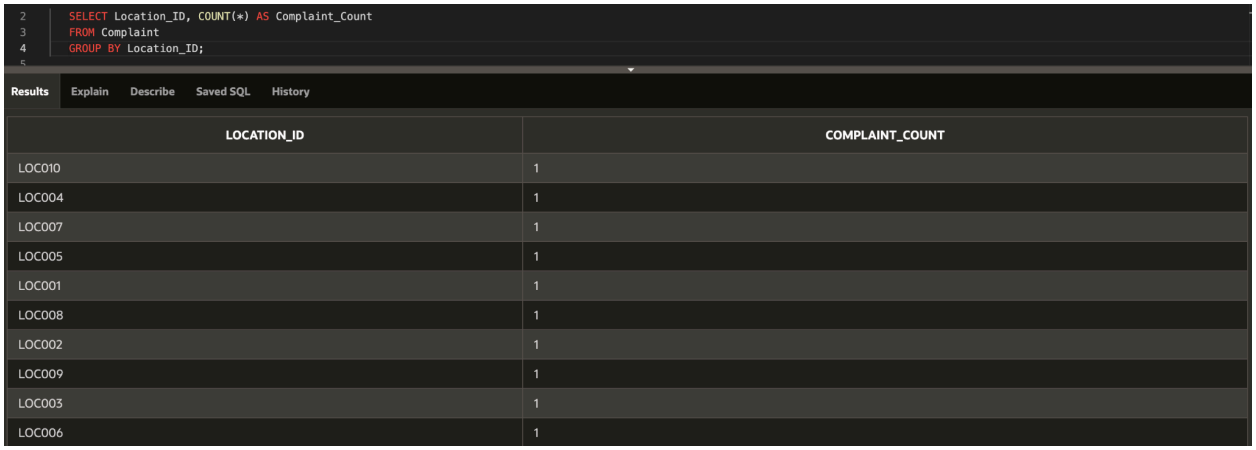
-Number of students majored in Finance

```
select Student_major, count( Student_ID) "Number of students majored in Finance"
from Student
where Student_major = 'Finance'
GROUP BY Student_major
```



-Count the number of complaints for each location and retrieve the location ID along with the corresponding complaint count

```
SELECT Location_ID, COUNT(*) AS Complaint_Count
FROM Complaint
GROUP BY Location_ID;
```



```
SELECT Location_ID, COUNT(*) AS Complaint_Count
FROM Complaint
GROUP BY Location_ID;
```

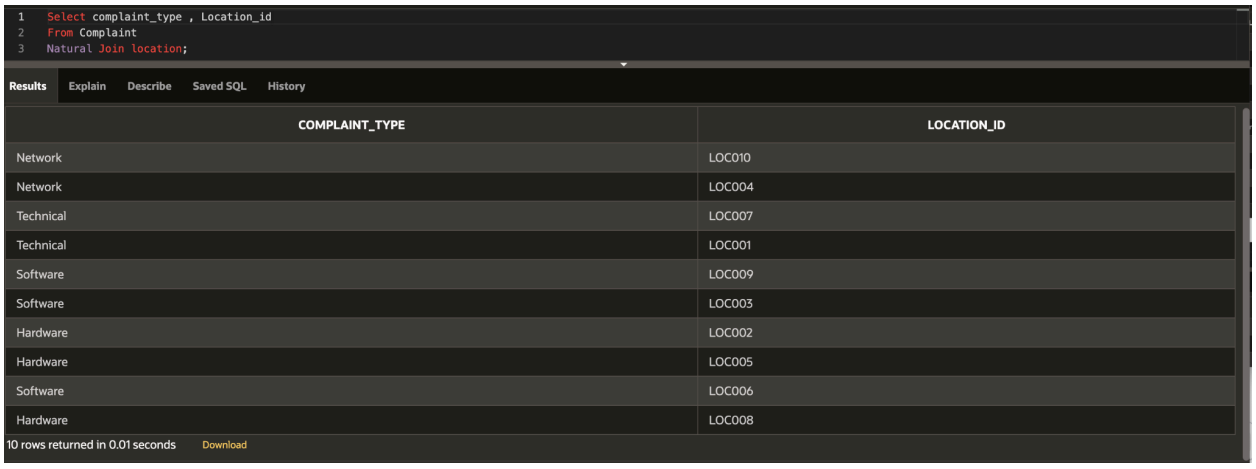
Results | Explain | Describe | Saved SQL | History

LOCATION_ID	COMPLAINT_COUNT
LOC010	1
LOC004	1
LOC007	1
LOC005	1
LOC001	1
LOC008	1
LOC002	1
LOC009	1
LOC003	1
LOC006	1

3

Display the complaint type along with its corresponding location ID

```
SELECT Complaint_Type, Location_ID
FROM Complaint
NATURAL JOIN Location;
```



```
Select complaint_type , Location_id
From Complaint
Natural Join location;
```

Results | Explain | Describe | Saved SQL | History

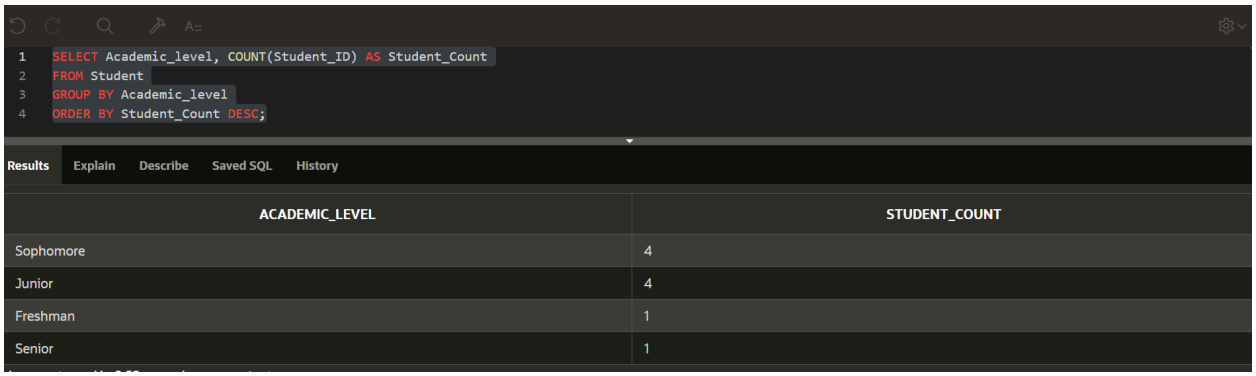
COMPLAINT_TYPE	LOCATION_ID
Network	LOC010
Network	LOC004
Technical	LOC007
Technical	LOC001
Software	LOC009
Software	LOC003
Hardware	LOC002
Hardware	LOC005
Software	LOC006
Hardware	LOC008

10 rows returned in 0.01 seconds Download

4

Retrieve the count of students in each academic level, ordered by the number of students in each level in descending order

```
SELECT Academic_level, COUNT(Student_ID) AS Student_Count
FROM Student
GROUP BY Academic_level
ORDER BY Student_Count DESC;
```



```
SELECT Academic_level, COUNT(Student_ID) AS Student_Count
FROM Student
GROUP BY Academic_level
ORDER BY Student_Count DESC;
```

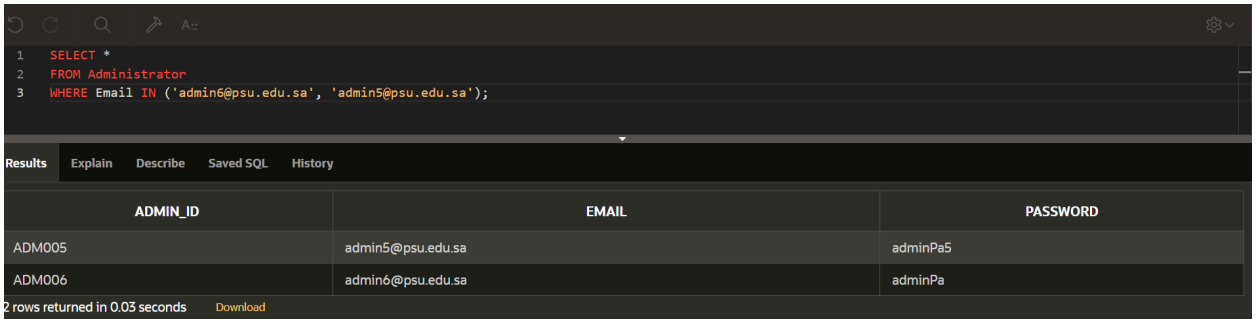
Results | Explain | Describe | Saved SQL | History

ACADEMIC_LEVEL	STUDENT_COUNT
Sophomore	4
Junior	4
Freshman	1
Senior	1

5

Retrieves all columns from the 'Administrator' table for rows where the email is either 'admin6@psu.edu.sa' or 'admin5@psu.edu.sa'

```
SELECT *
FROM Administrator
WHERE Email IN ('admin6@psu.edu.sa', 'admin5@psu.edu.sa');
```



```
SELECT *
FROM Administrator
WHERE Email IN ('admin6@psu.edu.sa', 'admin5@psu.edu.sa');
```

Results | Explain | Describe | Saved SQL | History

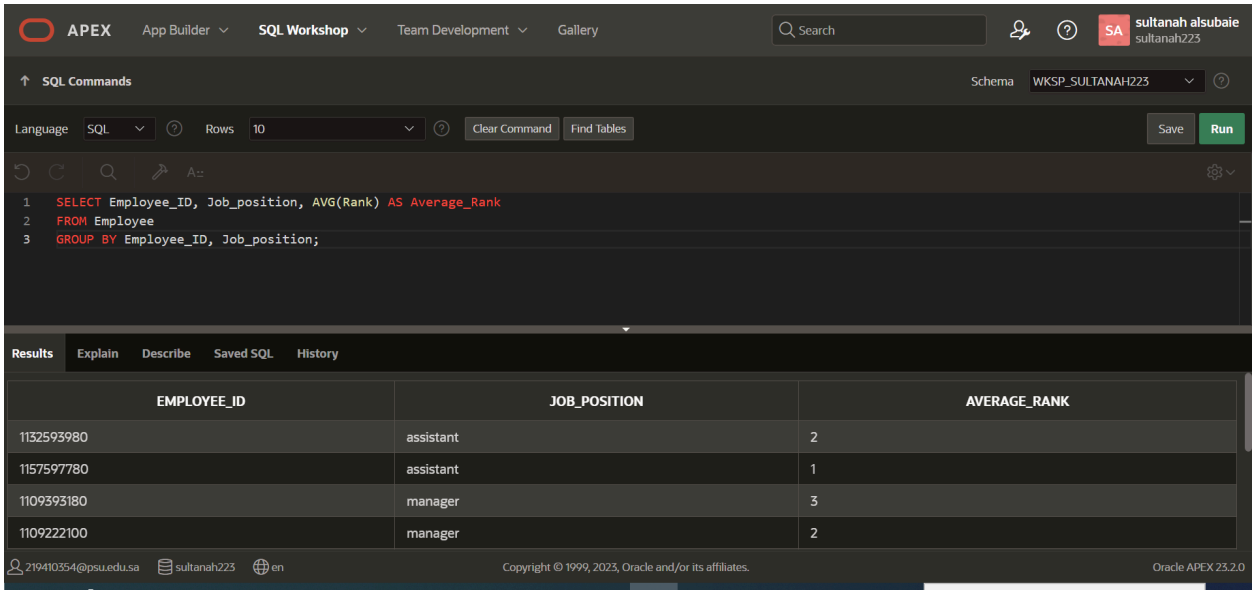
ADMIN_ID	EMAIL	PASSWORD
ADM005	admin5@psu.edu.sa	adminPa5
ADM006	admin6@psu.edu.sa	adminPa

2 rows returned in 0.03 seconds Download

6

List employees with their average performance ranking

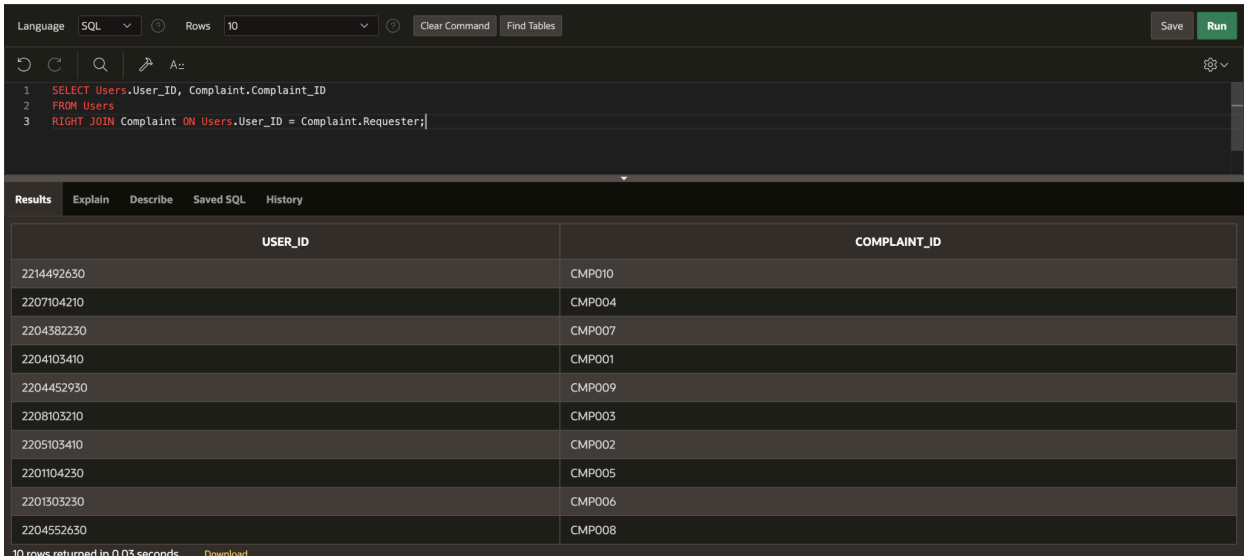
```
SELECT Employee_ID, Job_position, AVG(Rank) AS Average_Rank
FROM Employee
GROUP BY Employee_ID, Job_position;
```



7

Retrieve the User_ID and Complaint_ID, linking the Users and Complaint tables based on the User_ID and the Requester

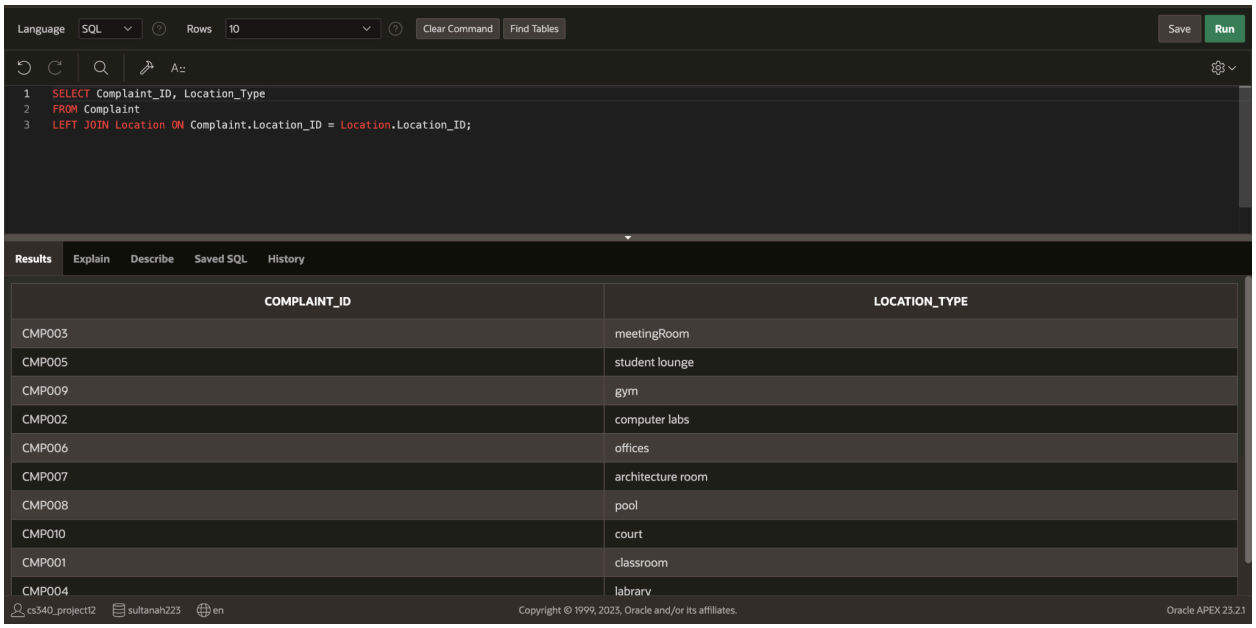
```
SELECT Users.User_ID, Complaint.Complaint_ID
FROM Users
RIGHT JOIN Complaint ON Users.User_ID = Complaint.Requester;
```



8

Retrieve the Complaint ID and Location Type

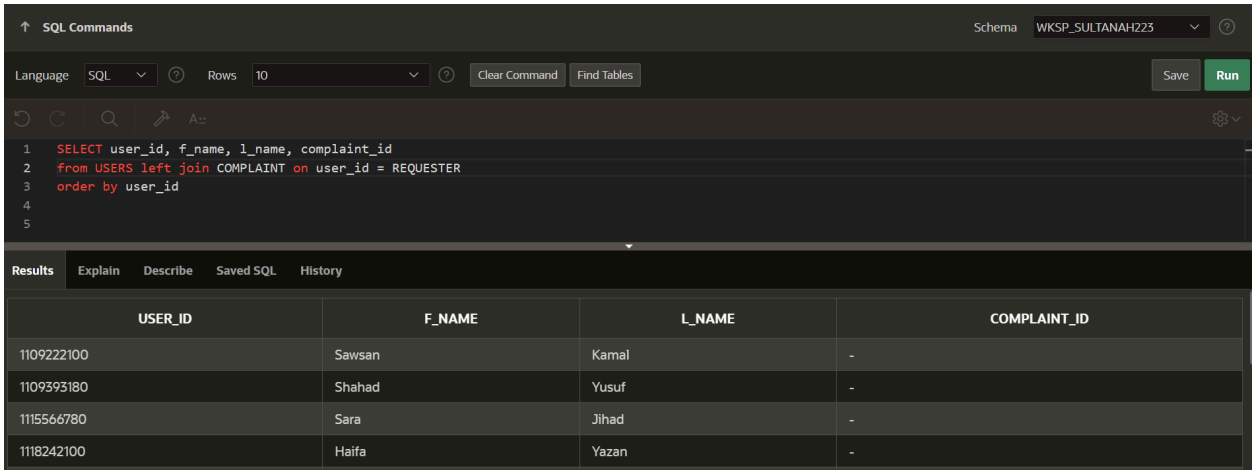
```
SELECT Complaint_ID, Location_Type
FROM Complaint
LEFT JOIN Location ON Complaint.Location_ID =
Location.Location_ID;
```



9

List all the users and the complaint ID. If the user has not registered any complaint, the user id and name should still be displayed :

```
SELECT user_id, f_name, l_name, complaint_id
from USERS left join COMPLAINT on user_id = REQUESTER
order by user_id
```

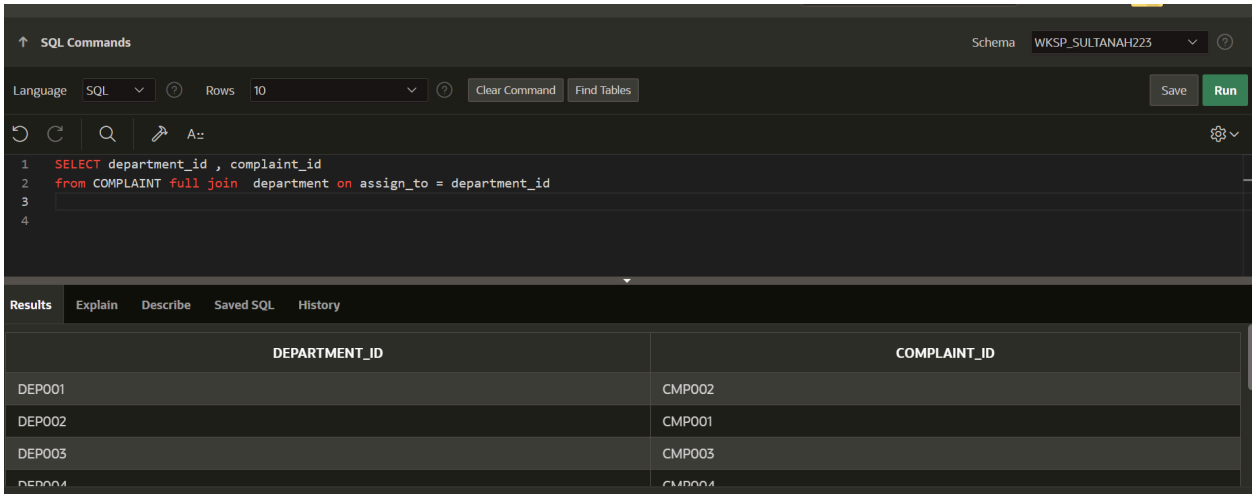


USER_ID	F_NAME	L_NAME	COMPLAINT_ID
1109222100	Sawsan	Kamal	-
1109393180	Shahad	Yusuf	-
1115566780	Sara	Jihad	-
1118242100	Haifa	Yazan	-

10

List all the complaint IDs assigned or not assigned to the departments. And display all the departments whether a complaint assigned or not to it :

```
SELECT department_id , complaint_id
from COMPLAINT full join department on assign_to = department_id
```

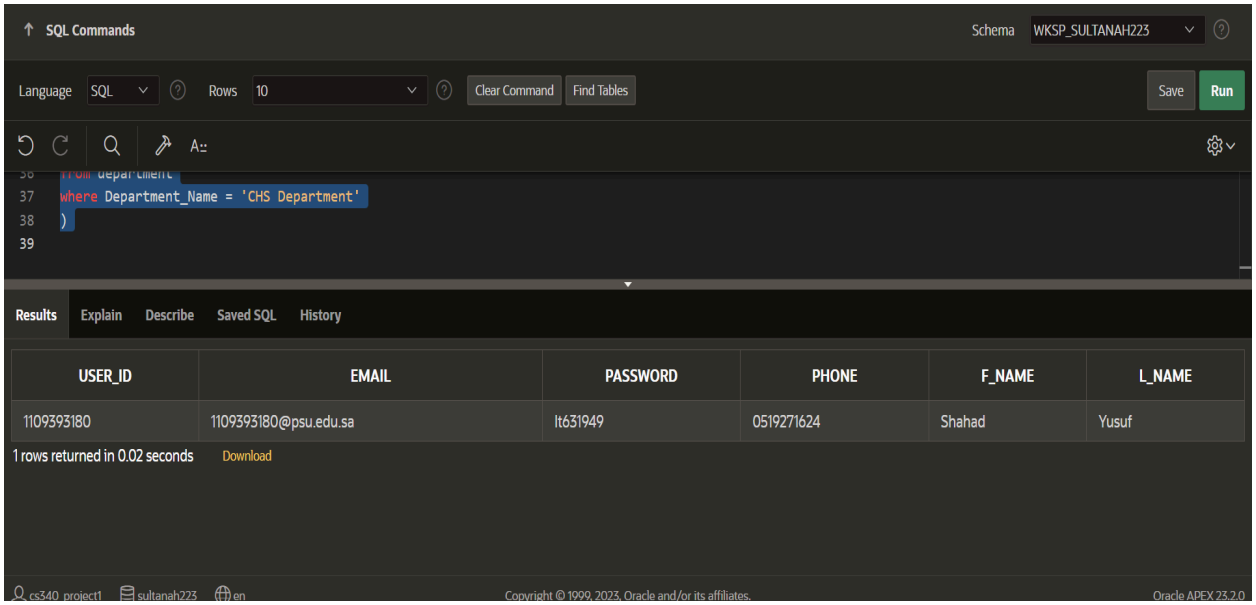


DEPARTMENT_ID	COMPLAINT_ID
DEP001	CMP002
DEP002	CMP001
DEP003	CMP003

2- Advanced queries

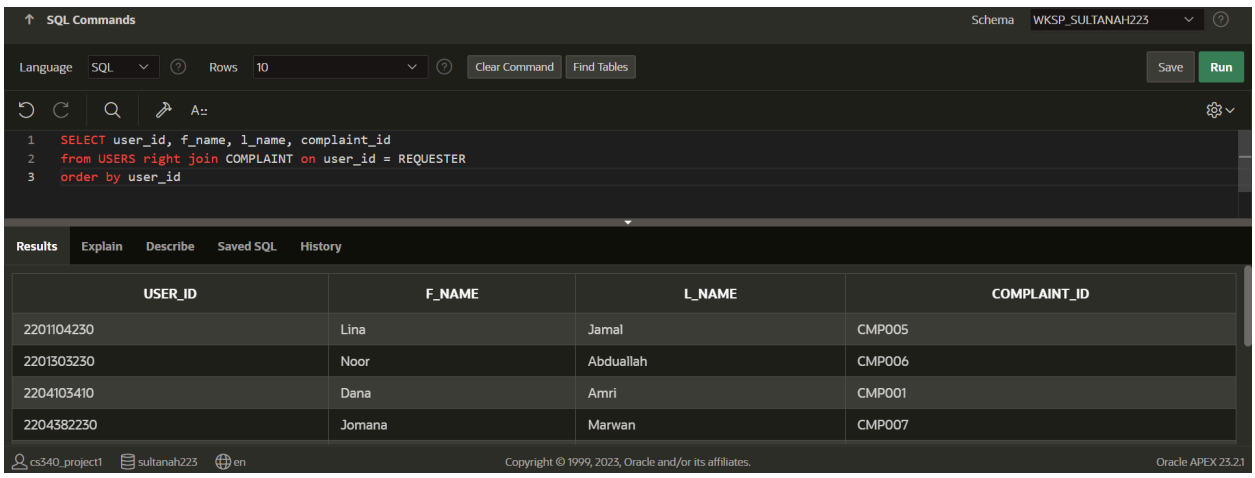
1- Show the employee that manage CHS department

```
select *
from users
where user_id in (select managed_by
from department
where Department_Name = 'CHS Department')
```



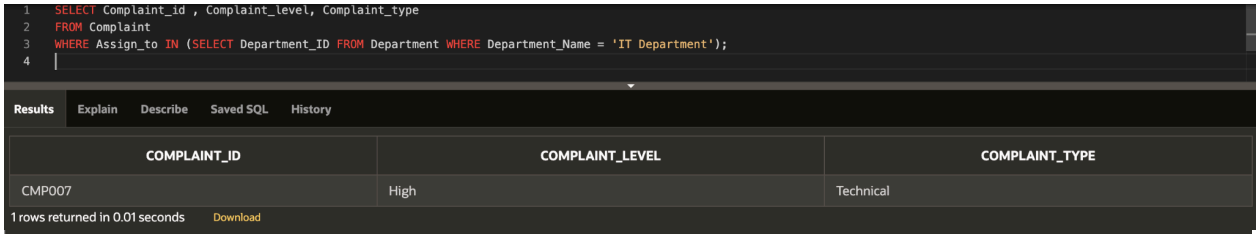
2- Show the user id , complaint type, and student majors

```
SELECT u.user_id, c.complaint_type, s.student_major
FROM USERS u
JOIN COMPLAINT c ON u.user_id = c.REQUESTER
JOIN STUDENT s ON u.user_id = s.student_id
order by user_id
```



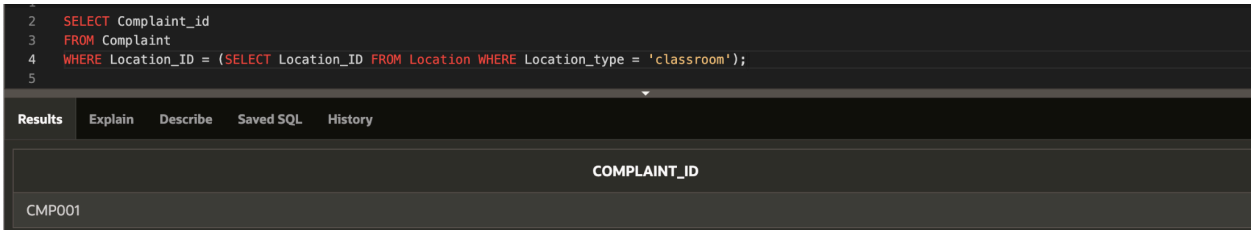
3-Retrieve the complaint id assigned to the IT Department

```
SELECT Complaint_id
FROM Complaint
WHERE Assign_to IN (SELECT Department_ID FROM Department WHERE
Department_Name = 'IT Department');
```



4- Display the Complaint id for complaints related to classrooms

```
SELECT Complaint_id
FROM Complaint
WHERE Location_ID = (SELECT Location_ID FROM Location WHERE
Location_type = 'classroom');
```



5- Retrieve the complaint id with a satisfaction rate greater than 4

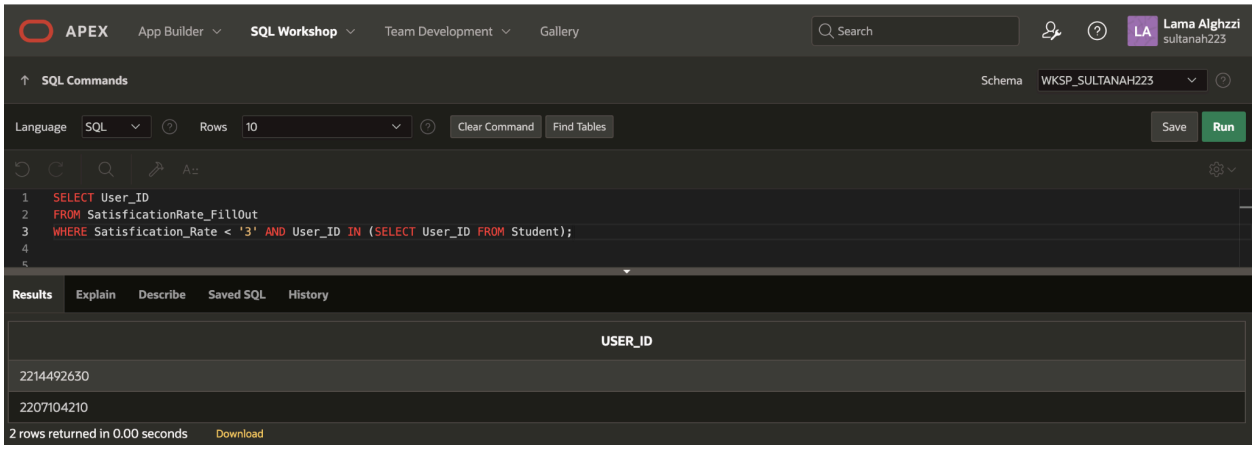
```
SELECT Complaint_ID
FROM Complaint
WHERE Complaint_ID IN (SELECT Complaint_ID FROM
SatisficationRate_FillOut WHERE Satisfication_Rate > '4');
```

6- Display the student id for students who have filled out a satisfaction rating of less than 3

```
SELECT User_ID
FROM SatisficationRate_FillOut
WHERE Satisfication_Rate < '3' AND User_ID IN (SELECT student_ID
FROM Student);
```

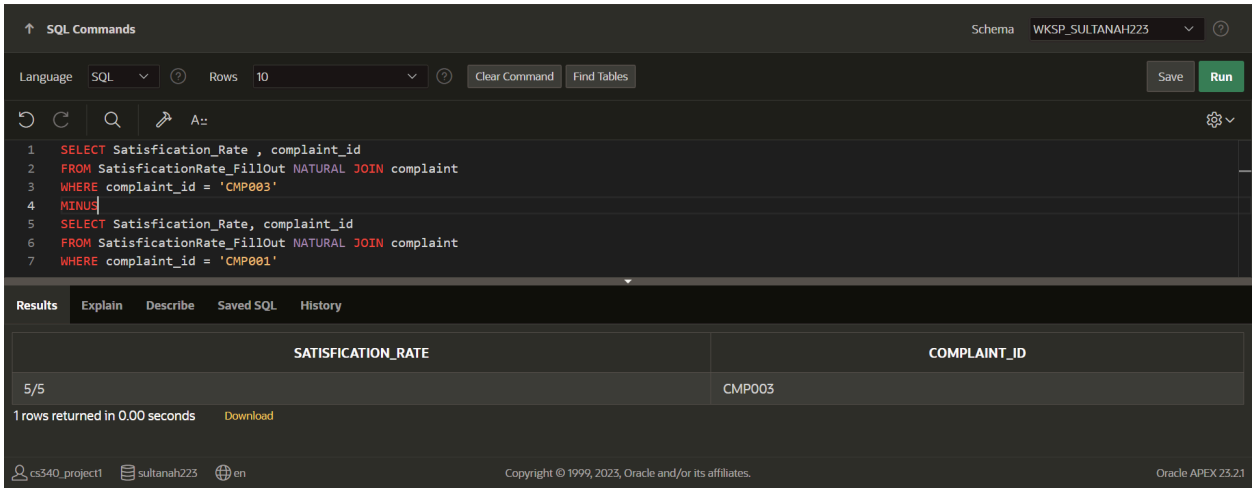
Or

```
SELECT User_ID
FROM SatisficationRate_FillOut
WHERE Satisfication_Rate < '3' AND User_ID IN (SELECT user_ID
FROM Student);
```



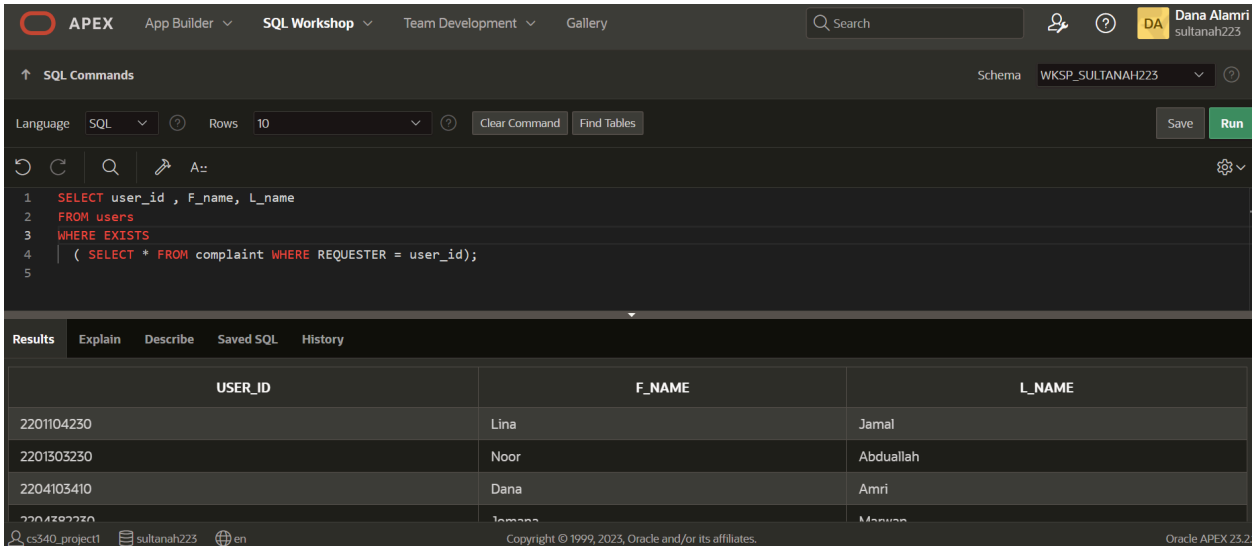
7- Show the Satisfaction Rate and complaint id for the complaint id 003 but not complaint id oo1 using MINUS

```
SELECT Satisfication_Rate , complaint_id
FROM SatisficationRate_FillOut NATURAL JOIN complaint
WHERE complaint_id = 'CMP003'
MINUS
SELECT Satisfication_Rate, complaint_id
FROM SatisficationRate_FillOut NATURAL JOIN complaint
WHERE complaint_id = 'CMP001'
```



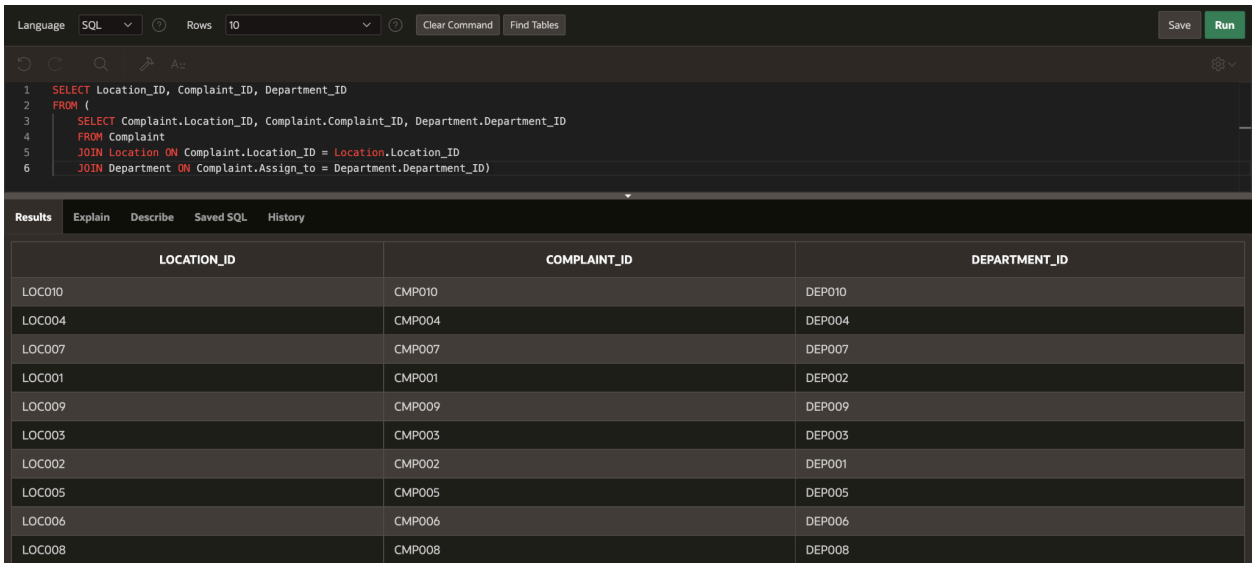
8-- Retrieve all the users who have complaints.

```
SELECT user_id , F_name, L_name
FROM users
WHERE EXISTS
   ( SELECT * FROM complaint WHERE REQUESTER = user_id);
```



9- Retrieve Location, Complaint, and Department IDs using three joins:

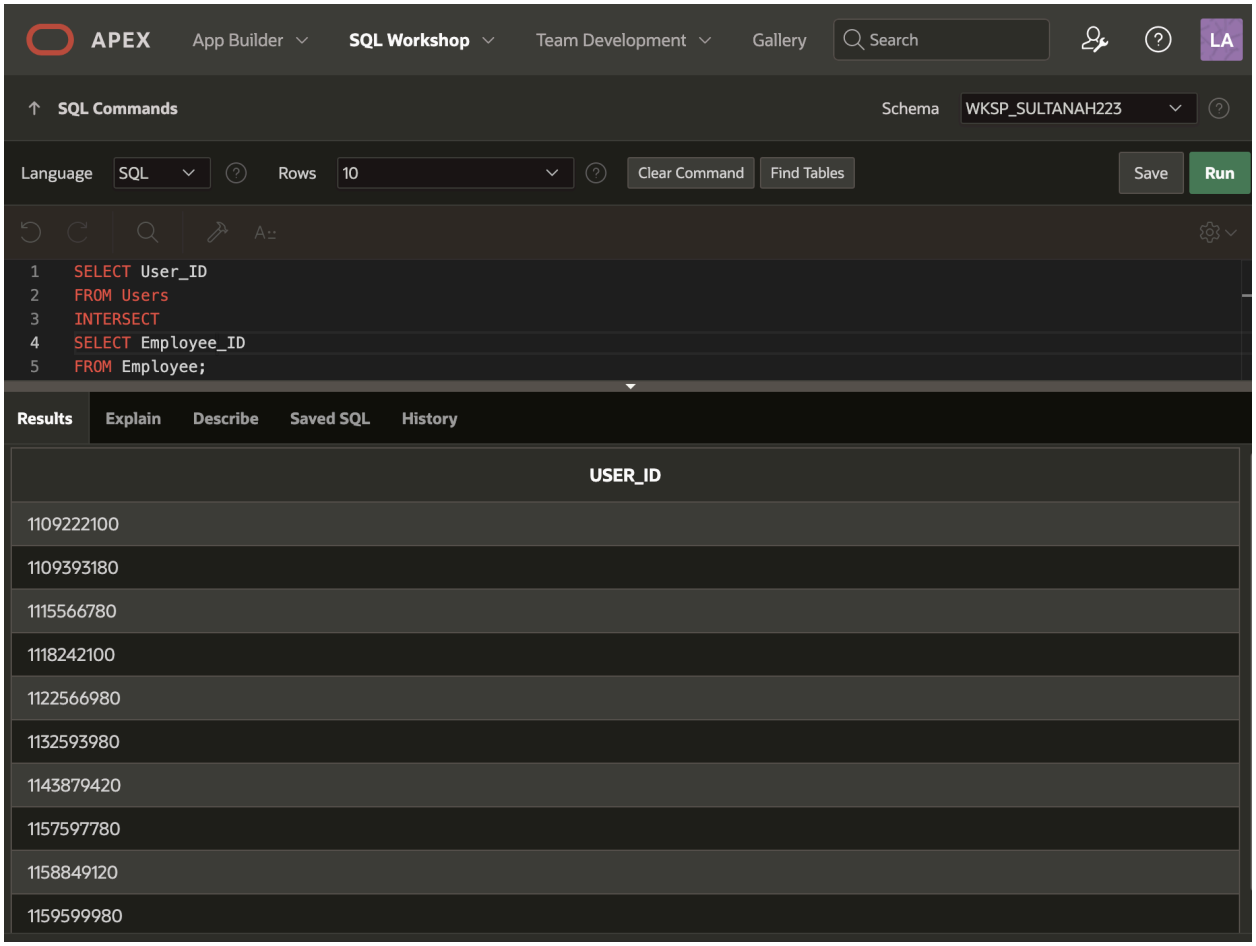
```
SELECT Location_ID, Complaint_ID, Department_ID
FROM (SELECT Complaint.Location_ID, Complaint.Complaint_ID,
Department.Department_ID
FROM Complaint
JOIN Location ON Complaint.Location_ID = Location.Location_ID
JOIN Department ON Complaint.Assign_to =
Department.Department_ID)
```



LOCATION_ID	COMPLAINT_ID	DEPARTMENT_ID
LOC010	CMP010	DEP010
LOC004	CMP004	DEP004
LOC007	CMP007	DEP007
LOC001	CMP001	DEP002
LOC009	CMP009	DEP009
LOC003	CMP003	DEP003
LOC002	CMP002	DEP001
LOC005	CMP005	DEP005
LOC006	CMP006	DEP006
LOC008	CMP008	DEP008

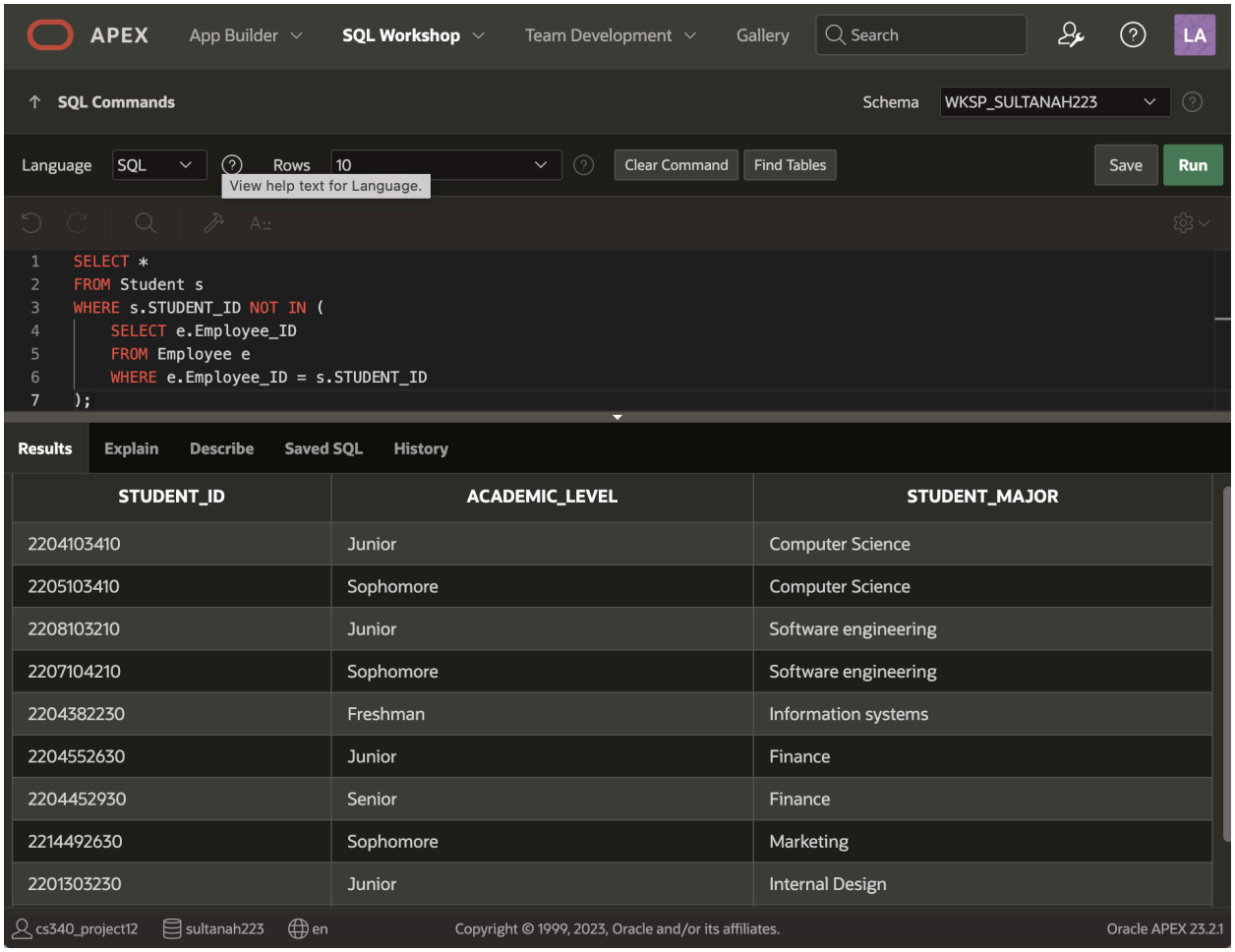
10- Find the users that they are employees

```
SELECT User_ID
FROM Users
INTERSECT
SELECT User_ID
FROM Employee;
```



11- Find students who are not employees

```
SELECT *
FROM Student s
WHERE s.STUDENT_ID NOT IN (
    SELECT e.Employee_ID
    FROM Employee e
    WHERE e.Employee_ID = s.STUDENT_ID);
```



Contribution of the team members to phase 5 and strategy used to demonstrate teamwork :

Dana	Creating simple and advanced queries
Lama	Creating simple and advanced queries
Sultanah	Creating simple and advanced queries