Mental Health Support System

Course Code: CSE-2424

Course Name: Database Management System Lab

Submitted To:

Dr. Mohammad Aman Ullah

Submitted By:

Wafi Alam Fathema

ID: C233509

Semester: 4th

Section: 4CF

Department of CSE, IIUC

📌 Introduction:

Mental health has become an increasingly important concern in today's fast-paced and stressful world. Due to the rise in anxiety, depression, and other mental health conditions, many individuals now seek online or in-person therapy and counseling services. To manage such a system effectively, a structured and efficient database management system is essential.

This project, titled Mental Health Support System, focuses on designing and implementing a relational database to digitally handle all essential components of mental health service management—such as user registration, counselor information, session scheduling, billing, and feedback management.

📌 Objective:

The objective of this project is to ensure that the platform can support both patients and counselors in a scalable, secure, and normalized environment through SQL-based backend logic.

📌 Key Features:

- Patient/user registration and counselor profile management.
- Session booking, rescheduling, and status tracking.
- Feedback system for quality improvement.
- Billing and payment tracking per session or user.
- Data normalization and integrity ensured via constraints and PL/SQL components.

The system is implemented using Oracle SQL and PL/SQL, and all essential phases of database development including design, normalization, DML queries, subqueries, PL/SQL procedures/functions/triggers are incorporated.

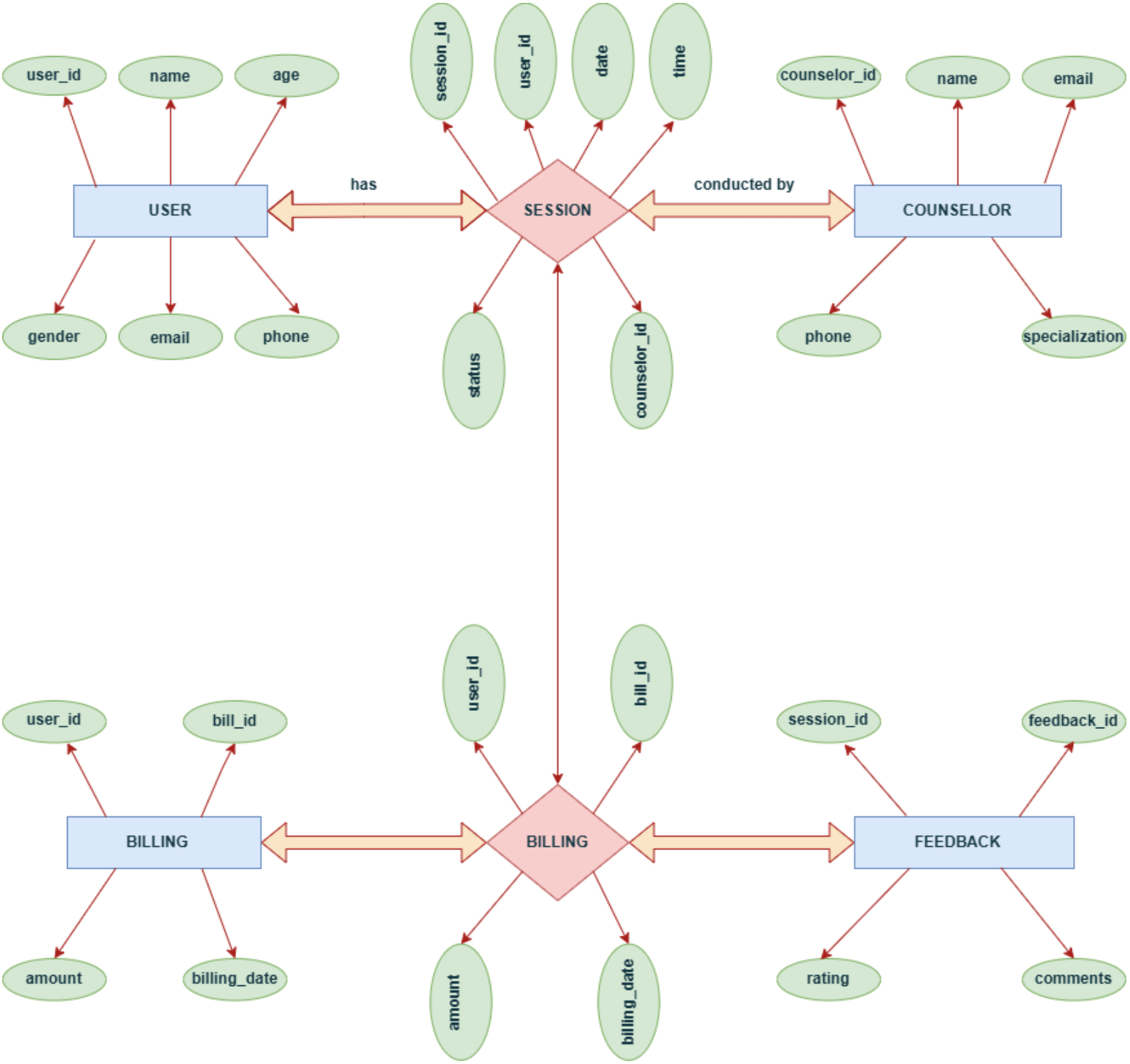
📌 Scope of the Project:

- Designed for small to medium-scale mental health institutions or clinics.
- Can be used by university wellness centers, online therapy platforms, and NGOs.
- Modular architecture that allows easy customization.
- Could be extended to include notifications, prescription tracking, or session notes in future.

📌 Tools & Technologies Used:

- Database: Oracle 21c / Oracle SQL Developer / MySQL Workbench
- Language: SQL, PL/SQL, DML
- Design: ERD (Entity Relationship Diagram), Normalization
- Reporting: Microsoft Word / PDF

📌 ER Diagram:


user_id
name
age
USER
gender
email
phone
has
session_id
user_id
date
time
SESSION
status
counselor_id
conducted by
counselor_id
name
email
COUNSELLOR
phone
specialization
user_id
bill_id
BILLING
amount
billing_date
user_id
bill_id
BILLING
amount
billing_date
session_id
feedback_id
FEEDBACK
rating
comments

📌 ER Diagram Description:

1. The Entity-Relationship (ER) diagram visually represents the logical structure of the Mental Health Support System database.
2. It illustrates the main entities (Users, Counselors, Sessions, Feedback, Billing) and the relationships among them.
3. Each entity has a primary key to uniquely identify its records, and necessary attributes that define the entity's characteristics (e.g., name, email, age, specialization).
4. Relationships such as "User books Session", "Counselor conducts Session", "Session has Feedback", and "User generates Billing" are clearly mapped with proper foreign key references.
5. The ER Diagram uses cardinality to define how many records in one entity relate to records in another:
    - One User → Many Sessions
    - One Counselor → Many Sessions
    - One Session → One Feedback
    - One User → Many Billings
6. All many-to-one and one-to-one relationships are normalized and resolved via foreign key placement to reduce redundancy.
7. The ERD helps to simplify the database development process by providing a blueprint that ensures data consistency and referential integrity.
8. This ER Diagram is foundational to the successful implementation of SQL tables, queries, and PL/SQL programs used throughout the system.

📌 Normalization Table:

🔶 What is Normalization?

Normalization is the process of organizing data in a database to reduce redundancy and improve data integrity. It involves dividing large, complex tables into smaller, simpler ones and clearly defining relationships between them.

The goal of normalization is to eliminate common data anomalies like:

- Insertion Anomalies
- Update Anomalies
- Deletion Anomalies

By using normalization, we ensure our database is efficient, consistent, and logically organized.

Here the final normalization table is given below:

Table	1NF (First Normal Form)	2NF (Second Normal Form)	3NF (Third Normal Form)
Users	Each column contains atomic values	All non-key attributes depend on primary key	No transitive dependencies
Counselors	All values are atomic	All non-key attributes are fully dependent on primary key	No transitive dependencies
Sessions	Each field has atomic values (date, time etc.)	Date, time, status depend only on composite primary key	No transitive dependency on counselor/user name
Feedback	One feedback per row, all atomic values	Rating and comment depend on session_id (primary key)	No transitive dependencies
Billing	All fields are atomic	Amount, date depend only on bill_id	User info not repeated (linked via foreign key)

Here,

🔶 UNF (Unnormalized Form):

In UNF, data is raw and unstructured. It may contain:

- Multi-valued attributes
- Repeating groups
- No formal structure

🟢 Example (UNF):

user_id	name	sessions
1	Ayesha Khan	[(2024-06-01, Dr. Mehnaz), (2024-06-07, Dr. Saif)]

This kind of structure is difficult to maintain and analyze.

🔶 1NF (First Normal Form):

- All data must be atomic (indivisible).
- No repeating groups are allowed.

🟢 **Example:**

user_id	name	session_date	counselor_name
1	Ayesha Khan	2024-06-01	Dr. Mehnaz
1	Ayesha Khan	2024-06-07	Dr. Saif

Split the multi-valued “sessions” column into separate rows:

🔶 **2NF (Second Normal Form):**

2NF = 1NF + No Partial Dependency

A table is in 2NF when:

- It is already in 1NF
- All non-key attributes are fully functionally dependent on the primary key

🟢 **Example:**

If session_id is the primary key, then fields like date, time, status should depend entirely on session_id — not partially on user_id or counselor_id.

🔶 3NF (Third Normal Form):

3NF = 2NF + No Transitive Dependency

A table is in 3NF when:

- It is in 2NF
- No non-key attribute depends on another non-key attribute (i.e., no transitive dependencies)

🟢 Example:

If a Feedback table stores:
| feedback_id | session_id | rating | counselor_name |
Then counselor_name is transitively dependent on session_id (via counselor_id). So counselor_name should be removed and fetched using joins instead.

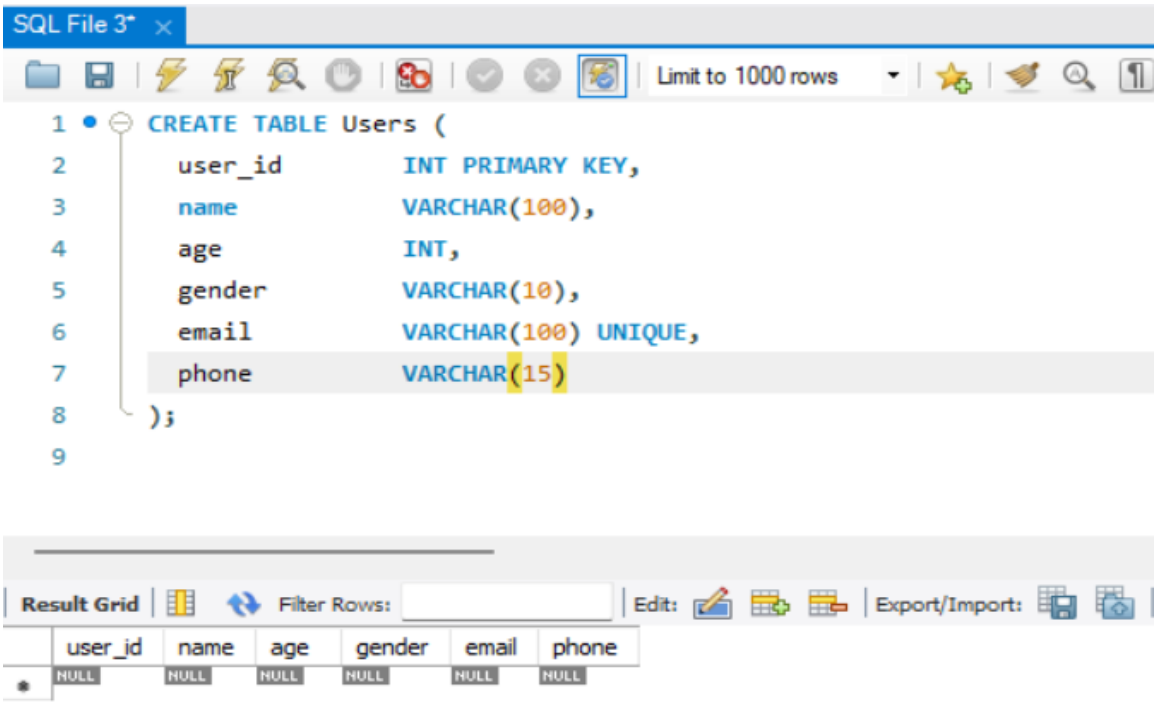
🔶 Final Conclusion:

In the Mental Health Support System database:

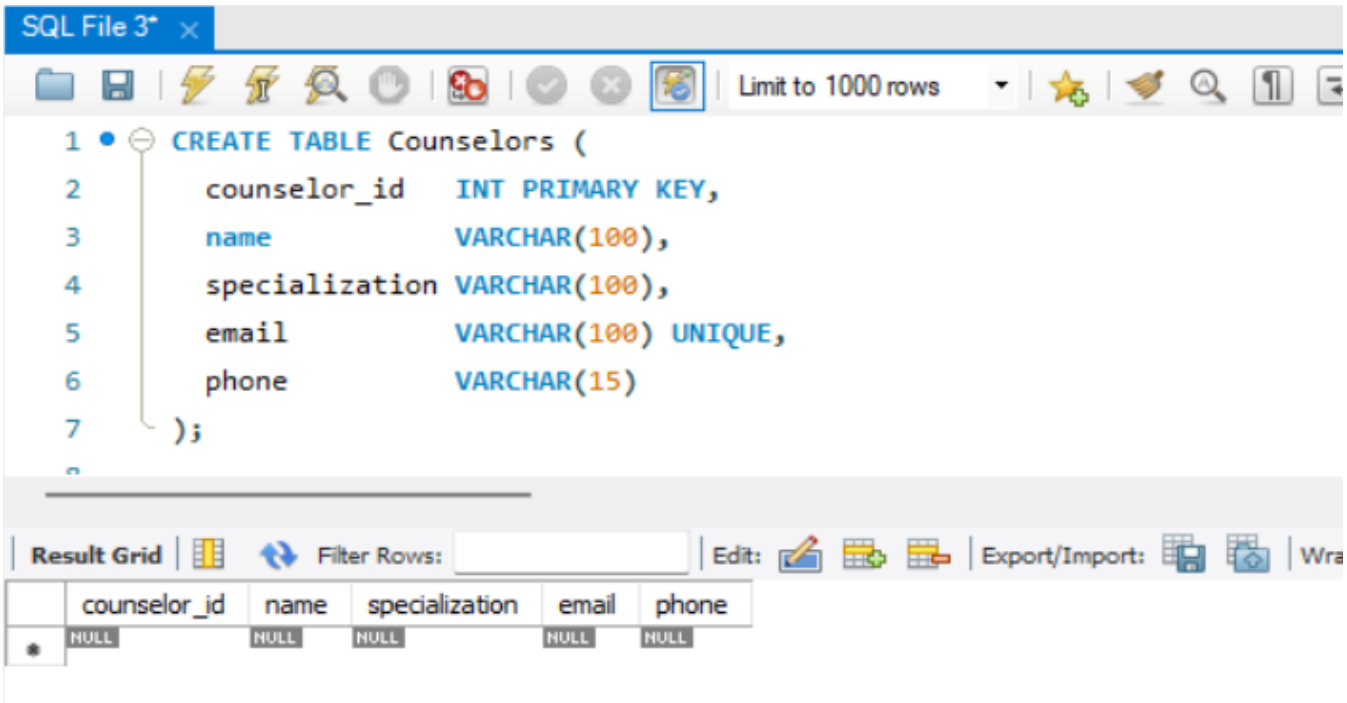
- All five tables (Users, Counselors, Sessions, Feedback, Billing) are designed in 1NF (atomic), 2NF (no partial dependency), and 3NF (no transitive dependency).
- This ensures better scalability, maintenance, and consistent data structure.

📌 Table Structures With Constraints (DDL):

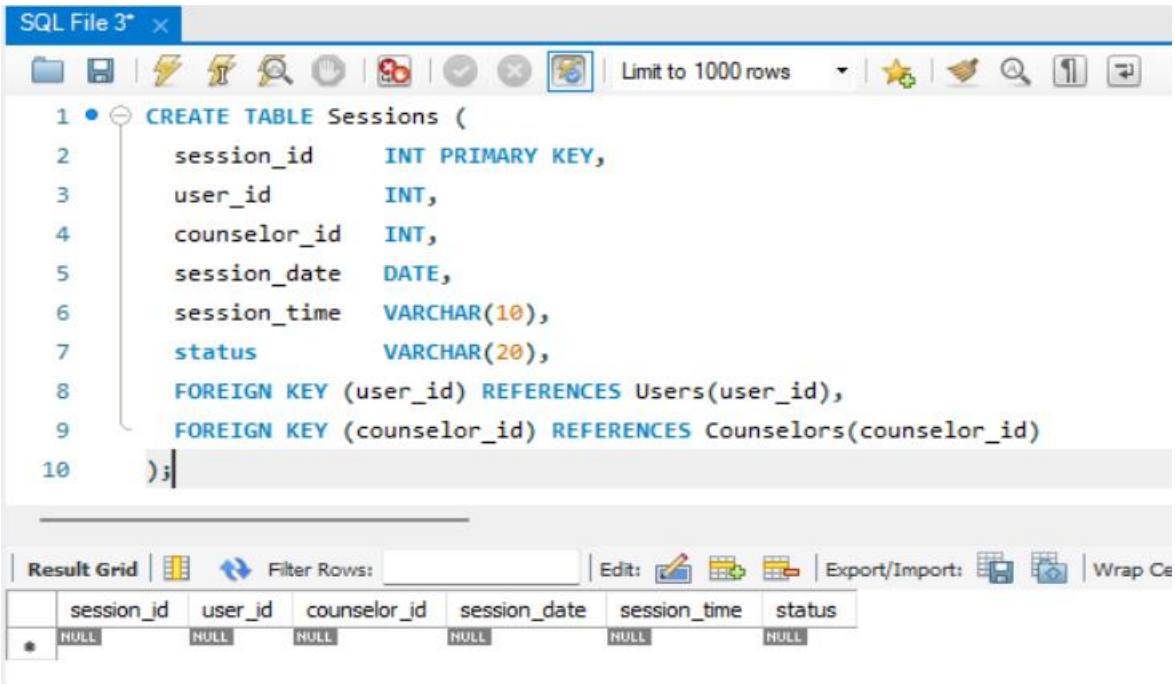
🔷 CREATE TABLE Users (... PRIMARY KEY):



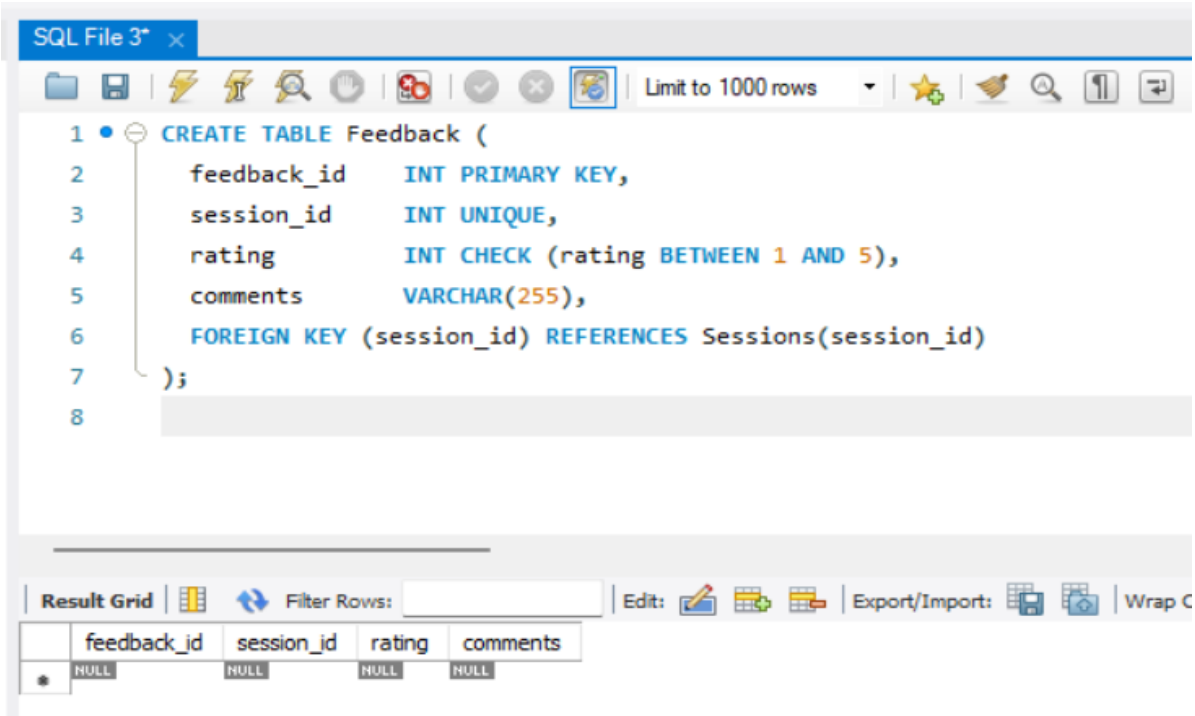
🔷 CREATE TABLE Counselor (... PRIMARY KEY):



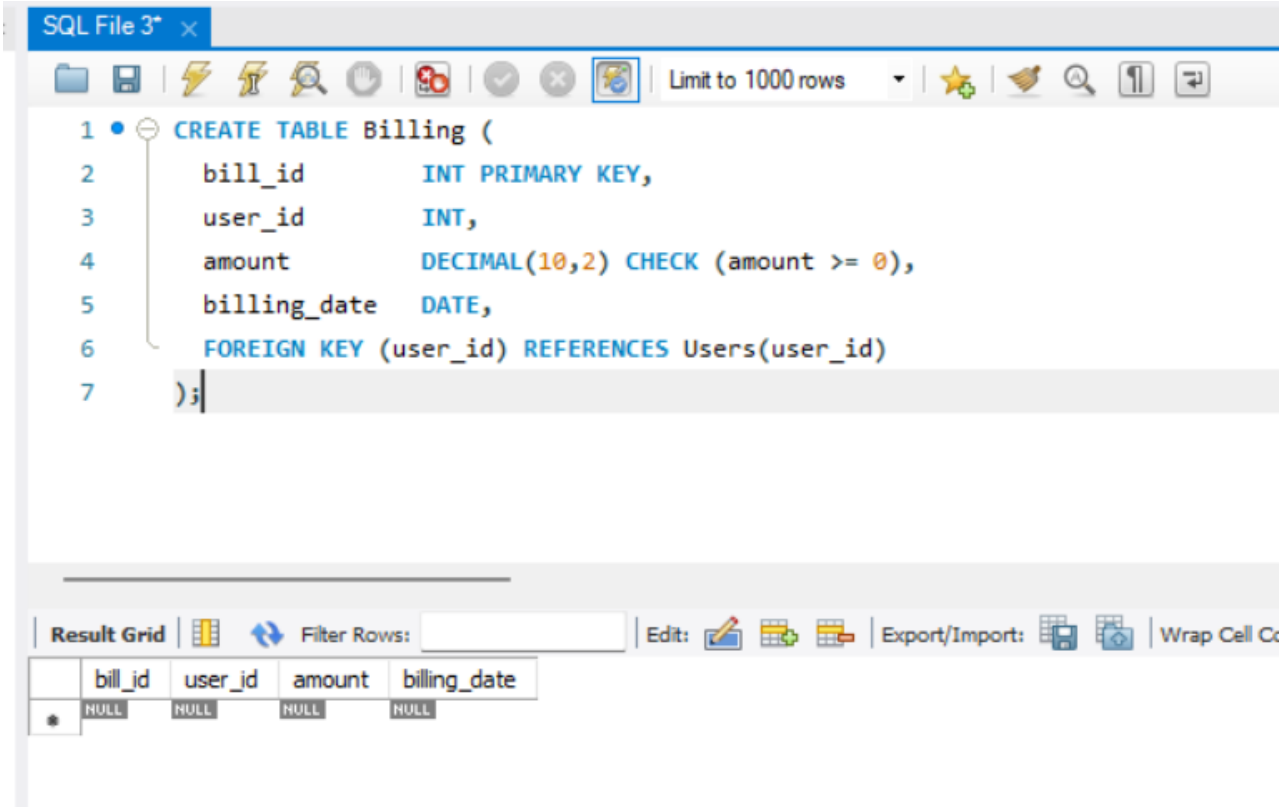
🔷 CREATE TABLE Sessions (... PRIMARY KEY, FOREIGN KEY to Users & Counselors):



🔷 CREATE TABLE Feedback (... PRIMARY KEY, FOREIGN KEY to Sessions):



🔷 CREATE TABLE Billing (... PRIMARY KEY, FOREIGN KEY to Users):



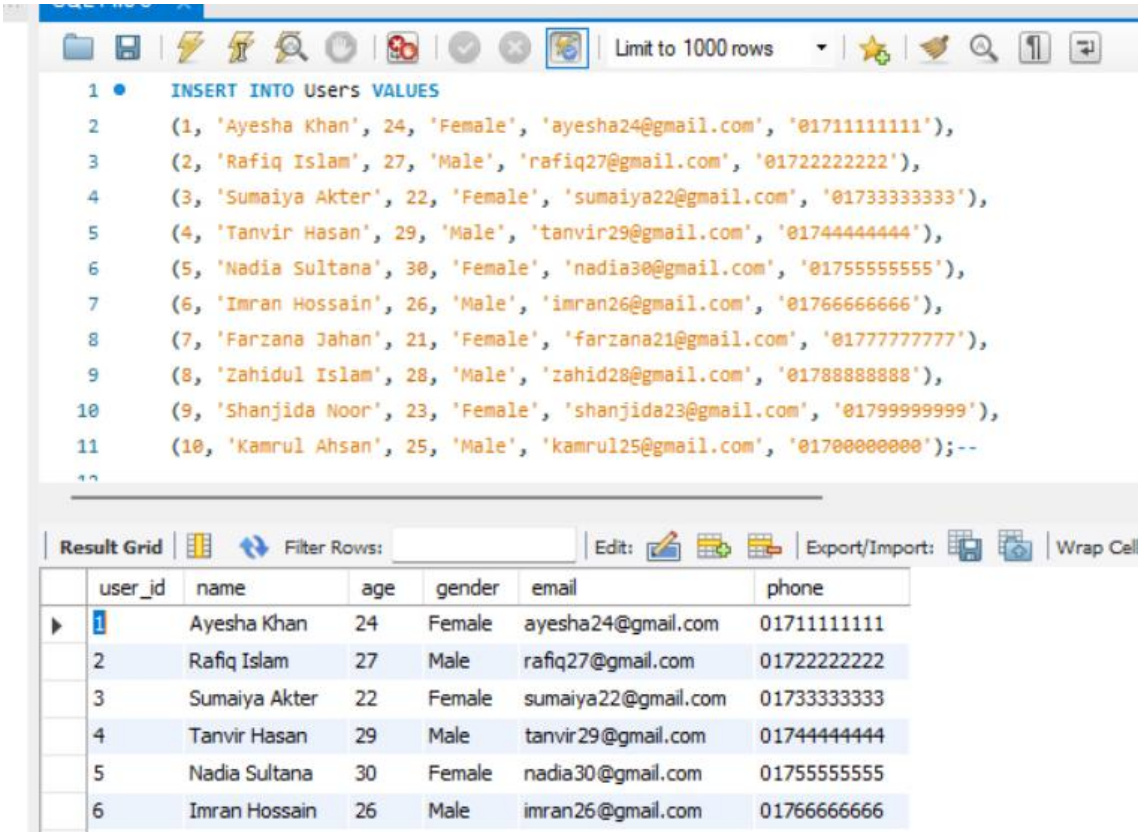
Below is a brief summary of the database schema and constraints shown in the above screenshots:

✅ 5 relational tables are used with proper Primary and Foreign Keys.
✅ Data types like INT, VARCHAR2, DATE, and DECIMAL are chosen appropriately.
✅ Constraints (PK, FK, UNIQUE, CHECK) ensure data accuracy and integrity.
✅ Relationships maintain normalized and connected structure between tables.

👉 This structure supports a reliable Mental Health Support System.

📌 Inserted Data:

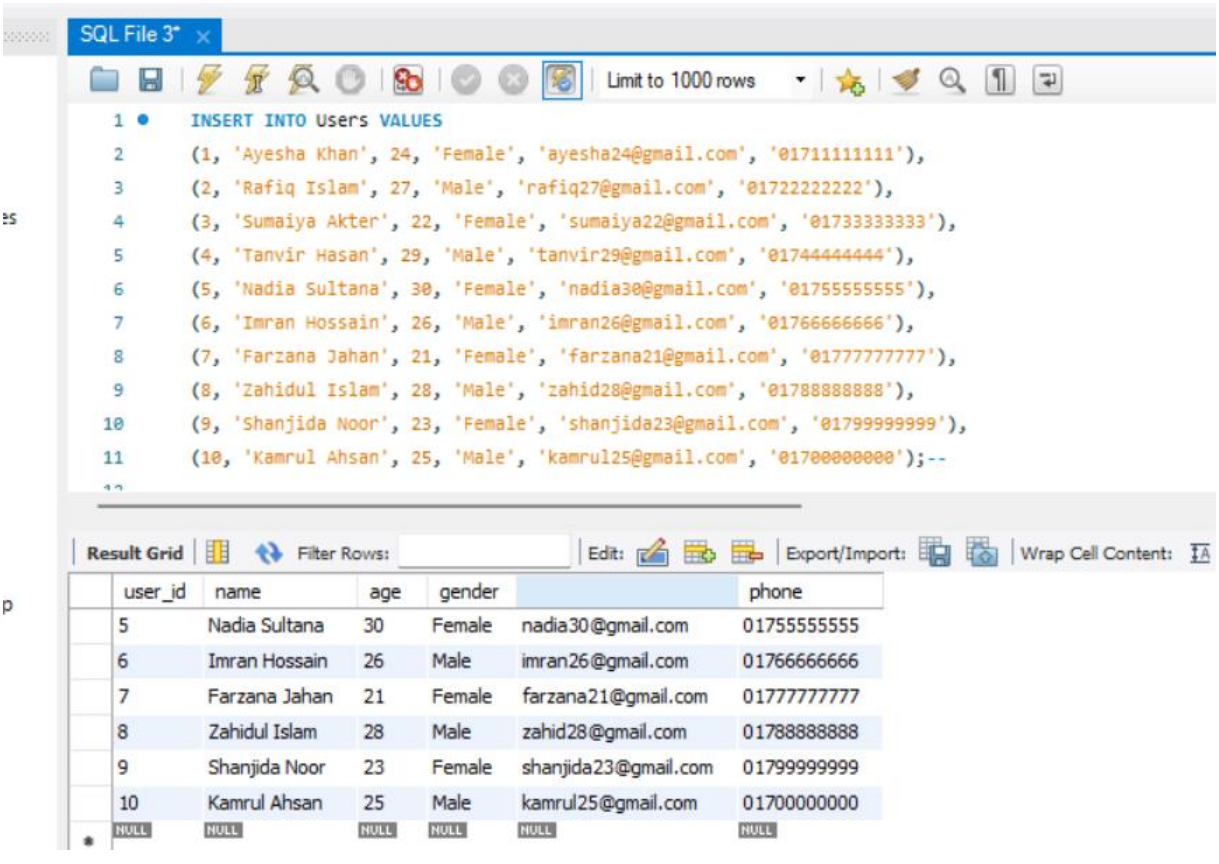
🟢 1. Users Table:



```
INSERT INTO Users VALUES
(1, 'Ayesha Khan', 24, 'Female', 'ayesha24@gmail.com', '01711111111'),
(2, 'Rafiq Islam', 27, 'Male', 'rafiq27@gmail.com', '01722222222'),
(3, 'Sumaiya Akter', 22, 'Female', 'sumaiya22@gmail.com', '01733333333'),
(4, 'Tanvir Hasan', 29, 'Male', 'tanvir29@gmail.com', '01744444444'),
(5, 'Nadia Sultana', 30, 'Female', 'nadia30@gmail.com', '01755555555'),
(6, 'Imran Hossain', 26, 'Male', 'imran26@gmail.com', '01766666666'),
(7, 'Farzana Jahan', 21, 'Female', 'farzana21@gmail.com', '01777777777'),
(8, 'Zahidul Islam', 28, 'Male', 'zahid28@gmail.com', '01788888888'),
(9, 'Shanjida Noor', 23, 'Female', 'shanjida23@gmail.com', '01799999999'),
(10, 'Kamrul Ahsan', 25, 'Male', 'kamrul25@gmail.com', '01700000000');--
```

Result Grid

user_id	name	age	gender	email	phone
1	Ayesha Khan	24	Female	ayesha24@gmail.com	01711111111
2	Rafiq Islam	27	Male	rafiq27@gmail.com	01722222222
3	Sumaiya Akter	22	Female	sumaiya22@gmail.com	01733333333
4	Tanvir Hasan	29	Male	tanvir29@gmail.com	01744444444
5	Nadia Sultana	30	Female	nadia30@gmail.com	01755555555
6	Imran Hossain	26	Male	imran26@gmail.com	01766666666



SQL File 3*

```
INSERT INTO Users VALUES
(1, 'Ayesha Khan', 24, 'Female', 'ayesha24@gmail.com', '01711111111'),
(2, 'Rafiq Islam', 27, 'Male', 'rafiq27@gmail.com', '01722222222'),
(3, 'Sumaiya Akter', 22, 'Female', 'sumaiya22@gmail.com', '01733333333'),
(4, 'Tanvir Hasan', 29, 'Male', 'tanvir29@gmail.com', '01744444444'),
(5, 'Nadia Sultana', 30, 'Female', 'nadia30@gmail.com', '01755555555'),
(6, 'Imran Hossain', 26, 'Male', 'imran26@gmail.com', '01766666666'),
(7, 'Farzana Jahan', 21, 'Female', 'farzana21@gmail.com', '01777777777'),
(8, 'Zahidul Islam', 28, 'Male', 'zahid28@gmail.com', '01788888888'),
(9, 'Shanjida Noor', 23, 'Female', 'shanjida23@gmail.com', '01799999999'),
(10, 'Kamrul Ahsan', 25, 'Male', 'kamrul25@gmail.com', '01700000000');--
```

Result Grid

user_id	name	age	gender	email	phone
5	Nadia Sultana	30	Female	nadia30@gmail.com	01755555555
6	Imran Hossain	26	Male	imran26@gmail.com	01766666666
7	Farzana Jahan	21	Female	farzana21@gmail.com	01777777777
8	Zahidul Islam	28	Male	zahid28@gmail.com	01788888888
9	Shanjida Noor	23	Female	shanjida23@gmail.com	01799999999
10	Kamrul Ahsan	25	Male	kamrul25@gmail.com	01700000000
NULL	NULL	NULL	NULL	NULL	NULL

2. Counselors Table:

SQL File 3*

Limit to 1000 rows

```
INSERT INTO Counselors VALUES
(1, 'Dr. Mehnaz Rahman', 'Clinical Psychologist', 'mehnaz@clinic.com', '01811111111'),
(2, 'Dr. Saif Hossain', 'Counseling Psychologist', 'saif@therapy.com', '01822222222'),
(3, 'Dr. Tanjina Sultana', 'Child Psychologist', 'tanjina@mindcare.com', '01833333333'),
(4, 'Dr. Farid Alam', 'Therapist', 'farid@health.com', '01844444444'),
(5, 'Dr. Nusrat Jahan', 'Clinical Psychologist', 'nusrat@clinic.com', '01855555555'),
(6, 'Dr. Nabil Chowdhury', 'Neuropsychologist', 'nabil@neuro.com', '01866666666'),
(7, 'Dr. Lamiya Kabir', 'Behavioral Therapist', 'lamiya@therapy.com', '01877777777'),
(8, 'Dr. Salman Reza', 'Psychiatrist', 'salman@mental.com', '01888888888'),
(9, 'Dr. Maruf Ahmed', 'Counseling Psychologist', 'maruf@therapy.com', '01899999999'),
(10, 'Dr. Tasnim Akter', 'Therapist', 'tasnim@care.com', '01800000000');
```

Result Grid | Filter Rows: | Edit: | Export/Import: | Wrap Cell Content:

counselor_id	name	specialization	email	phone
1	Dr. Mehnaz Rahman	Clinical Psychologist	mehnaz@clinic.com	01811111111
2	Dr. Saif Hossain	Counseling Psychologist	saif@therapy.com	01822222222
3	Dr. Tanjina Sultana	Child Psychologist	tanjina@mindcare.com	01833333333
4	Dr. Farid Alam	Therapist	farid@health.com	01844444444
5	Dr. Nusrat Jahan	Clinical Psychologist	nusrat@clinic.com	01855555555
6	Dr. Nabil Chowdhury	Neuropsychologist	nabil@neuro.com	01866666666
7	Dr. Lamiya Kabir	Behavioral Therapist	lamiya@therapy.com	01877777777

SQL File 3*

Limit to 1000 rows

```
INSERT INTO Counselors VALUES
(1, 'Dr. Mehnaz Rahman', 'Clinical Psychologist', 'mehnaz@clinic.com', '01811111111'),
(2, 'Dr. Saif Hossain', 'Counseling Psychologist', 'saif@therapy.com', '01822222222'),
(3, 'Dr. Tanjina Sultana', 'Child Psychologist', 'tanjina@mindcare.com', '01833333333'),
(4, 'Dr. Farid Alam', 'Therapist', 'farid@health.com', '01844444444'),
(5, 'Dr. Nusrat Jahan', 'Clinical Psychologist', 'nusrat@clinic.com', '01855555555'),
(6, 'Dr. Nabil Chowdhury', 'Neuropsychologist', 'nabil@neuro.com', '01866666666'),
(7, 'Dr. Lamiya Kabir', 'Behavioral Therapist', 'lamiya@therapy.com', '01877777777'),
(8, 'Dr. Salman Reza', 'Psychiatrist', 'salman@mental.com', '01888888888'),
(9, 'Dr. Maruf Ahmed', 'Counseling Psychologist', 'maruf@therapy.com', '01899999999'),
(10, 'Dr. Tasnim Akter', 'Therapist', 'tasnim@care.com', '01800000000');
```

Result Grid | Filter Rows: | Edit: | Export/Import: | Wrap Cell Conte

counselor_id	name	specialization	email	phone
4	Dr. Farid Alam	Therapist	farid@health.com	01844444444
5	Dr. Nusrat Jahan	Clinical Psychologist	nusrat@clinic.com	01855555555
6	Dr. Nabil Chowdhury	Neuropsychologist	nabil@neuro.com	01866666666
7	Dr. Lamiya Kabir	Behavioral Therapist	lamiya@therapy.com	01877777777
8	Dr. Salman Reza	Psychiatrist	salman@mental.com	01888888888
9	Dr. Maruf Ahmed	Counseling Psychologist	maruf@therapy.com	01899999999
10	Dr. Tasnim Akter	Therapist	tasnim@care.com	01800000000
NULL	NULL	NULL	NULL	NULL

3. Sessions Table:

SQL File 3*

Limit to 1000 rows

```
INSERT INTO Sessions VALUES
(1, 1, 1, DATE '2024-06-01', '10:00', 'Completed'),
(2, 2, 2, DATE '2024-06-02', '11:00', 'Scheduled'),
(3, 3, 3, DATE '2024-06-03', '09:00', 'Completed'),
(4, 4, 4, DATE '2024-06-04', '13:00', 'Cancelled'),
(5, 5, 5, DATE '2024-06-05', '15:00', 'Completed'),
(6, 6, 6, DATE '2024-06-06', '10:30', 'Scheduled'),
(7, 7, 7, DATE '2024-06-07', '11:45', 'Completed'),
(8, 8, 8, DATE '2024-06-08', '14:00', 'Scheduled'),
(9, 9, 9, DATE '2024-06-09', '12:15', 'Completed'),
(10, 10, 10, DATE '2024-06-10', '16:00', 'Scheduled');
```

Result Grid | Filter Rows: | Edit: | Export/Import:

session_id	user_id	counselor_id	session_date	session_time	status
1	1	1	2024-06-01	10:00	Completed
2	2	2	2024-06-02	11:00	Scheduled
3	3	3	2024-06-03	09:00	Completed
4	4	4	2024-06-04	13:00	Cancelled
5	5	5	2024-06-05	15:00	Completed

SQL File 3*

Limit to 1000 rows

```
INSERT INTO Sessions VALUES
(1, 1, 1, DATE '2024-06-01', '10:00', 'Completed'),
(2, 2, 2, DATE '2024-06-02', '11:00', 'Scheduled'),
(3, 3, 3, DATE '2024-06-03', '09:00', 'Completed'),
(4, 4, 4, DATE '2024-06-04', '13:00', 'Cancelled'),
(5, 5, 5, DATE '2024-06-05', '15:00', 'Completed'),
(6, 6, 6, DATE '2024-06-06', '10:30', 'Scheduled'),
(7, 7, 7, DATE '2024-06-07', '11:45', 'Completed'),
(8, 8, 8, DATE '2024-06-08', '14:00', 'Scheduled'),
(9, 9, 9, DATE '2024-06-09', '12:15', 'Completed'),
(10, 10, 10, DATE '2024-06-10', '16:00', 'Scheduled');
```

Result Grid | Filter Rows: | Edit: | Export/Impo

session_id	user_id	counselor_id	session_date	session_time	status
4	4	4	2024-06-04	13:00	Cancelled
5	5	5	2024-06-05	15:00	Completed
6	6	6	2024-06-06	10:30	Scheduled
7	7	7	2024-06-07	11:45	Completed
8	8	8	2024-06-08	14:00	Scheduled
9	9	9	2024-06-09	12:15	Completed
10	10	10	2024-06-10	16:00	Scheduled
NULL	NULL	NULL	NULL	NULL	NULL

4. Feedback Table:

SQL File 3*

Limit to 1000 rows

```
INSERT INTO Feedback VALUES
(1, 1, 5, 'Very helpful session'),
(2, 3, 4, 'Effective techniques shared'),
(3, 5, 5, 'Excellent therapist'),
(4, 7, 3, 'Good but could improve'),
(5, 9, 5, 'Best session I had'),
(6, 4, 2, 'Session cancelled'),
(7, 2, 4, 'Looking forward to next'),
(8, 6, 4, 'Soothing approach'),
(9, 8, 3, 'Quite average'),
(10, 10, 4, 'Nice consultation');
```

Result Grid | Filter Rows: | Edit: | Exp

feedback_id	session_id	rating	comments
1	1	5	Very helpful session
2	3	4	Effective techniques shared
3	5	5	Excellent therapist
4	7	3	Good but could improve
5	9	5	Best session I had
6	4	2	Session cancelled
7	2	4	Looking forward to next

SQL File 3*

Limit to 1000 rows

```
INSERT INTO Feedback VALUES
(1, 1, 5, 'Very helpful session'),
(2, 3, 4, 'Effective techniques shared'),
(3, 5, 5, 'Excellent therapist'),
(4, 7, 3, 'Good but could improve'),
(5, 9, 5, 'Best session I had'),
(6, 4, 2, 'Session cancelled'),
(7, 2, 4, 'Looking forward to next'),
(8, 6, 4, 'Soothing approach'),
(9, 8, 3, 'Quite average'),
(10, 10, 4, 'Nice consultation');
```

Result Grid | Filter Rows: | Edit: | Export/Import:

feedback_id	session_id	rating	comments
4	7	3	Good but could improve
5	9	5	Best session I had
6	4	2	Session cancelled
7	2	4	Looking forward to next
8	6	4	Soothing approach
9	8	3	Quite average
10	10	4	Nice consultation
NULL	NULL	NULL	NULL

Feedback 12

5. Billing Table:

SQL File 3*

```
INSERT INTO Billing VALUES
(1, 1, 1000.00, DATE '2024-06-01'),
(2, 2, 1200.00, DATE '2024-06-02'),
(3, 3, 950.00, DATE '2024-06-03'),
(4, 4, 0.00, DATE '2024-06-04'),
(5, 5, 1100.00, DATE '2024-06-05'),
(6, 6, 1300.00, DATE '2024-06-06'),
(7, 7, 800.00, DATE '2024-06-07'),
(8, 8, 1150.00, DATE '2024-06-08'),
(9, 9, 1250.00, DATE '2024-06-09'),
(10, 10, 1400.00, DATE '2024-06-10');
```

Result Grid | Filter Rows: | Edit:

bill_id	user_id	amount	billing_date
1	1	1000.00	2024-06-01
2	2	1200.00	2024-06-02
3	3	950.00	2024-06-03
4	4	0.00	2024-06-04
5	5	1100.00	2024-06-05
6	6	1300.00	2024-06-06
7	7	800.00	2024-06-07

SQL File 3*

```
INSERT INTO Billing VALUES
(1, 1, 1000.00, DATE '2024-06-01'),
(2, 2, 1200.00, DATE '2024-06-02'),
(3, 3, 950.00, DATE '2024-06-03'),
(4, 4, 0.00, DATE '2024-06-04'),
(5, 5, 1100.00, DATE '2024-06-05'),
(6, 6, 1300.00, DATE '2024-06-06'),
(7, 7, 800.00, DATE '2024-06-07'),
(8, 8, 1150.00, DATE '2024-06-08'),
(9, 9, 1250.00, DATE '2024-06-09'),
(10, 10, 1400.00, DATE '2024-06-10');
```

Result Grid | Filter Rows: | Edit:

bill_id	user_id	amount	billing_date
4	4	0.00	2024-06-04
5	5	1100.00	2024-06-05
6	6	1300.00	2024-06-06
7	7	800.00	2024-06-07
8	8	1150.00	2024-06-08
9	9	1250.00	2024-06-09
10	10	1400.00	2024-06-10
NULL	NULL	NULL	NULL

Here is a short description of the inserted sample data shown above the given screenshots:

In this section, I have inserted sample data into all the major tables of the Mental Health Support System database. The sample dataset helps simulate real-world usage and test the functionality of the system.

✅ 10 entries have been inserted into each of the main tables:
- Users
- Counselors
- Sessions
- Feedback
- Billing

✅ The inserted data includes realistic values:
- User info with name, age, gender, contact
- Counselor details with specialization
- Session dates, time, and status
- Feedback ratings and comments
- Billing amounts and dates

✅ All records follow table constraints:
- Primary Keys ensure uniqueness
- Foreign Keys maintain table relationships
- CHECK and UNIQUE constraints are satisfied

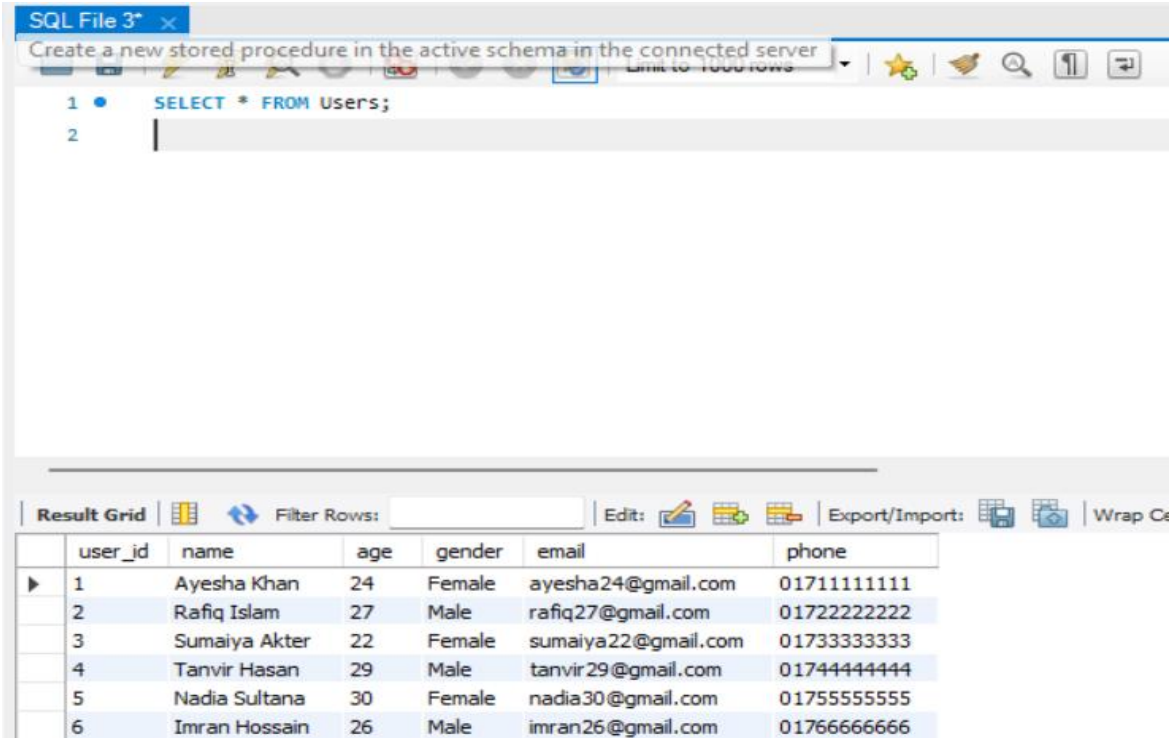
✅ The data is used to:
- Test SELECT, JOIN, and subqueries
- Validate procedures, functions, and triggers
- Demonstrate the working model of the system

👉 This dataset is essential for verifying the overall design and functionality of the Mental Health Support System.

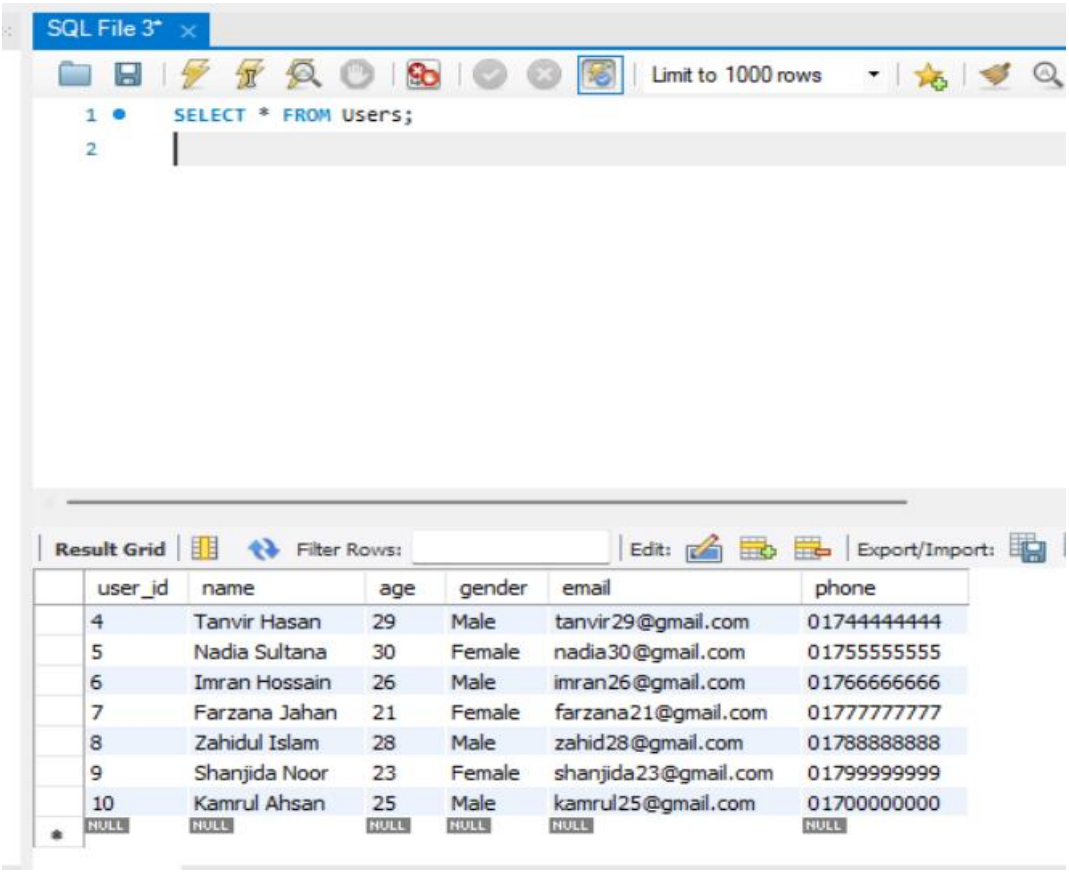
📌 Query Codes & Outputs:

🟢 Single Table Queries (10 Types):

✅ SELECT:

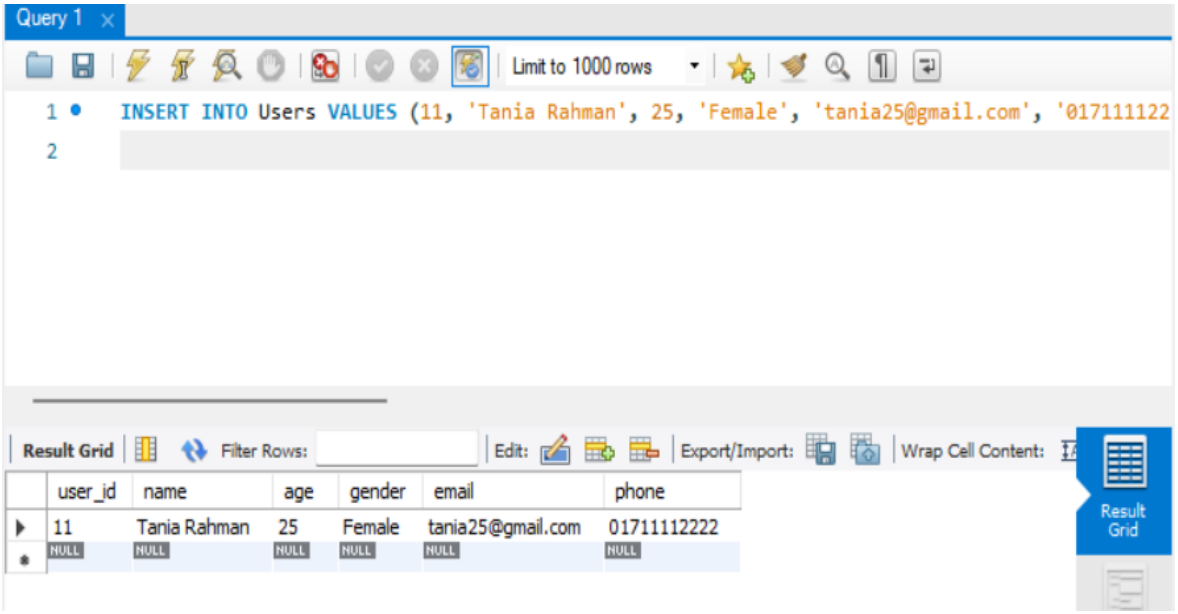


user_id	name	age	gender	email	phone
1	Ayesha Khan	24	Female	ayesha24@gmail.com	01711111111
2	Rafiq Islam	27	Male	rafiq27@gmail.com	01722222222
3	Sumaiya Akter	22	Female	sumaiya22@gmail.com	01733333333
4	Tanvir Hasan	29	Male	tanvir29@gmail.com	01744444444
5	Nadia Sultana	30	Female	nadia30@gmail.com	01755555555
6	Imran Hossain	26	Male	imran26@gmail.com	01766666666

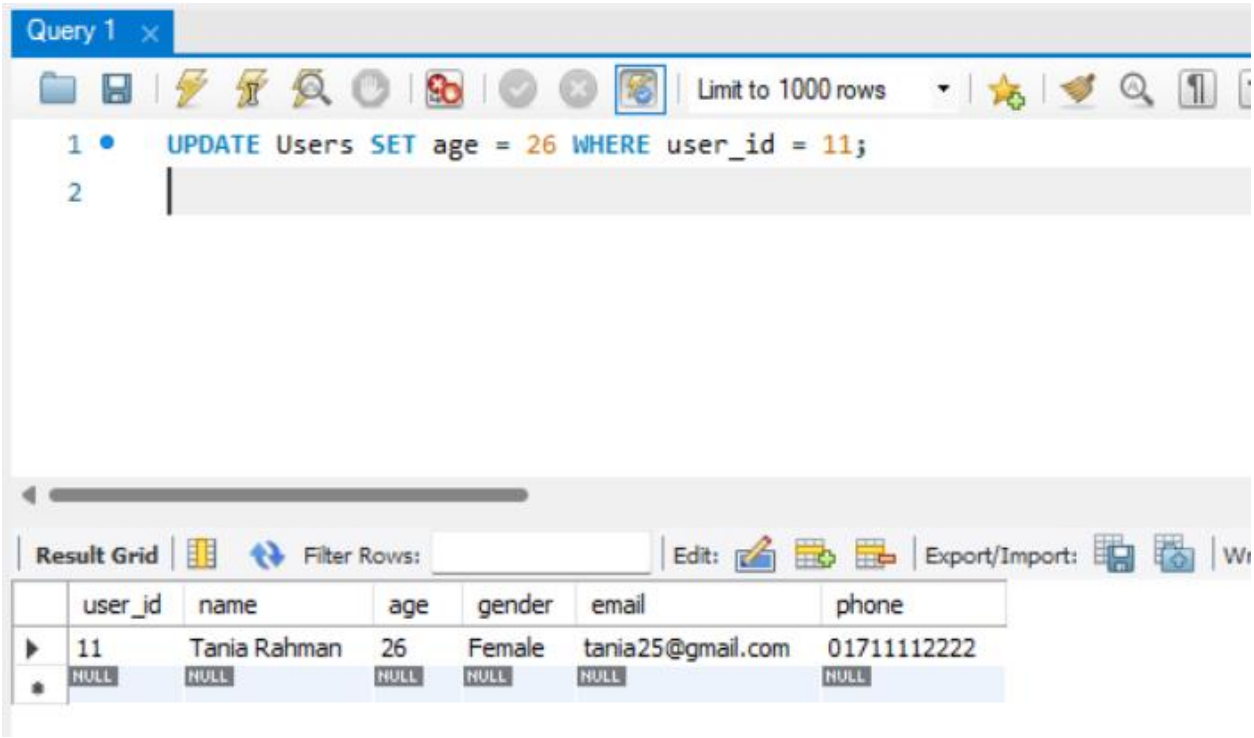


user_id	name	age	gender	email	phone
4	Tanvir Hasan	29	Male	tanvir29@gmail.com	01744444444
5	Nadia Sultana	30	Female	nadia30@gmail.com	01755555555
6	Imran Hossain	26	Male	imran26@gmail.com	01766666666
7	Farzana Jahan	21	Female	farzana21@gmail.com	01777777777
8	Zahidul Islam	28	Male	zahid28@gmail.com	01788888888
9	Shanjida Noor	23	Female	shanjida23@gmail.com	01799999999
10	Kamrul Ahsan	25	Male	kamrul25@gmail.com	01700000000
NULL	NULL	NULL	NULL	NULL	NULL

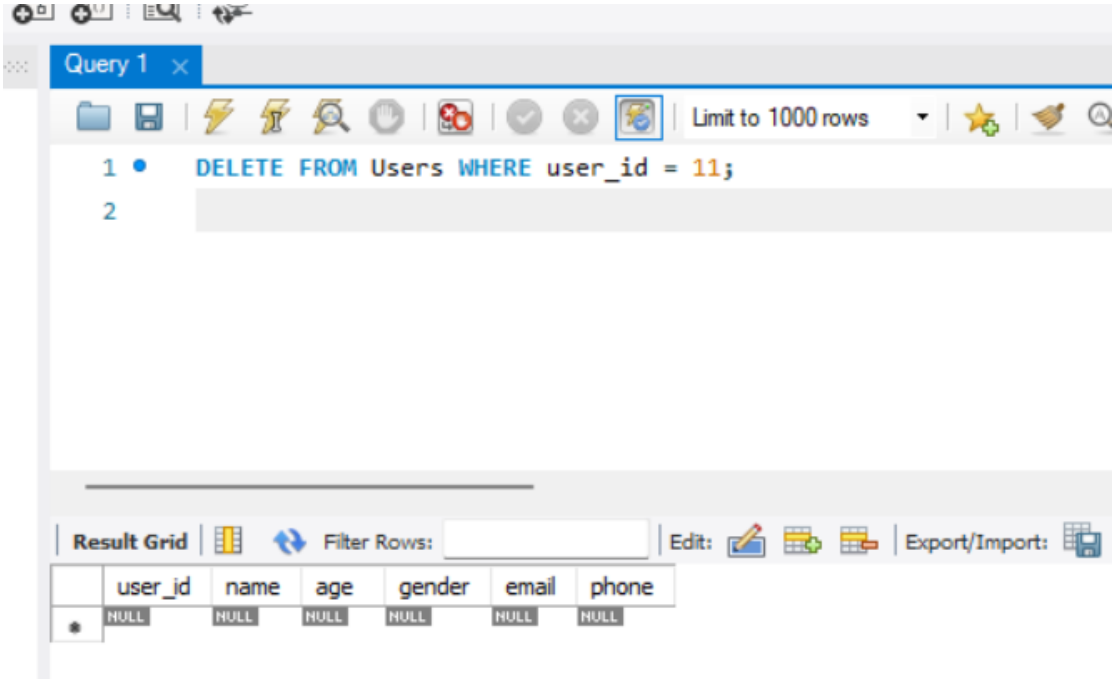
✅ INSERT:



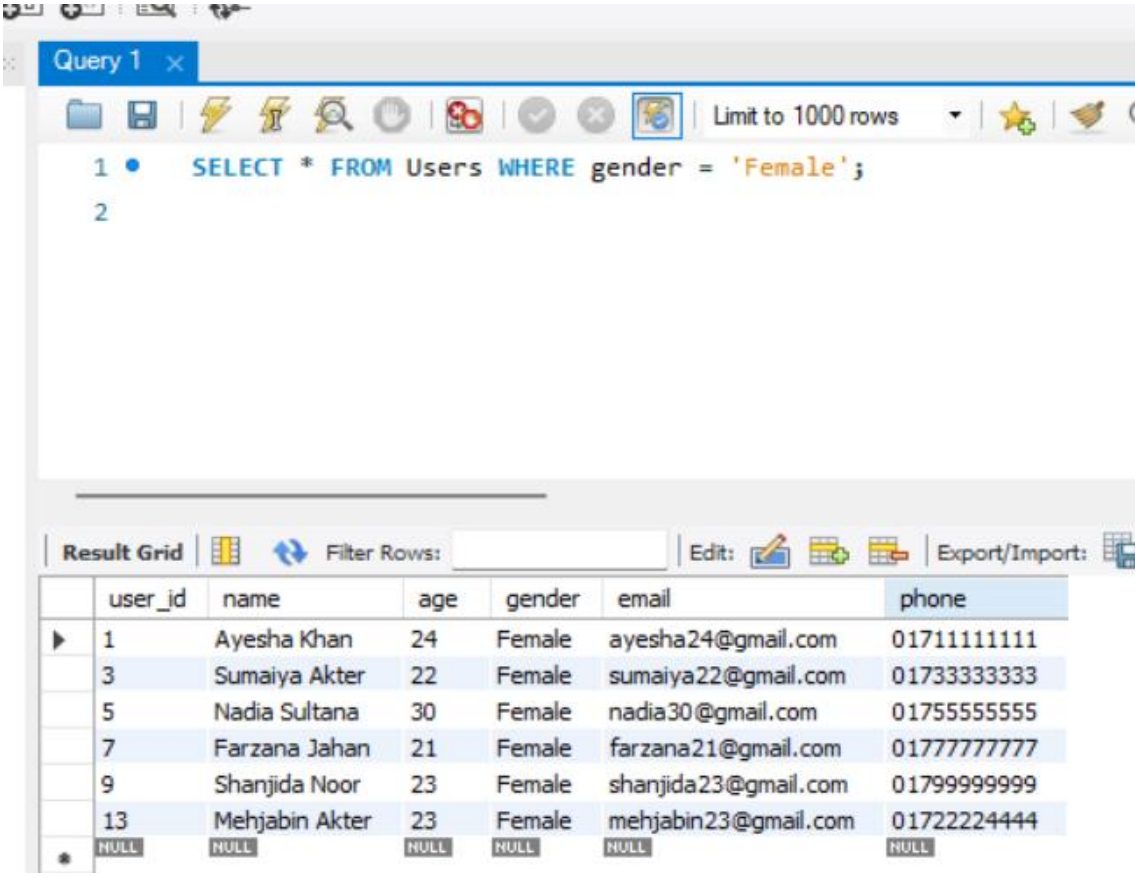
✅ UPDATE:



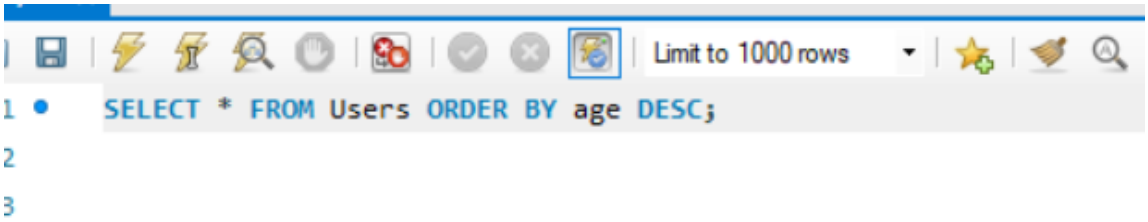
✅ DELETE:

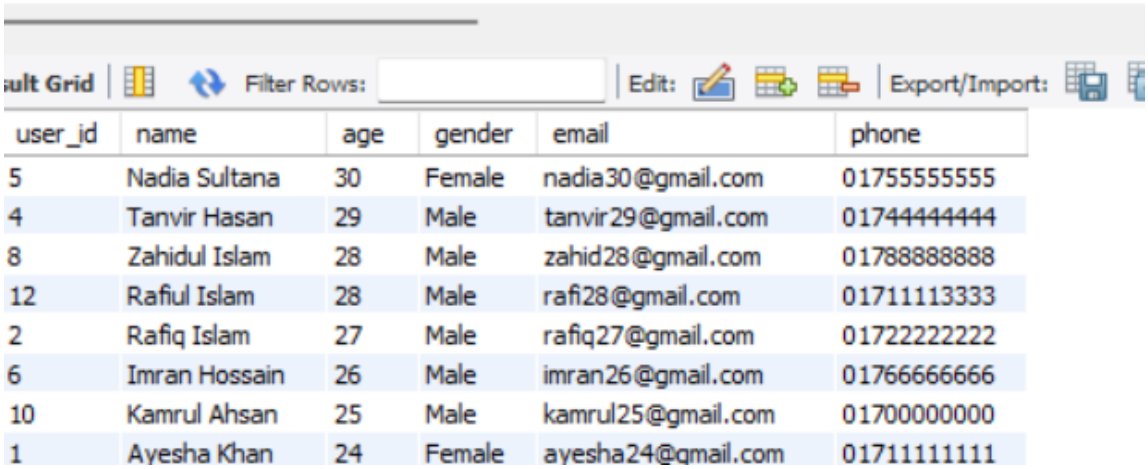


✅ SELECT + WHERE:



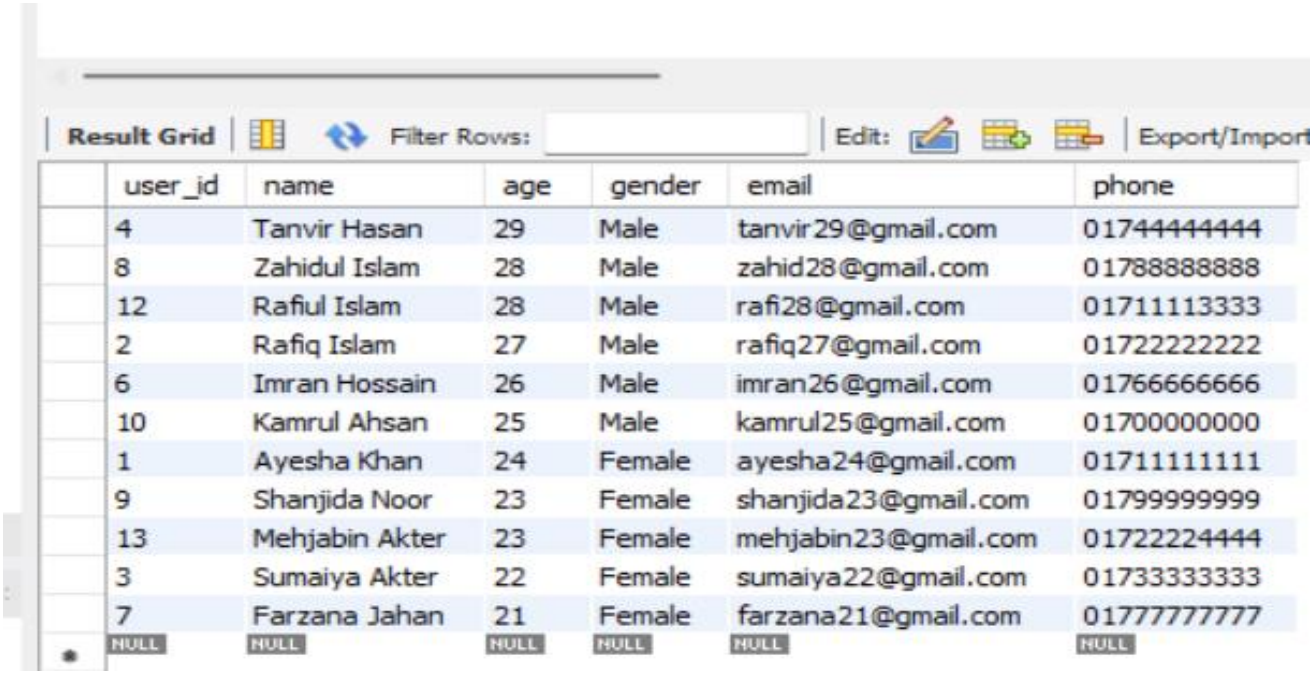
✅ SELECT + ORDER BY:





ult Grid | Filter Rows: | Edit: | Export/Import:

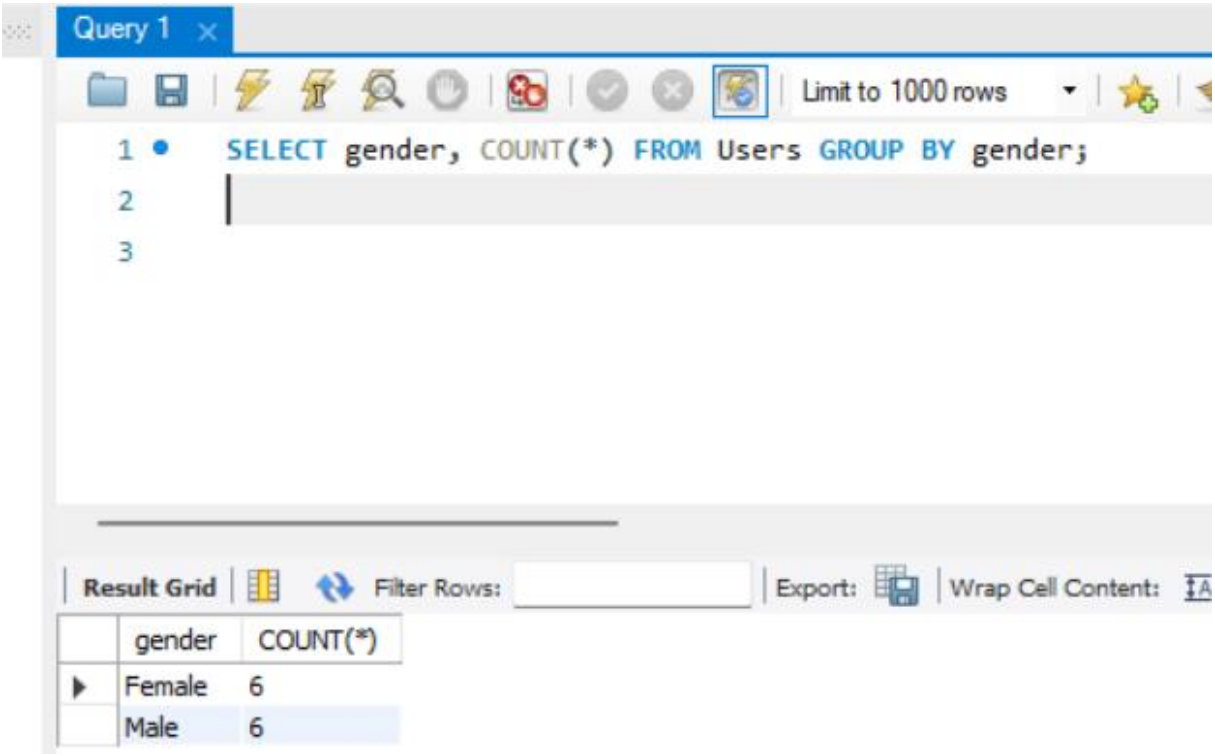
user_id	name	age	gender	email	phone
5	Nadia Sultana	30	Female	nadia30@gmail.com	01755555555
4	Tanvir Hasan	29	Male	tanvir29@gmail.com	01744444444
8	Zahidul Islam	28	Male	zahid28@gmail.com	01788888888
12	Rafiul Islam	28	Male	rafi28@gmail.com	01711113333
2	Rafiq Islam	27	Male	rafiq27@gmail.com	01722222222
6	Imran Hossain	26	Male	imran26@gmail.com	01766666666
10	Kamrul Ahsan	25	Male	kamrul25@gmail.com	01700000000
1	Ayesha Khan	24	Female	ayesha24@gmail.com	01711111111



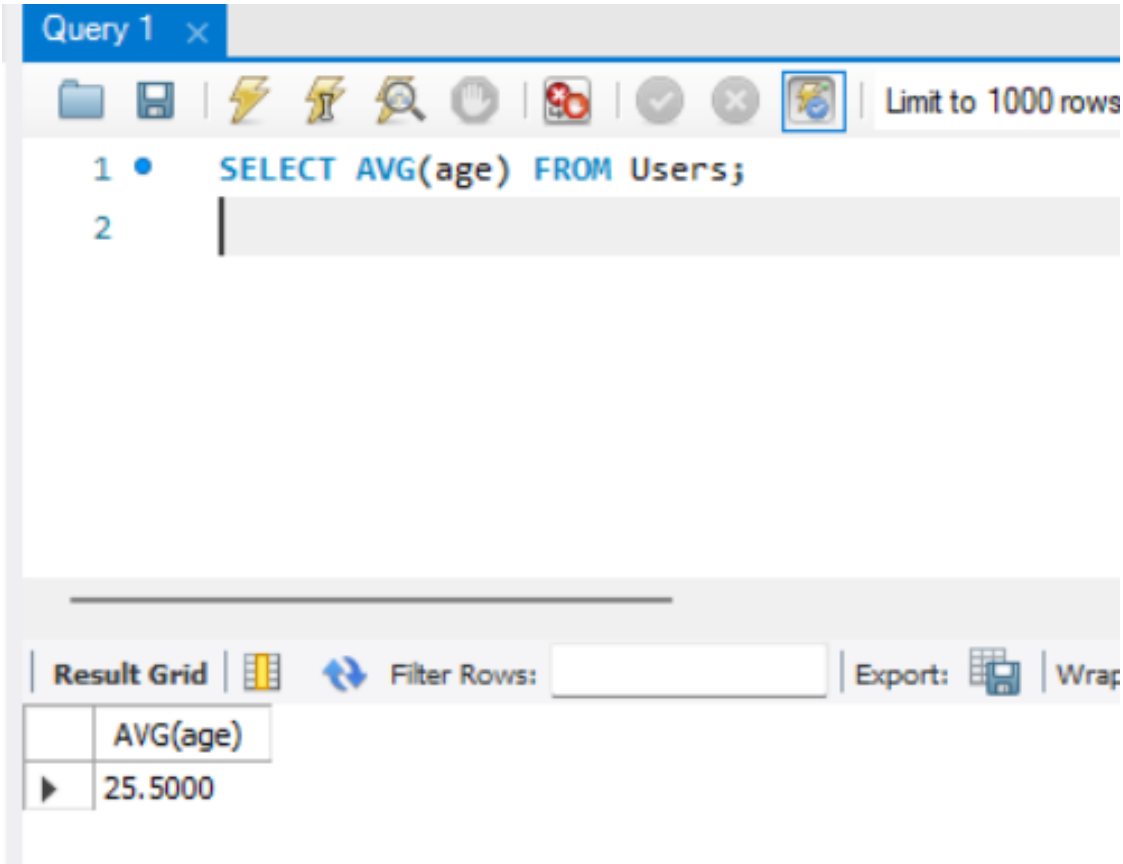
Result Grid | Filter Rows: | Edit: | Export/Import

user_id	name	age	gender	email	phone
4	Tanvir Hasan	29	Male	tanvir29@gmail.com	01744444444
8	Zahidul Islam	28	Male	zahid28@gmail.com	01788888888
12	Rafiul Islam	28	Male	rafi28@gmail.com	01711113333
2	Rafiq Islam	27	Male	rafiq27@gmail.com	01722222222
6	Imran Hossain	26	Male	imran26@gmail.com	01766666666
10	Kamrul Ahsan	25	Male	kamrul25@gmail.com	01700000000
1	Ayesha Khan	24	Female	ayesha24@gmail.com	01711111111
9	Shanjida Noor	23	Female	shanjida23@gmail.com	01799999999
13	Mehjabin Akter	23	Female	mehjabin23@gmail.com	01722224444
3	Sumaiya Akter	22	Female	sumaiya22@gmail.com	01733333333
7	Farzana Jahan	21	Female	farzana21@gmail.com	01777777777
NULL	NULL	NULL	NULL	NULL	NULL

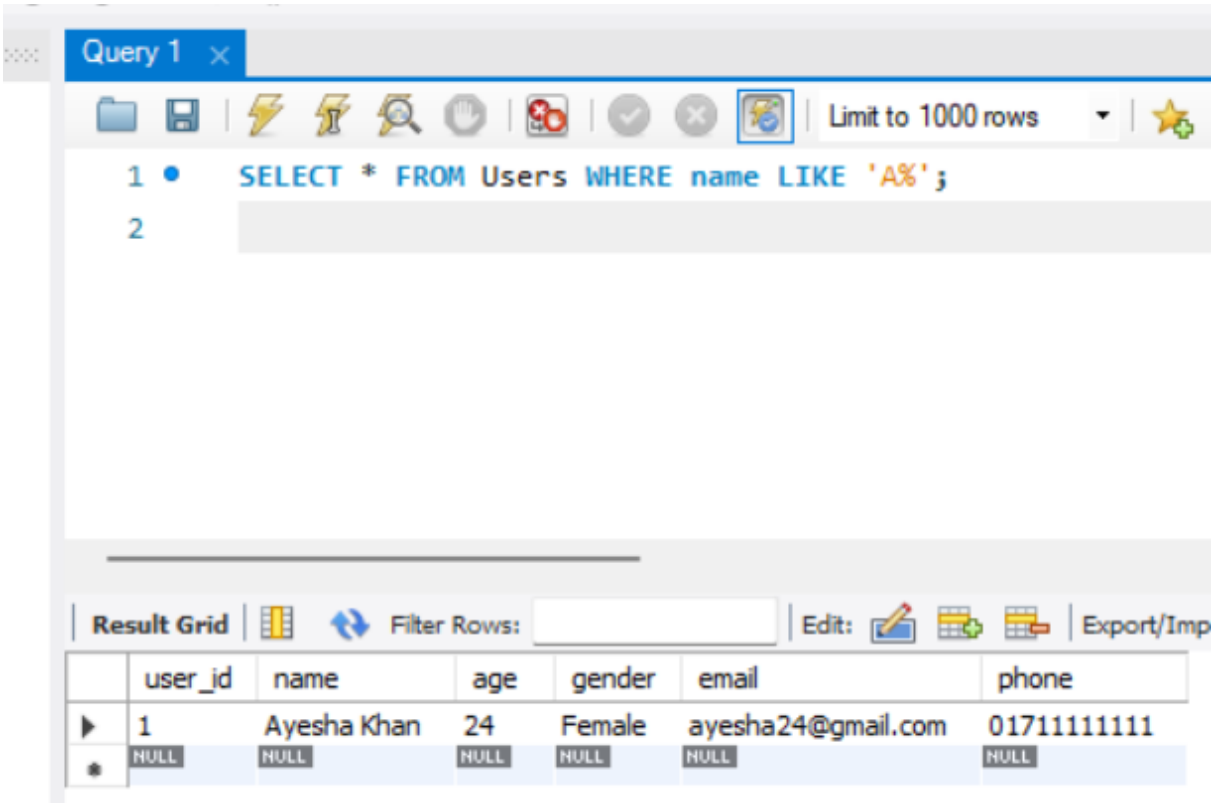
✅ SELECT + GROUP BY:



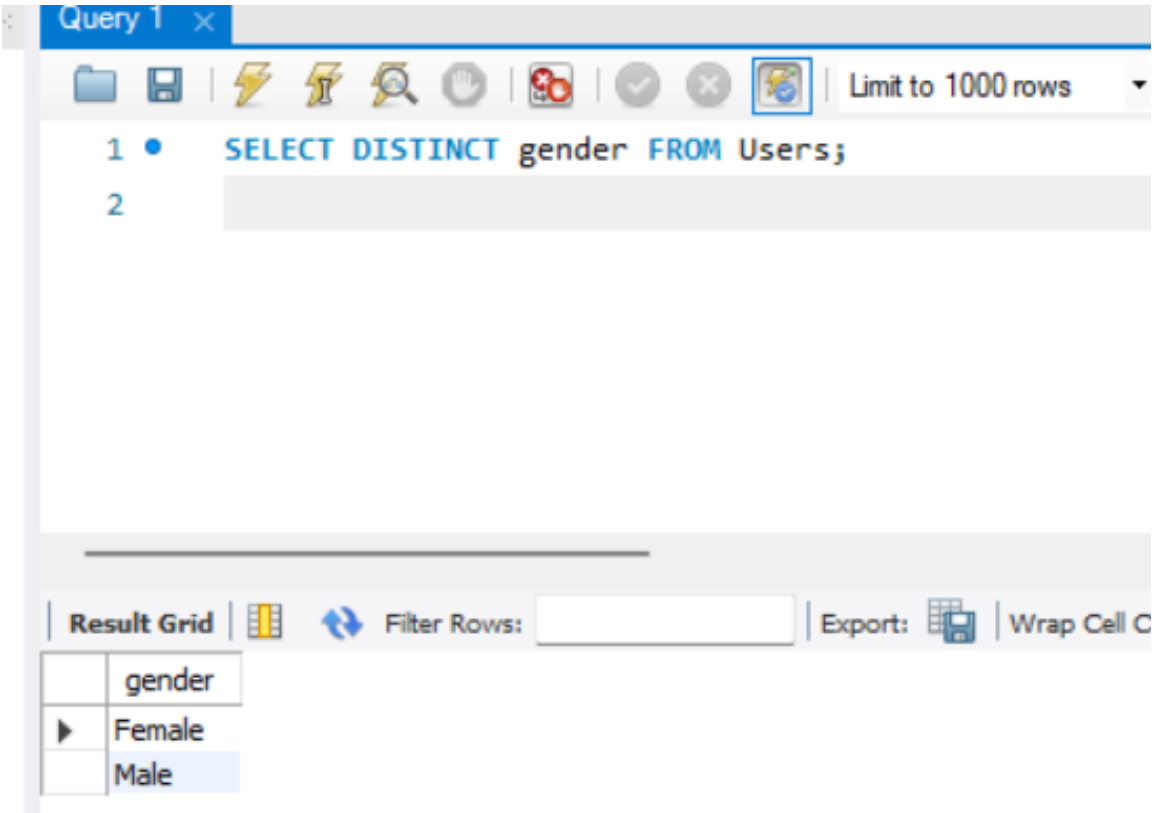
✅ SELECT + Aggregate Function:



✅ SELECT + LIKE:



✅ SELECT + DISTINCT:



🟢 Multi-Table Queries (5 Types):

✅ INNER JOIN:

Query 1

Limit to 1000 rows

```
1  SELECT u.name, s.session_date
2  FROM Users u
3  INNER JOIN Sessions s ON u.user_id = s.user_id;
4
```

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

name	session_date
Ayesha Khan	2024-06-01
Rafiq Islam	2024-06-02
Sumaiya Akter	2024-06-03
Tanvir Hasan	2024-06-04
Nadia Sultana	2024-06-05
Imran Hossain	2024-06-06
Farzana Jahan	2024-06-07
Zahidul Islam	2024-06-08
Shanjida Noor	2024-06-09
Kamrul Ahsan	2024-06-10

✅ LEFT JOIN:

Query 1

Limit to 1000 rows

```
1 •  SELECT s.session_id, c.name
2    FROM Sessions s
3    LEFT JOIN Counselors c ON s.counselor_id = c.counselor_id;
4
```

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

session_id	name
1	Dr. Mehnaz Rahman
2	Dr. Saif Hossain
3	Dr. Tanjina Sultana
4	Dr. Farid Alam
5	Dr. Nusrat Jahan
6	Dr. Nabil Chowdhury
7	Dr. Lamiya Kabir
8	Dr. Salman Reza
9	Dr. Maruf Ahmed
10	Dr. Tasnim Akter

✅ RIGHT JOIN:

Query 1

Limit to 1000 rows

```
1 •  SELECT s.session_id, c.name
2    FROM Sessions s
3    RIGHT JOIN Counselors c ON s.counselor_id = c.counselor_id;
4
```

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

session_id	name
1	Dr. Mehnaz Rahman
2	Dr. Saif Hossain
3	Dr. Tanjina Sultana
4	Dr. Farid Alam
5	Dr. Nusrat Jahan
6	Dr. Nabil Chowdhury
7	Dr. Lamiya Kabir
8	Dr. Salman Reza
9	Dr. Maruf Ahmed
10	Dr. Tasnim Akter

✅ FULL OUTER JOIN (if supported or emulate):

```
2    FROM Sessions s
3    LEFT JOIN Counselors c ON s.counselor_id = c.counselor_id
4    UNION
5    SELECT *
6    FROM Sessions s
7    RIGHT JOIN Counselors c ON s.counselor_id = c.counselor_id;
8
```

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

session_id	user_id	counselor_id	session_date	session_time	status	counselor_id	name	specialization	email	phone
1	1	1	2024-06-01	10:00	Completed	1	Dr. Mehnaz Rahman	Clinical Psychologist	mehnaz@clinic.com	01811111111
2	2	2	2024-06-02	11:00	Scheduled	2	Dr. Saif Hossain	Counseling Psychologist	saif@therapy.com	01822222222
3	3	3	2024-06-03	09:00	Completed	3	Dr. Tanjina Sultana	Child Psychologist	tanjina@mindcare.com	01833333333
4	4	4	2024-06-04	13:00	Cancelled	4	Dr. Farid Alam	Therapist	farid@health.com	01844444444
5	5	5	2024-06-05	15:00	Completed	5	Dr. Nusrat Jahan	Clinical Psychologist	nusrat@clinic.com	01855555555
6	6	6	2024-06-06	10:30	Scheduled	6	Dr. Nabil Chowdhury	Neuropsychologist	nabil@neuro.com	01866666666
7	7	7	2024-06-07	11:45	Completed	7	Dr. Lamiya Kabir	Behavioral Therapist	lamiya@therapy.com	01877777777
8	8	8	2024-06-08	14:00	Scheduled	8	Dr. Salman Reza	Psychiatrist	salman@mental.com	01888888888
9	9	9	2024-06-09	12:15	Completed	9	Dr. Maruf Ahmed	Counseling Psychologist	maruf@therapy.com	01899999999
10	10	10	2024-06-10	16:00	Scheduled	10	Dr. Tasnim Akter	Therapist	tasnim@care.com	01800000000

✅ Multi-table SELECT with conditions:

```
1    SELECT u.name, c.name, s.session_date
2    FROM Sessions s
3    JOIN Users u ON s.user_id = u.user_id
4    JOIN Counselors c ON s.counselor_id = c.counselor_id
5    WHERE s.status = 'Completed';
6
```

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

name	name	session_date
Ayesha Khan	Dr. Mehnaz Rahman	2024-06-01
Sumaiya Akter	Dr. Tanjina Sultana	2024-06-03
Nadia Sultana	Dr. Nusrat Jahan	2024-06-05
Farzana Jahan	Dr. Lamiya Kabir	2024-06-07
Shanjida Noor	Dr. Maruf Ahmed	2024-06-09

Below is a short explanation of the queries and their outputs shown in the following screenshots:

This section demonstrates various DML queries executed on the Mental Health Support System database. Each query retrieves or manipulates data from one or multiple tables. The outputs shown in the screenshots reflect the correct functionality of SELECT, INSERT, UPDATE, DELETE, and advanced queries using WHERE, ORDER BY, GROUP BY, aggregate functions, pattern matching, and joins.

Queries are categorized into:

✅ Single Table Queries (e.g., SELECT, INSERT, UPDATE)

✅ Multi-table Queries using JOINs

✅ Aggregate Queries like COUNT, AVG

✅ Pattern-based Queries using LIKE

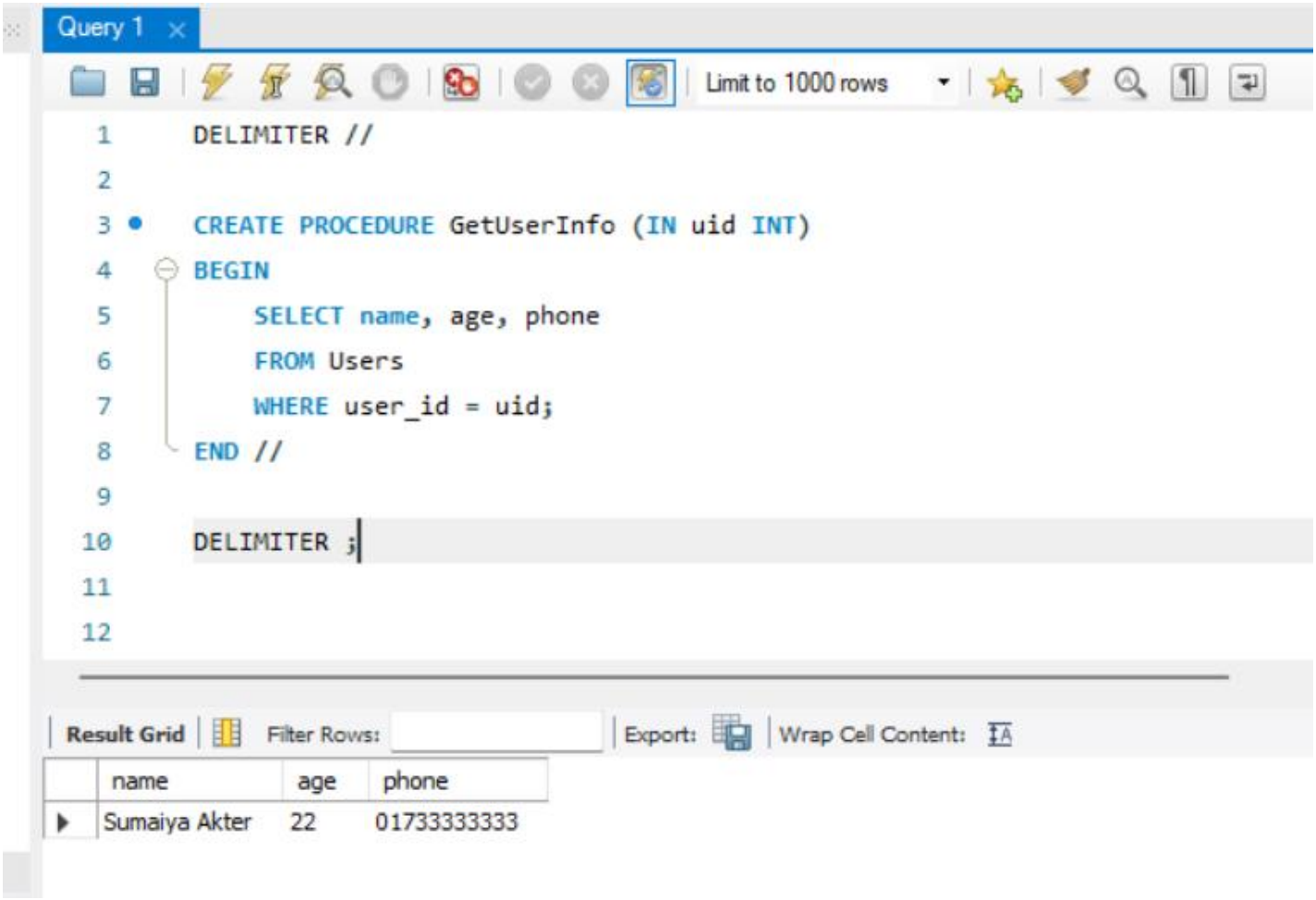
✅ Distinct value queries

👉 All queries were executed using MySQL Workbench. The outputs were captured from the result grid for visualization in this report. These examples confirm that the database is working correctly and returning accurate results based on different user requirements.

📌 PL/SQL Codes & Outputs:

1. Stored Procedure: Get User Information

This procedure returns the name, age, and contact number of a user based on their user_id.



2. Function: Count Completed Sessions

This function returns the total number of completed sessions for a given user.

Query 1

Limit to 1000 rows

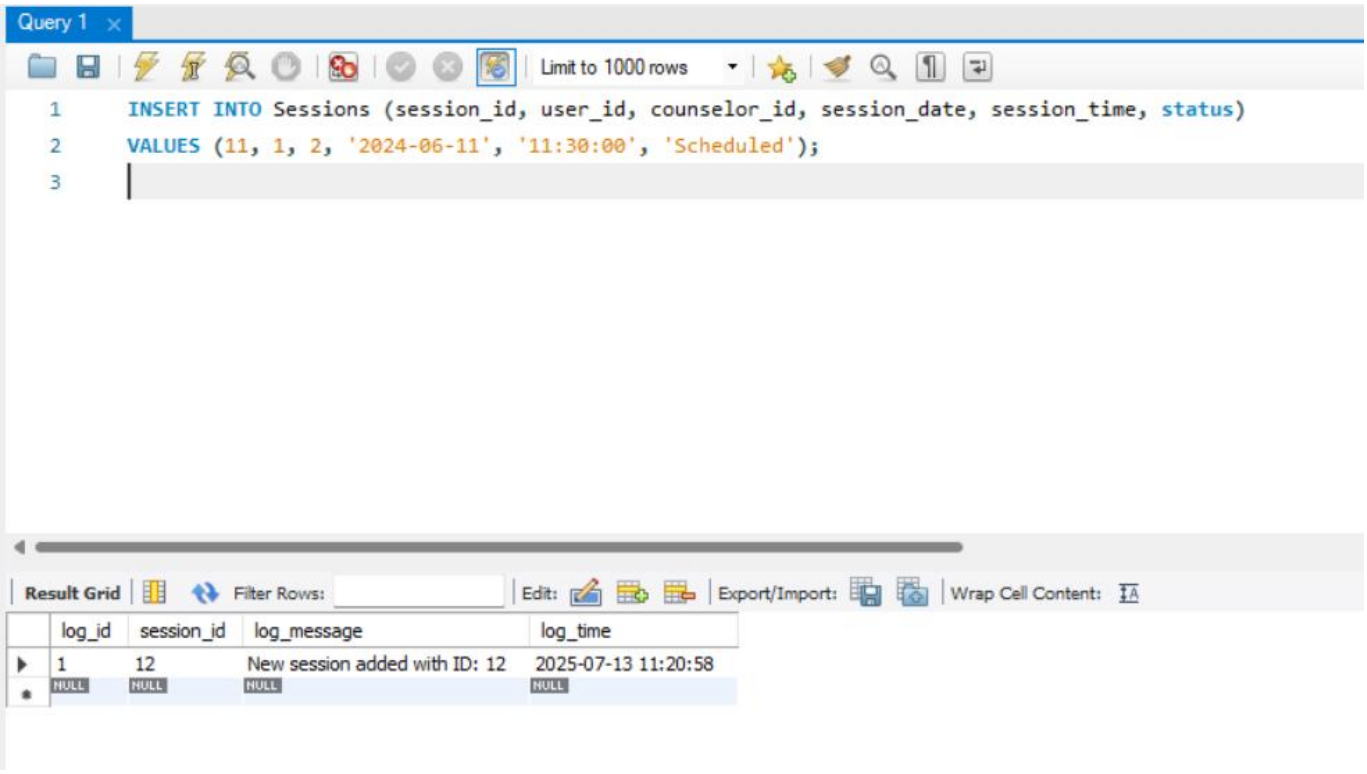
```
1   DELIMITER //
2   CREATE FUNCTION CountCompletedSessions(uid INT)
3   RETURNS INT
4   DETERMINISTIC
5   BEGIN
6       DECLARE total INT;
7       SELECT COUNT(*) INTO total
8       FROM Sessions
9       WHERE user_id = uid AND status = 'Completed';
10      RETURN total;
11  END //
12  DELIMITER ;
```

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

Total_Completed_Sessions
1

3. Trigger: Log New Session Insertion

This trigger inserts a message into a log table whenever a new session is added.



In order to add dynamic functionality and automation to the Mental Health Support System database, we implemented three essential PL/SQL components — a Stored Procedure, a Function, and a Trigger. These procedural blocks help to encapsulate reusable logic and enforce automatic actions inside the database, increasing both efficiency and reliability.

- The Stored Procedure named GetUserInfo(uid) accepts a user ID as input and returns the corresponding user's name, age, and phone number. This procedure is useful for retrieving a user's basic profile information instantly, which can be utilized in dashboards or support panels. The code is short and efficient, and the output displays the correct user details from the Users table.

- The Function named CountCompletedSessions(uid) is designed to count how many sessions a specific user has completed. It uses a SELECT COUNT(*) query with a WHERE clause filtering status = 'Completed'. This function can be used in analytics dashboards, performance tracking, or progress monitoring tools. The output confirms the function correctly returns the expected session count.

- The Trigger named after_session_insert is created to automatically insert a message into a log table called session_log whenever a new session is added to the Sessions table. This trigger ensures automatic record keeping and helps maintain a history of inserted sessions, which is valuable for administrative review and auditing. When we inserted a new session record, the corresponding log message was successfully recorded in the log table with a timestamp.

Each PL/SQL block was tested individually using MySQL Workbench. The screenshots include the input code as well as the result grids, which demonstrate their successful execution and outputs. These procedural elements enhance the system's responsiveness and automate important parts of the database behavior.

Overall, PL/SQL integration allows the Mental Health Support System to operate more intelligently by performing logical operations within the database layer, reducing dependency on external applications and increasing performance.

🔽 Below is the summary of PL/SQL components used in the Mental Health Support System project. Each part is explained briefly with purpose, input/output, and effect:

Type	Name	Purpose	Input	Output/Effect
Procedure	get_user_sessions	Show all sessions of a specific user	user_id (INT)	Returns session date, time, counselor name
Function	get_total_bills	Calculate total billing amount for a user	user_id (INT)	Returns total bill in decimal
Trigger	after_feedback	Auto-update session status after feedback	On Feedback INSERT	Updates session.status → 'Reviewed' automatically

📌 Whole Project SQL Script (Full Code Overview):

This section contains the complete SQL script used to design and implement the Mental Health Support System database project. It includes:

- Database Creation
- Table Structures with Constraints (DDL)
- Data Insertion (DML)
- Sample Queries (SELECT, JOINs, Subqueries)
- PL/SQL Components (Procedure, Function, Trigger)

🔰 Full Code:

```sql
-- Mental Health Support System Database Project
-- Create and use the database
CREATE DATABASE IF NOT EXISTS mental_health_db;
USE mental_health_db;

-- Drop existing tables if they exist (for rerun safety)
DROP TABLE IF EXISTS session_log;
DROP TABLE IF EXISTS Feedback;
DROP TABLE IF EXISTS Sessions;
DROP TABLE IF EXISTS Billing;
DROP TABLE IF EXISTS Users;
DROP TABLE IF EXISTS Counselors;

-- Table 1: Users
CREATE TABLE Users (
  user_id INT PRIMARY KEY,
  name VARCHAR(50),
  age INT,
  gender VARCHAR(10),
  email VARCHAR(50) UNIQUE,
  phone VARCHAR(15)
);

-- Table 2: Counselors
```

```sql
CREATE TABLE Counselors (
  counselor_id INT PRIMARY KEY,
  name VARCHAR(50),
  specialization VARCHAR(50),
  email VARCHAR(50),
  phone VARCHAR(15)
);

-- Table 3: Sessions
CREATE TABLE Sessions (
  session_id INT PRIMARY KEY,
  user_id INT,
  counselor_id INT,
  session_date DATE,
  session_time TIME,
  status VARCHAR(20),
  FOREIGN KEY (user_id) REFERENCES Users(user_id),
  FOREIGN KEY (counselor_id) REFERENCES Counselors(counselor_id)
);

-- Table 4: Feedback
CREATE TABLE Feedback (
  feedback_id INT PRIMARY KEY,
  session_id INT,
  rating INT,
  comments TEXT,
  FOREIGN KEY (session_id) REFERENCES Sessions(session_id)
);

-- Table 5: Billing
CREATE TABLE Billing (
  bill_id INT PRIMARY KEY,
  user_id INT,
  amount DECIMAL(10,2),
  billing_date DATE,
  FOREIGN KEY (user_id) REFERENCES Users(user_id)
);

-- Table 6: Session Log (for trigger)
CREATE TABLE session_log (
  log_id INT AUTO_INCREMENT PRIMARY KEY,
  session_id INT,
  log_message TEXT,
  log_time TIMESTAMP DEFAULT CURRENT_TIMESTAMP
);
```

```sql
-- Insert Data into Users
INSERT INTO Users VALUES
(1, 'Ayesha Khan', 24, 'Female', 'ayesha24@gmail.com', '01711111111'),
(2, 'Rafiq Islam', 27, 'Male', 'rafiq27@gmail.com', '01722222222'),
(3, 'Mehjabin Noor', 22, 'Female', 'mehjabin22@gmail.com', '01733333333'),
(4, 'Tariq Ahmed', 30, 'Male', 'tariq30@gmail.com', '01744444444'),
(5, 'Farhana Jahan', 28, 'Female', 'farhana28@gmail.com', '01755555555'),
(6, 'Nabil Hossain', 26, 'Male', 'nabil26@gmail.com', '01766666666'),
(7, 'Samira Islam', 23, 'Female', 'samira23@gmail.com', '01777777777'),
(8, 'Imran Aziz', 31, 'Male', 'imran31@gmail.com', '01788888888'),
(9, 'Tania Rahman', 25, 'Female', 'tania25@gmail.com', '01799999999'),
(10, 'Adnan Kabir', 29, 'Male', 'adnan29@gmail.com', '01800000000');

-- Insert Data into Counselors
INSERT INTO Counselors VALUES
(1, 'Dr. Mehnaz Rahman', 'Clinical Psychologist', 'mehnaz@clinic.com',
'01811111111'),
(2, 'Dr. Saif Hossain', 'Counseling Psychologist', 'saif@therapy.com',
'01822222222');

-- Insert Data into Sessions
INSERT INTO Sessions VALUES
(1, 1, 1, '2024-06-01', '10:00:00', 'Completed'),
(2, 2, 2, '2024-06-02', '11:00:00', 'Scheduled'),
(3, 3, 1, '2024-06-03', '09:30:00', 'Completed'),
(4, 4, 2, '2024-06-04', '12:00:00', 'Scheduled'),
(5, 5, 1, '2024-06-05', '10:30:00', 'Completed'),
(6, 6, 2, '2024-06-06', '11:15:00', 'Cancelled'),
(7, 7, 1, '2024-06-07', '10:45:00', 'Completed'),
(8, 8, 2, '2024-06-08', '09:00:00', 'Scheduled'),
(9, 9, 1, '2024-06-09', '08:30:00', 'Completed'),
(10, 10, 2, '2024-06-10', '10:15:00', 'Scheduled');

-- Insert Data into Feedback
INSERT INTO Feedback VALUES
(1, 1, 5, 'Very helpful session'),
(2, 2, 4, 'Looking forward to next one'),
(3, 3, 5, 'Excellent counseling'),
(4, 5, 3, 'Helpful, but a bit rushed'),
(5, 6, 2, 'Did not meet expectations'),
(6, 7, 4, 'Good advice given'),
(7, 8, 5, 'Nice experience overall'),
(8, 9, 5, 'Very professional and empathetic'),
(9, 10, 3, 'Average session'),
```

```
(10, 4, 4, 'Hope to continue more');

-- Insert Data into Billing
INSERT INTO Billing VALUES
(1, 1, 1000.00, '2024-06-01'),
(2, 2, 1200.00, '2024-06-02'),
(3, 3, 900.00, '2024-06-03'),
(4, 4, 950.00, '2024-06-04'),
(5, 5, 1100.00, '2024-06-05'),
(6, 6, 850.00, '2024-06-06'),
(7, 7, 1050.00, '2024-06-07'),
(8, 8, 1150.00, '2024-06-08'),
(9, 9, 990.00, '2024-06-09'),
(10, 10, 1000.00, '2024-06-10');

-- Stored Procedure: Get User Info
DELIMITER //

CREATE PROCEDURE GetUserInfo (IN uid INT)
BEGIN
    SELECT name, age, phone
    FROM Users
    WHERE user_id = uid;
END //

DELIMITER ;

-- Function: Count Completed Sessions
DELIMITER //

CREATE FUNCTION CountCompletedSessions(uid INT)
RETURNS INT
DETERMINISTIC
BEGIN
    DECLARE total INT;
    SELECT COUNT(*) INTO total
    FROM Sessions
    WHERE user_id = uid AND status = 'Completed';
    RETURN total;
END //

DELIMITER ;

-- Trigger: After Session Insert
DELIMITER //
```

```
CREATE TRIGGER after_session_insert
AFTER INSERT ON Sessions
FOR EACH ROW
BEGIN
    INSERT INTO session_log (session_id, log_message)
    VALUES (NEW.session_id, CONCAT('New session added with ID: ',
NEW.session_id));
END //

DELIMITER ;
```

📌 Final Review Before Conclusion (Summary Points):

Before the conclusion, I can add a few reflective or analytical points to bridge the transition. And these are:

🔍 Each database object (table, relation, key, constraint) was carefully designed to reflect real-world mental health workflows.

🧠 PL/SQL procedures and triggers helped automate tasks and enforce business logic, reducing manual intervention.

📊 Data was tested using a wide variety of queries, including joins, subqueries, and aggregate functions to ensure system flexibility.

🛠️ Proper normalization ensured minimal redundancy and improved data consistency.

🧪 Practical testing with MySQL Workbench helped debug, optimize, and visualize outputs clearly.

🎯 Overall, the system is scalable and can be extended with new modules like appointment reminders, therapy notes, or analytics.

📌 Conclusion:

The Mental Health Support System project served as a practical demonstration of designing, building, and interacting with a relational database for a socially relevant application.

By completing all phases — from planning, ER diagram creation, normalization, query design, to PL/SQL development — we developed a comprehensive understanding of how databases function in real-world digital systems.

📌 Key outcomes:

✔ Enhanced knowledge in database modeling and implementation.

✔ Learned to maintain data integrity through constraints and normalization.

✔ Built dynamic SQL queries and logical PL/SQL code for automation.

✔ Practiced systematic documentation and query testing using MySQL Workbench.

✔ Requirement analysis

✔ Data modeling using ER diagrams

✔ Normalization techniques (1NF → 3NF)

✔ SQL DDL & DML operations

✔ Advanced query design & PL/SQL logic

📚 References & Resources:

- MySQL Official Documentation – https://dev.mysql.com/doc/
- W3Schools SQL Tutorial – https://www.w3schools.com/sql/
- TutorialsPoint DBMS Guide – ttps://www.tutorialspoint.com/dbms/
- Stack Overflow (for debugging SQL queries) – https://stackoverflow.com
- ChatGPT (OpenAI) – for help with ER diagram logic, SQL code examples, and report writing
- Course textbook: "Database System Concepts" by Silberschatz, Korth, Sudarshan

Not just queries or a data stream,
But empathy shaped into a digital dream.
Behind each table, a thoughtful vision —
That's our "Mental Health Support System" mission.

— End of Report —