Wolaita sodo university

Collage of informatics

Department of computer science

	NAME	UGR
1	Mekides alemayewu	ugr/91975 /16
2	Metages oroba	ugr/92705/16
3	bereket tilahun	ugr/93801/16
4	maidote birhan	ugr/91947/16
5	bekalu dems	ugr/92553/16

Introduction

Hospital DB is a comprehensive system Supports day-to-day hospital operations Ensures secure, reliable, and scalable data management Future-ready for integration with health analytics tools

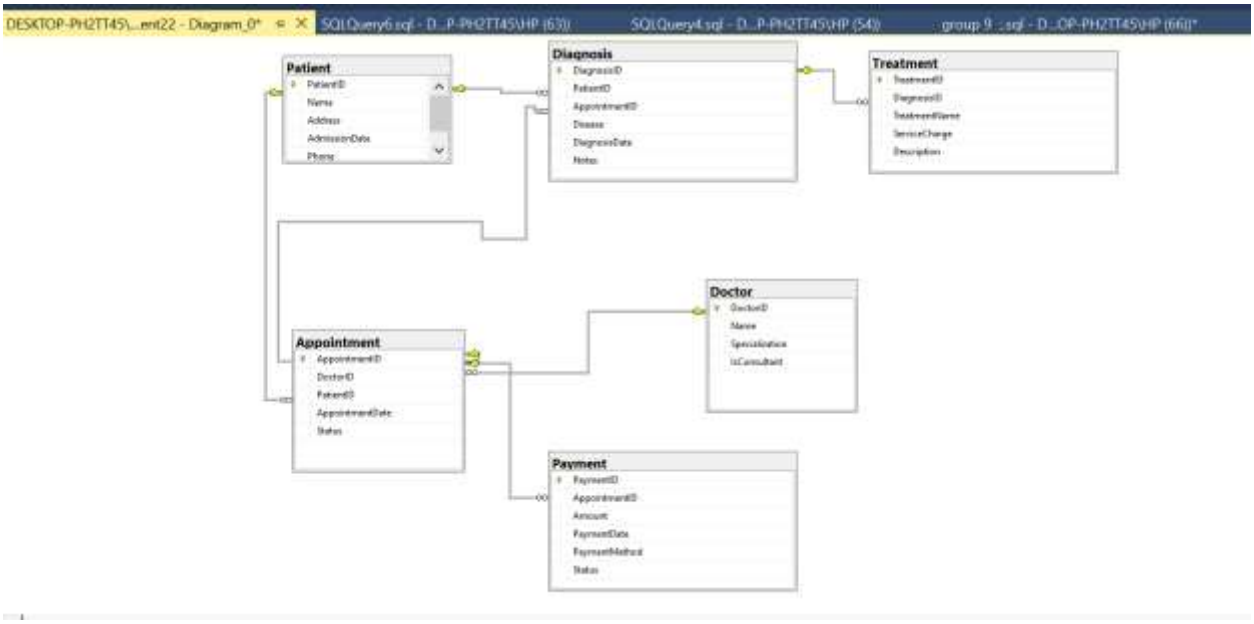
Efficiently manage hospital operations Provide centralized storage of all medical data Improve access to critical information Support decision-making and reporting

Hospital DB is a comprehensive system Supports day-to-day hospital operations Ensures secure, reliable, and scalable data management

Purpose

Improves efficiency and accuracy Reduces paperwork and manual errors Enables fast access to data Enhances patient care and satisfaction The main purpose of this project is to computerize all details regarding patient details and hospital details. Organizes and stores patient ,doctor, and treatment records systematically. Facilitates smooth appointment scheduling between patients and doctors. Tracks diagnosesandtreatmenthistoriesaccurately.Ensuresdataconsistencyandaccessibilityforbetterhospitaldecision-making.Reduces manual paperwork administrative burden and human error.

1, DESIGN ARELATIONAL DATABASE APPLICATION BASED ON YOUR CHOSEN PROJECT TITLE



Patient

Represents individuals receiving medical care. Stores personal and contact information.

- PatientID: Unique identifier for each patient (Primary Key).

- Name: Full name of the patient.

- Address: Patient's residential address.

- AdmissionDate: Date when the patient was admitted to the healthcare facility.

- Phone: Contact phone number of the patient.

Appointment

Records scheduled meetings between patients and doctors.

- AppointmentID: Unique identifier for each appointment (Primary Key).

- DoctorID: References the doctor assigned to the appointment (Foreign Key).

- PatientID: References the patient who booked the appointment (Foreign Key).

- AppointmentDate: Date and time when the appointment is scheduled.

- Status: Current status of the appointment (e.g., scheduled, completed, canceled).

Doctor

Contains information about medical professionals providing care.

- DoctorID: Unique identifier for each doctor (Primary Key).

- Name: Full name of the doctor.

- Specialization: Area of medical expertise (e.g., cardiology, orthopedics).

- IsConsultant: Boolean or flag indicating if the doctor holds a consultant position.

Diagnosis

Details the medical assessment made for a patient during an appointment.

- DiagnosisID: Unique identifier for each diagnosis record (Primary Key).

- PatientID: References the patient diagnosed (Foreign Key).

- AppointmentID: References the appointment linked to the diagnosis (Foreign Key).

- Disease: Name or description of the diagnosed disease or condition.

- DiagnosisDate: Date when the diagnosis was made.

- Notes: Additional remarks or observations related to the diagnosis.

Treatment

Describes medical procedures or therapies prescribed for a diagnosis.

- TreatmentID: Unique identifier for each treatment record (Primary Key).

- DiagnosisID: References the diagnosis associated with the treatment (Foreign Key).

- TreatmentName: Name of the treatment or procedure.

- ServiceCharge: Cost associated with the treatment.

- Description: Details or explanation of the treatment.

Relationships:

- Patient to Appointment: One-to-many (one patient can have many appointments).

- Doctor to Appointment: One-to-many (one doctor can have many appointments).

- Appointment to Diagnosis: One-to-one or one-to-many (each appointment may have one or more diagnoses).

- Diagnosis to Treatment: One-to-many (one diagnosis can have multiple treatments).

- Appointment to Payment: One-to-many (one appointment can have multiple payments).

- Patient to Diagnosis: One-to-many (one patient can have many diagnoses).

2, IMPLIMENT TRANSACTION MANGEMENT FOR CRITICAL OPERATION

Definition:

Transactions are a set of database operations that must be executed as a single logical unit to ensure data consistency and integrity.

Goal:

Guarantee that critical operations (e.g., creating an appointment, recording diagnosis and treatment, updating payment) are atomic, consistent, isolated, and durable (ACID).

Example in Hospital System:

When a patient books an appointment:

- Insert a new record in the Appointment table.

- If successful, update the Payment table with the corresponding payment details.

- If either fails, rollback to keep the database consistent.

Key Components:

- BEGIN TRANSACTION: Start the transaction.

- COMMIT: Make all changes permanent if no errors.

- ROLLBACK: Undo changes if an error occurs.

3: Demonstrate the Use of Commit, Rollback, and Savepoints

Definition:

These are essential commands to control transactions.

COMMIT:

- Confirms and saves all changes made during the transaction to the database.

- Example: Once the treatment details are updated, commit the transaction so that data is permanently stored.

- ROLLBACK:

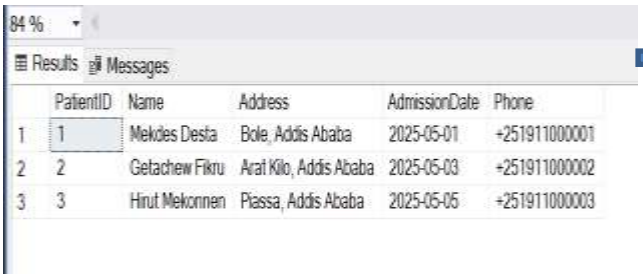
- Cancels all changes made in the transaction, restoring the database to its state before the transaction began.

- Example: If adding a new diagnosis record fails due to a constraint error, rollback to ensure no partial changes remain.

- SAVEPOINT:

- Marks a specific point within a transaction to which you can rollback

LET SEE



	PatientID	Name	Address	AdmissionDate	Phone
1	1	Mekdes Desta	Bole, Addis Ababa	2025-05-01	+251911000001
2	2	Getachew Fikru	Arat Kilo, Addis Ababa	2025-05-03	+251911000002
3	3	Hirut Mekonnen	Piassa, Addis Ababa	2025-05-05	+251911000003

Before transaction

h

	PatientID	Name	Address	AdmissionDate	Phone
1	1	Mekdes Desta	Bole, Addis Ababa	2025-05-01	+251911000001
2	2	Getachew Fikru	Arat Kilo, Addis Ababa	2025-05-03	+251911000002
3	3	Hirut Mekonnen	Piassa, Addis Ababa	2025-05-05	+251911000003
4	4	Selamawit Tulu	Megenagna, Addis Ababa	2025-05-24	+251911000004

After transaction and commit

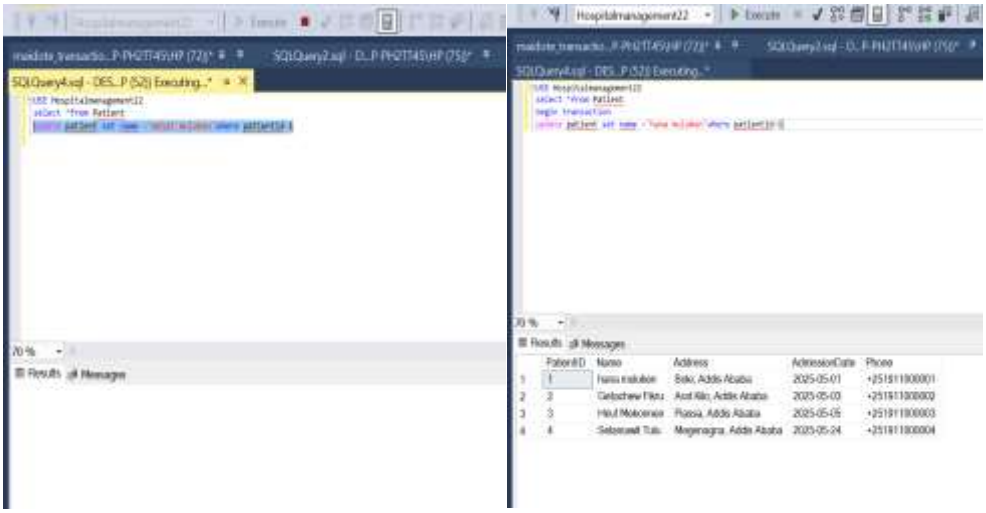
	PatientID	Name	Address	AdmissionDate	Phone
1	1	Mekdes Desta	Bole, Addis Ababa	2025-05-01	+251911000001
2	2	Getachew Fikru	Arat Kilo, Addis Ababa	2025-05-03	+251911000002
3	3	Hirut Mekonnen	Piassa, Addis Ababa	2025-05-05	+251911000003
4	4	Selamawit Tulu	Megenagna, Addis Ababa	2025-05-24	+251911000004
5	5	Eyerusalem Hailu	CMC, Addis Ababa	2025-05-24	+251911000008

After transaction and save point(not commit)

	PatientID	Name	Address	AdmissionDate	Phone
1	1	Mekdes Desta	Bole, Addis Ababa	2025-05-01	+251911000001
2	2	Getachew Fikru	Arat Kilo, Addis Ababa	2025-05-03	+251911000002
3	3	Hirut Mekonnen	Piassa, Addis Ababa	2025-05-05	+251911000003
4	4	Selamawit Tulu	Megenagna, Addis Ababa	2025-05-24	+251911000004

After roll back(save point transaction is backed but commit transaction is not rollback because it permanently saVED)

4, SIMULATE CONCURRENT TRANSACTION AND ANALYZE THE IMPACT OF CONCURRANCY CONTROL TECHNIQUIS



Concurrency control protocols ensure atomicity, isolation, and ***serializability*** of concurrent transactions.

The concurrency control protocol can be divided into the following categories:

- Lock based (Locking) protocol
- Time-stamp protocol
- Validation (optimistic) based protocol,
- Multi version protocol,
- Multiple granularity lock

From the above we did by locked based

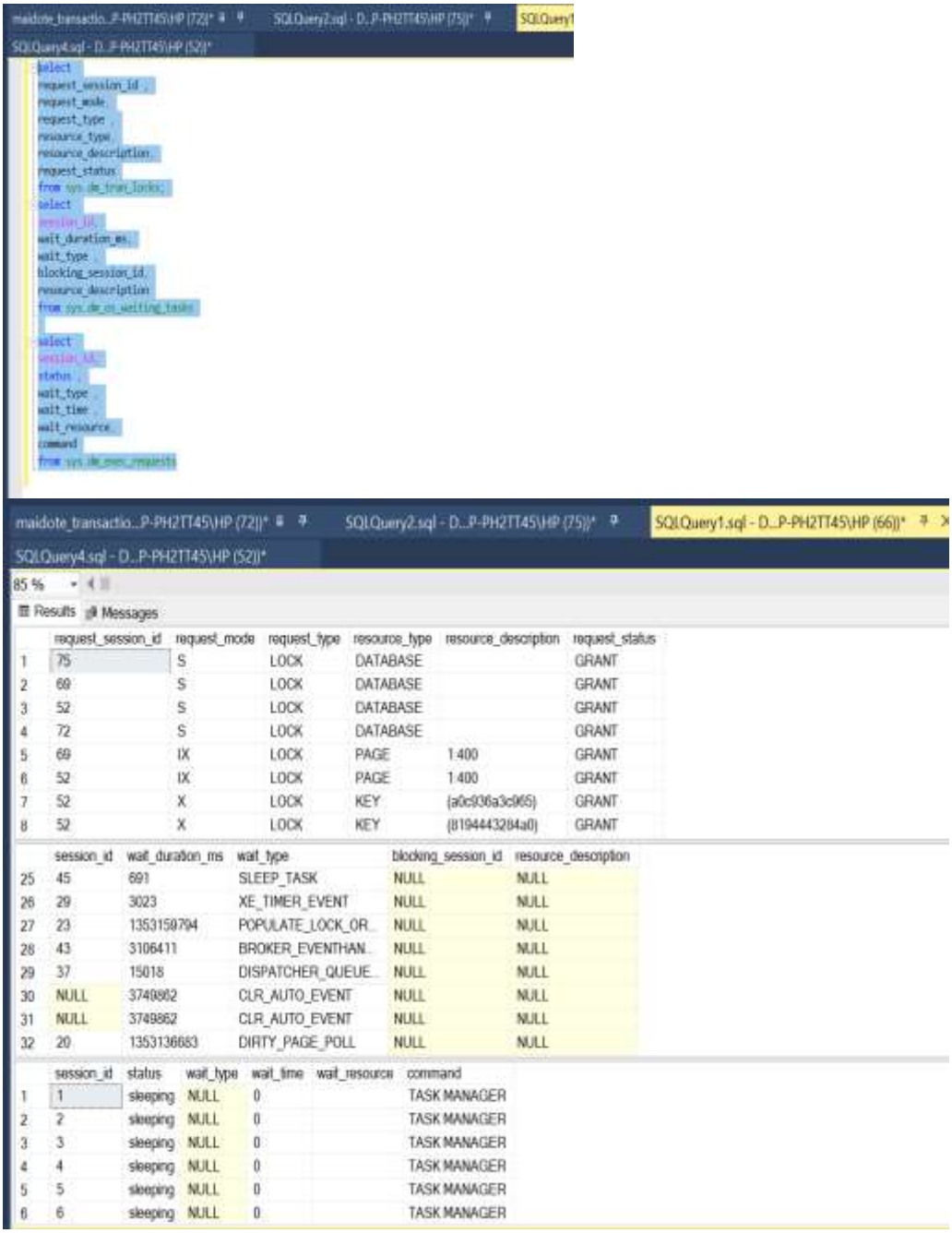
They manage simultaneous access to data items by multiple transactions, preventing conflicts and ensuring data integrity.

Shared Lock (S-lock) or Read Lock:

Exclusive Lock (X-lock) or Write Lock:

When a transaction executes Lock(X), it is requesting and being granted a lock on the data item X. The type of lock (shared or exclusive) determines whether other transactions can also read X or if the requesting transaction has exclusive access for modification.

Task 5



```
select
request_session_id ,
request_mode,
request_type ,
resource_type,
resource_description,
request_status
from sys.dm_tran_locks;
select
session_id,
wait_duration_ms,
wait_type ,
blocking_session_id,
resource_description
from sys.dm_os_waiting_tasks
select
session_id,
status ,
wait_type ,
wait_time ,
wait_resource,
command
from sys.dm_exec_requests
```

maidote_transactio...P-PH2TT45\HP (72))* | SQLQuery2.sql - D...P-PH2TT45\HP (75))* | SQLQuery1.sql - D...P-PH2TT45\HP (66))*

SQLQuery4.sql - D...P-PH2TT45\HP (52))*

85 %

Results | Messages

	request_session_id	request_mode	request_type	resource_type	resource_description	request_status
1	75	S	LOCK	DATABASE		GRANT
2	69	S	LOCK	DATABASE		GRANT
3	52	S	LOCK	DATABASE		GRANT
4	72	S	LOCK	DATABASE		GRANT
5	69	IX	LOCK	PAGE	1:400	GRANT
6	52	IX	LOCK	PAGE	1:400	GRANT
7	52	X	LOCK	KEY	(a0c936a3c965)	GRANT
8	52	X	LOCK	KEY	(8194443284a0)	GRANT

	session_id	wait_duration_ms	wait_type	blocking_session_id	resource_description
25	45	691	SLEEP_TASK	NULL	NULL
26	29	3023	XE_TIMER_EVENT	NULL	NULL
27	23	1353159794	POPULATE_LOCK_OR...	NULL	NULL
28	43	3106411	BROKER_EVENTHAN...	NULL	NULL
29	37	15018	DISPATCHER_QUEUE...	NULL	NULL
30	NULL	3749862	CLR_AUTO_EVENT	NULL	NULL
31	NULL	3749862	CLR_AUTO_EVENT	NULL	NULL
32	20	1353136683	DIRTY_PAGE_POLL	NULL	NULL

	session_id	status	wait_type	wait_time	wait_resource	command
1	1	sleeping	NULL	0		TASK MANAGER
2	2	sleeping	NULL	0		TASK MANAGER
3	3	sleeping	NULL	0		TASK MANAGER
4	4	sleeping	NULL	0		TASK MANAGER
5	5	sleeping	NULL	0		TASK MANAGER
6	6	sleeping	NULL	0		TASK MANAGER

Explanation of Each Query

Query 1: sys.dm_tran_locks

- Purpose: Shows information about current locks held or requested by active transactions.

- Use: Helps identify which sessions hold locks and what type of locks they have, useful to detect blocking and deadlocks.

Query 2: sys.dm_os_waiting_tasks

Purpose: Displays tasks that are currently waiting (blocked) and the reasons why.

- Use: Helps identify blocking situations and which sessions/resources are causing delays.

Query 3: sys.dm_exec_requests

- Purpose: Shows details about currently executing requests (queries) on SQL Server.

- Use: Provides insight into active queries, their status, and whether they are waiting on resources, aiding in performance and blocking analysis.

Summary

Together, these queries help database administrators:

- Monitor locks held by transactions.

- Detect blocking and waiting tasks.

- Identify deadlocks or long waits.

- Analyze which sessions or queries are causing contention.

====

Create a new login for SQL Server

```
CREATE LOGIN CSG5 WITH PASSWORD = '1010';
```

GO

Create a new user in the current database associated with the login

```
CREATE USER USER5 FOR LOGIN CSG5;
```

GO

Create a new role in the current database

```
CREATE ROLE CSG5;
```

GO

Add the user to the newly created role

```
EXEC sp_addrolemember 'CSG5', 'USER5';
```

GO

Grant SELECT and INSERT permissions on the Doctor table to the user with grant option

```
GRANT SELECT, INSERT ON Doctor TO USER5 WITH GRANT OPTION;
```

GO

Revoke the grant option for SELECT and INSERT on the Doctor table from the user

REVOKE GRANT OPTION FOR INSERT, SELECT ON Doctor FROM USER5 CASCADE;

GO

Key Points:

1. Creating Login and User: The CREATE LOGIN statement creates a new SQL Server login, and CREATE USER associates that login with a database user.

2. Creating Role: The CREATE ROLE statement creates a new role in the database. Roles are used to group users together for easier permission management.

3. Adding User to Role: The EXEC sp_addrolemember command adds the user to the specified role.

4. Granting Permissions: The GRANT statement gives the specified permissions (in this case, SELECT and INSERT) on the Doctor table to the user. The WITH GRANT OPTION allows the user to grant those permissions to others.

5. Revoking Permissions: The REVOKE statement removes the granted permissions from the user. The CASCADE option ensures that any dependent permissions are also revoked.

Notes:

• Ensure that you have sufficient privileges to execute these commands.

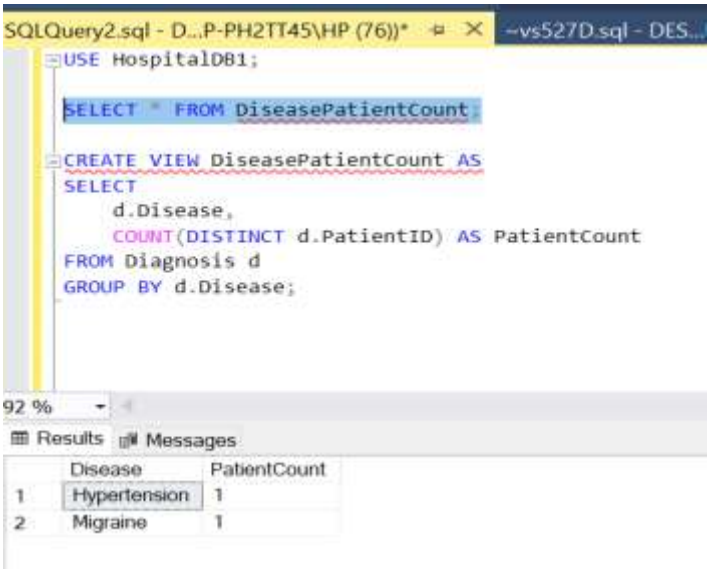
• Always use strong passwords in production environments.

• Be cautious when granting permissions and using the WITH GRANT OPTION, as it can lead to privilege escalation if not managed properly.

• The CSG5 role name could be confusing since it matches the login name; consider using more descriptive names for roles.

Task 6

Personal data SAFE WHEN SHARING REPORTS



VIEW only number of patients with disease it not show about patients personal information

It hide patient identity