

## Database 2

Khaula Molapo

Year 2 Semester 2

### Question 1

Normalizing a hospital database ensures efficient storage, reduces redundancy and maintains data integrity. Starting with unorganized data, we split information into separate tables. In 3NF every non-key attribute depends only on the primary key, preventing duplication and anomalies.

Patients table (3NF)

Patient ID	Name	Age	Gender
1	John	30	M
2	Mary	25	F

Appointment table (3NF)

Appointment ID	Patient ID	Date	Doctor
101	1	2025-08-23	Dr.A
102	2	2025-08-24	Dr. B

Here, patient ID links appointments to patients, ensuring no duplicate patient details. Normalization enforces data integrity by keeping relationships consistent, preventing inconsistencies like age for the same patient. It also simplifies updates, deletions and queries, making the database more reliable and scalable for hospital operations.

### Question 2

Unnormalized table

Student ID	Name	Dept. ID	Dept. Name	Dept. Head
1	Alice	D01	Computer Sci	Dr. Brown
2	Bob	D02	Mathematics	Dr. Green
3	Charlie	D01	Computer Sci	Dr. Brown

## Normalized tables (3NF)

### Students

Student ID(pk)	Name	Dept. ID(fk)
1	Alice	D01
2	Bob	D02
3	Charlie	D01

### Departments

Dept.ID	Dept. Name	Dept. Head
D01	Computer Sci	Dr. Brown
D02	Mathematics	Dr. Green

Normalizing a student database separates repeating and dependant data into distinct tables, reducing redundancy and improving efficiency. By moving departments details into a separate table, it ensures each department is stored once. Students reference departments via foreign keys, which maintains consistency and simplifies updates. This process eliminates anomalies, such as inconsistent department heads or duplicate department names. In 3NF, all attributes depend only on their primary key, ensuring no transitive dependencies. This structure improves data integrity, simplifies querying and supports reliable reporting. Overall, normalization provides a clean, scalable and maintainable database structure for student and department management.

### Question 3

Normalization improves performance in an e-commerce database by organizing data into related tables, reducing redundancy and maintaining consistency. Products, customers and orders are separated so each piece of information is stored only once, preventing duplications. This structure speeds up queries, as the system retrieves precise data without scanning repetitive entries. It also eliminates anomalies, like conflicting product prices or multiple shipping addresses for a single customer. Primary and foreign keys maintain relationships accurately, ensuring data integrity. Overall normalization enhances scalability, simplifies maintenance, supports reliable reporting and allows the e-commerce platform to handle growth efficiently, securely and consistently.

### Question 4

Second normal form (2NF) requires a table to be in 1NF and ensures that all non-key attributes fully depend on the entire primary key, not just part of it. Third normal form (3NF) builds on 2NF and requires that non-key attributes depend only on the primary key, with no transitive dependencies through other non-key. These rules reduce redundancy, prevent update and delete anomalies.

Example;

Order table

Order ID	Product ID	Quantity ID	Product name	Price
----------	------------	-------------	--------------	-------

Split into;

Orders

Order ID	Product ID	Quantity
----------	------------	----------

Products

Product ID	Product name	Price
------------	--------------	-------

This keeps the database clean, consistent and easy to manage. It also makes querying faster because each table contains only relevant information. Updating and deleting is now easier and safer, reducing errors or accidental data loss. Adding new products, orders or changes to department details becomes straight forward without affecting unrelated data. Proper normalization is essential for long term efficiency and smooth operations in real world applications.