Course Code	**:**	**MCS-207**
Course Title	**:**	**Database Management Systems**
Assignment Number	**:**	**BCA_NEW/BCAOL(III)207/Assign/2026-27**
Maximum Marks	**:**	**100**
Last Date of Submission	**:**	**31st October, 2026 (For July 2026 Session)**
	:	**30th April, 2027 (For January 2027 Session)**

Note: There are FOUR questions in this assignment, which carry 80 marks. Rest 20 marks are for viva voce. You may use illustrations and diagrams to enhance the explanation. Please go through the guidelines regarding assignments given in the Programme Guide for the format of the presentation. The answer to each part of the question should be confined to about 300 words. Make suitable assumptions, if any.

Q1: Foundations of Data-Centric Database Systems **(5 × 4 = 20 Marks)**

a) Data scientists often work with large, evolving datasets. Explain the **limitations of traditional file-based systems** in the context of analytics and machine learning workflows. How does a DBMS address these limitations?

b) Explain the following concepts with respect to the **relational data model**, giving one data-science–oriented example for each:

- Candidate Key
- Functional Dependency
- Referential Integrity
- Selection Operation
- Projection Operation

c) A **health analytics platform** maintains data about patients, diagnostic tests, doctors, and test results. Analysts want to query trends across diseases, age groups, and regions. Design an **ER diagram** for this system. Clearly identify entities, relationships, key attributes, and constraints. State assumptions made.

d) Convert the ER diagram designed in part (c) into **normalized relations up to 3NF**, clearly indicating primary and foreign keys.

e) Explain the importance of **indexes** in analytical databases. Differentiate between primary index, secondary index, and clustering index with suitable examples.

Q2: Data Normalization, Dependencies, and SQL for Analytics **(4+4+4+8= 20 Marks)**

a) Consider a Relation:

Dataset(DatasetID, DatasetName, Source, CollectionDate, Domain, OwnerName, OwnerEmail, UpdateFrequency)

- Identify the **primary key**
- List meaningful **functional dependencies**

b) Populate the relation with **8–10 sample records** and highlight potential **data redundancy and anomalies**.

c) Decompose the above relation into **2NF and 3NF**, explaining each step clearly.

d) Consider the following relations used in a **research publication analytics system**:

- Researcher(ResearcherID, Name, Email, Affiliation)
- Publication(PubID, Title, Year, Venue)
- Authorship(ResearcherID, PubID, AuthorOrder)

Perform the following using SQL:

- Create tables with appropriate primary and foreign keys
- Insert sample data (minimum 5 researchers, 6 publications)
- List publications by a given researcher
- Find researchers who have not authored any publication
- Find the publication with the highest number of authors
- List venues that have more than two publications

Q3: Transactions, Concurrency, and Consistency in Data Systems (4 + 8 + 8 = 20 Marks)

a) Explain the **ACID properties** of transactions with the help of an example of banking transaction; "Transfer of an amount of rupees 50,000/- from an account ABC to XYZ".

b) Consider the following schedule involving two transactions T1 and T2 operating on a dataset table storing aggregate metrics:

Time	**T1**	**T2**
t1	READ(X)	
t2	X = X + 50	
t3		READ(X)
t4		X = X * 1.2
t5		WRITE(X)
t6	WRITE(X)	

i) Compute the **final value of X** (assume initial X = 100).
ii) Determine whether the schedule is **serializable**.
iii) Identify the concurrency issue involved and explain its impact on analytics accuracy.

c) Explain the **Two-Phase Locking (2PL) protocol**. Discuss how it ensures consistency in multi-user analytical environments. Can deadlocks still occur? Explain with a suitable example.

Q4: Advanced Topics and Case-Based Understanding (5 × 4 = 20 Marks)

Write short notes on the following. Support your answers with examples relevant to data science or analytics.

a) Centralized vs Distributed Databases in Large-Scale Analytics
b) Star Schema and Snowflake Schema in Data Warehousing
c) NoSQL Databases for Data Science (Explain one type with use cases)
d) Query Optimization Techniques for Analytical Queries
e) Log-Based Recovery and Checkpointing in Data-Intensive Systems

Q1: Foundations of Data-Centric Database Systems (5 × 4 = 20 Marks)

a) Data scientists often work with large, evolving datasets. Explain the limitations of traditional file-based systems in the context of analytics and machine learning workflows. How does a DBMS address these limitations?

Answer

Traditional file-based systems keep data in independently maintained files such as CSV files, spreadsheets and application-specific binary files. They are useful for small, static tasks but become difficult to manage when a data-science team collects data continuously from multiple sources. A file does not automatically enforce a common schema or coordinate simultaneous changes.

Limitations in analytics and machine learning

- **Redundancy and inconsistency:** Customer or device information copied into several files may be updated in one copy but not another. Different model-training runs may then use conflicting values.
- **Isolation and difficult integration:** Analysts need separate scripts to combine logs, surveys and transaction files. Changed column names, missing fields and inconsistent date formats often break an ETL pipeline.
- **Weak concurrency and recovery:** Two users can overwrite the same file; a failed write may leave an incomplete dataset. Ordinary files lack built-in transactions, rollback and crash-recovery guarantees.
- **Limited security and governance:** File permissions do not easily express row- or column-level access, controlled updates, audit trails or dependable lineage for sensitive healthcare data.
- **Slow selective access:** A program may need to scan an entire large file to answer a small query, and there is little automatic optimization of joins or filtering.

How a DBMS helps

A database management system defines tables, keys and constraints so data has a controlled structure. SQL provides a uniform way to select, group and join related records. Indexes and the query optimizer reduce unnecessary reading; views can present reusable datasets for analysts. Transaction management provides ACID guarantees, and concurrency control stops conflicting updates from silently corrupting records. Backup, logs and recovery support dependable