

## Guidelines of B.Sc. (H) Computer Science IV Semester (NEP UGCF 2022)

### Database Management Systems

#### (DSC 11) Core Course

(Effective from Academic Year 2023-24)

Unit	Topics	Chapter	Ref.	Suggested Number of Weeks
1	<b>Introduction to Database:</b> Purpose of database system, Characteristics of database approach, data models, database management system, database system architecture, three-schema architecture, components of DBMS, data independence, and file system approach vs database system approach.	1.1 - 1.3, 1.4 - 1.5, 1.6, 1.8, 2.1 - 2.2, 2.3.1, 2.4 - 2.4.1, 2.6	1	2
2	<b>Entity Relationship (ER) Modeling:</b> conceptual data modeling - motivation, entities, entity types, attributes, relationships, relationship types, constraints on relationship, Entity Relationship diagram notation.	3 (except 3.8)	1	2
3	<b>Relational Data Model:</b> Update anomalies, Relational Data Model - Concept of relations, schema-instance distinction, keys, relational integrity constraints, referential integrity and foreign keys, relational algebra operators and queries.	5, 8.1 - 8.3.3, 8.4 - 8.5	1	2
4	<b>Structured Query Language (SQL):</b> Querying in SQL, DDL to create database and tables, table constraints, update database-update behaviours, DML, aggregation functions group by and having clauses, retrieve data from the database, generate and query views. Access and manipulate databases using ODBC. Basic Database administration SQL commands.	6, 7 - 7.1.8, 7.3 - 7.4	1	3
		2*(Pg. 48), 3*(Listing 3.4)	2	
		11*(Pg. 356-357), 18*(Pg. 532-552)	3	
5	<b>Database Design:</b> Mapping an Entity Relationship (ER) model to relational database, functional dependencies and Normal forms, 1NF, 2NF, 3NF and BCNF decompositions and desirable properties of them.	9.1, 14.1 - 14.5, 15.1 .1	1	3
6	<b>File indexing and Transaction Processing:</b> Data Storage and Indexes- Need of file indexes, file organizations, index structures, single- and multi-level indexing, concurrent execution of transactions, ACID properties, need of data recovery and log file.	16.6 - 16.7, 17.1 - 17.2, 20.1 - 20.3	1	3

\* To be covered in practical only

## References

1. Elmasri, R., Navathe, B. S. *Fundamentals of Database Systems*, 7<sup>th</sup> Edition, Pearson Education, 2015.
2. Krogh, J. W. *MySQL Connector/Python Revealed: SQL and NoSQL Data Storage Using MySQL for Python Programmers*, Apress, 2018.
3. Murach J. *Murach's MySQL*, 3<sup>rd</sup> edition, Pearson, 2019.

## Additional References

- (i) Ramakrishnan, R., Gehrke J. *Database Management Systems*, 3<sup>rd</sup> Edition, McGraw-Hill, 2014.
- (ii) Silberschatz, A., Korth, H. F., Sudarshan S. *Database System Concepts*, 7<sup>th</sup> Edition, McGraw Hill, 2019.
- (iii) Connolly, T. M., Begg, C. E. *Database Systems: A Practical Approach to Design, Implementation, and Management*, 6<sup>th</sup> edition, Pearson, 2019.

## Suggested Practical List

It has four components.

- I. Create and use the following student-society database schema for a college to answer the given (sample) queries using the standalone SQL editor.

STUDENT	<u>Roll No</u>	StudentName	Course	DOB
	Char(6)	Varchar(20)	Varchar(10)	Date

SOCIETY	<u>SocID</u>	SocName	MentorName	TotalSeats
	Char(6)	Varchar(20)	Varchar(15)	Unsigned int

ENROLLMENT	<u>Roll No</u>	<u>SID</u>	DateOfEnrollment
	Char(6)	Char(6)	Date

Here Rollno (ENROLLMENT) and SID (ENROLLMENT) are foreign keys.

1. Retrieve names of students enrolled in any society.
2. Retrieve all society names.
3. Retrieve students' names starting with letter 'A'.
4. Retrieve students' details studying in courses 'computer science' or 'chemistry'.
5. Retrieve students' names whose roll no either starts with 'X' or 'Z' and ends with '9'.
6. Find society details with more than N TotalSeats where N is to be input by the user.
7. Update society table for mentor name of a specific society.

8. Find society names in which more than five students have enrolled
9. Find the name of youngest student enrolled in society 'NSS'
10. Find the name of most popular society (on the basis of enrolled students)
11. Find the name of two least popular societies (on the basis of enrolled students)
12. Find the student names who are not enrolled in any society
13. Find the student names enrolled in at least two societies
14. Find society names in which maximum students are enrolled
15. Find names of all students who have enrolled in any society and society names in which at least one student has enrolled
16. Find names of students who are enrolled in any of the three societies 'Debating', 'Dancing' and 'Sashakt'.
17. Find society names such that its mentor has a name with 'Gupta' in it.
18. Find the society names in which the number of enrolled students is only 10% of its capacity.
19. Display the vacant seats for each society.
20. Increment Total Seats of each society by 10%
21. Add the enrollment fees paid ('yes'/'No') field in the enrollment table.
22. Update date of enrollment of society id 's1' to '2018-01-15', 's2' to current date and 's3' to '2018-01-02'.
23. Create a view to keep track of society names with the total number of students enrolled in it.
24. Find student names enrolled in all the societies.
25. Count the number of societies with more than 5 students enrolled in it
26. Add column Mobile number in student table with default value '9999999999'
27. Find the total number of students whose age is > 20 years.
28. Find names of students who are born in 2001 and are enrolled in at least one society.
29. Count all societies whose name starts with 'S' and ends with 't' and at least 5 students are enrolled in the society.
30. Display the following information:  

Society name	Mentor name	Total Capacity	Total Enrolled	Unfilled Seats
--------------	-------------	----------------	----------------	----------------

## II. Do the following database administration commands:

create user, create role, grant privileges to a role, revoke privileges from a role, create index

## III. Execute queries given in part I through a high-level language using ODBC connection.

## IV. Students should implement the COMPANY database schema from Chapter 3 [1] and execute the solved queries of Chapter 7 [1].

