Databases

Section A

1. Define a **relational database**.
2. What is a **table** in a database?
3. Explain the purpose of a **primary key**.
4. What is a **field** in a database table?
5. Define the term **foreign key**.
6. What does **1NF** (First Normal Form) ensure in a database?
7. What is a **record** in a database?
8. List two disadvantages of a file-based approach.
9. Name three data types commonly used in SQL.
10. What is the purpose of the **SELECT** statement in SQL?
11. State one reason to normalize a database.
12. Define **referential integrity**.
13. What does the acronym **DBMS** stand for?
14. List two features of a DBMS.
15. Define **normalization** in databases.
16. Explain the term **tuple** in the context of a relational database.
17. Name the SQL command used to add data to a table.
18. Write a simple SQL command to display all rows in a table named Student.
19. What is the purpose of an **index** in a database?
20. Explain the relationship between **entities** and **attributes**.
21. What does **DDL** stand for?
22. Name the SQL command used to remove rows from a table.
23. Define the term **composite key**.
24. What is a **candidate key**?
25. List one benefit of using SQL in relational databases.
26. Name three tasks a Database Administrator (DBA) might perform.
27. What is the **GROUP BY** clause used for in SQL?
28. Define **many-to-many relationship** in database design.
29. State one advantage of a relational database over a file-based system.
30. What does the **ORDER BY** clause do in SQL?

Section B

31. Why is **data redundancy** a problem, and how does a DBMS address it?
32. Write an SQL query to retrieve all students in a class 7A, ordered by their last name.
33. What is the difference between a **logical schema** and a **physical schema**?
34. Explain how a **DBMS** ensures data security using **access rights**.
35. Illustrate the concept of a **one-to-many relationship** with an example.
36. Explain the difference between **1NF** and **2NF**.
37. Write an SQL query to count the number of rows in a table named Orders.
38. Describe how an **E-R diagram** represents relationships between entities.
39. Write the SQL command to create a foreign key relationship between Student and Class tables.
40. What are the advantages of splitting a large table into smaller, related tables?
41. Write an SQL query to find the average marks of students in a subject.
42. Describe the purpose of the **query processor** in a DBMS.
43. What is meant by the **cardinality** of a relationship? Provide an example.
44. Write the SQL command to delete all students born before the year 2000.
45. How does the **normalization process** reduce data redundancy?
46. What is the purpose of a **data dictionary** in a DBMS?

47. Describe the difference between a **primary key** and a **secondary key**.

48. Explain why **3NF** is considered optimal for most relational databases.

49. Write an SQL query to add a new column named Email to an existing table Employee.

50. Describe how a DBMS uses **metadata** to manage databases.

51. Explain the purpose of the **INNER JOIN** clause in SQL.

52. What are the steps involved in **designing a relational database**?

53. How can you enforce referential integrity in a relational database?

54. What is the difference between **DELETE FROM** and **DROP TABLE** in SQL?

55. Write an SQL query to find students who scored above 90 in an ExamResults table.

56. Explain the term **partial dependency** in the context of 2NF.

57. How does indexing improve database performance?

58. Write the SQL command to update a teacher's name from Mr. Smith to Mr. Jones in the Teacher table.

59. Describe how **relationships** are implemented between tables in a relational database.

60. Explain the term **query optimization**.

Section C

61. Normalize the following unnormalized table into 1NF, 2NF, and 3NF:

62. Table: Employee(EmpID, EmpName, Dept, DeptLocation, ManagerName)

63. Write the SQL query to list all students and their subjects, using a join between Student and Subject tables.

64. Design an E-R diagram for a university database storing data about students, courses, and professors.

65. Write an SQL query to calculate the total sales for each salesperson, grouped by region.

66. Explain the challenges of maintaining referential integrity in a distributed database.

67. Write the SQL commands to create a database for a library system with tables for books, authors, and borrowers.

68. Discuss the implications of redundant data on database consistency and performance.

69. Explain how normalization can impact query performance negatively.

70. Write an SQL query to display all employees who have the same manager in the Employee table.

71. Suggest and explain improvements to a poorly normalized database structure.

72. Describe how a DBMS handles simultaneous updates to the same record.

73. What is the difference between **clustered** and **non-clustered indexes**?

74. Write an SQL script to display the top three students by average score in their subjects.

75. Create a database structure for an e-commerce platform, ensuring it is normalized to 3NF.

76. Explain the role of **ACID properties** in maintaining transactional integrity in databases.

77. Write an SQL query to retrieve the names of employees who do not belong to any department.

78. How would you design a database to support versioning of records?

79. Explain the concept of **denormalization** and when it might be used.

80. Write an SQL query to find all customers who placed more than five orders last year.
81. Discuss the trade-offs between using **fixed-length** and **variable-length** fields in database design.
82. Design a fully normalized database for tracking student attendance, courses, and teachers.
83. What are the implications of a database lock in multi-user systems?
84. Write an SQL query to retrieve the names of students studying the maximum number of subjects.
85. How can recursive relationships be implemented in a database? Provide an example.
86. Explain how **hashing** is used in database indexing.
87. Write an SQL query to calculate the rank of students based on their scores.
88. Discuss potential problems that can arise in a database with poor normalization.
89. How would you handle changes to a database schema in a production environment?
90. Write an SQL script to backup a database and explain its components.