Odd Semester, 2020

(Held in March, 2021)

COMPUTER SCIENCE

(Elective/Honours)

(CS-301 T)

(Database Management System)

Marks : 56

Time : 3 hours

The figures in the margin indicate full marks for the questions

Answer *any* **one** from each Unit

UNIT—I

1. *(a)* Write a short note on the database approach. What is the role of a DBA? 3+2=5

(b) Explain the three-tier schema architecture of database. 6

(c) Explain cardinality ratio with examples. 4

2. *(a)* A company needs to store information about employees (identified by ssn, with salary and phone as attributes), departments (identified by dno, with dname and budget as attributes), and children of employees (with name and age as attributes). Employees work in departments, each department is managed by an employee; a child must be identified uniquely by name when the parent (who is an employee; assume that only one parent works for the company) is known. We are not interested in information about a child once the parent leaves the company. Design an E-R diagram indicating all entities with generalization and specialization, attributes with key and cardinality ratio. 10

(b) What is data independence? Explain the difference between physical and logical data independence. 2+3=5

UNIT—II

3. *(a)* What is clustered index organization? Illustrate with example. 4

(b) Consider a disk block with block size of 128 bytes. A block pointer is 2 bytes and a record pointer is 6 bytes long. A file has 1000 employee records of

fixed length. Each record has following fields : Employee_id (4 bytes), Name (50 bytes), Department (10 bytes), Address (50 bytes), Designation (10 bytes) and salary (6 bytes). An additional byte is used for deletion marker. Suppose that the file is ordered by the key field Employee_id, and a primary index is constructed of this. Calculate—

(i) the index blocking factor;

(ii) the number of first-level index entries;

(iii) the number of first-level index blocks;

(iv) the number of levels needed if we make it a multilevel index. 1×4=4

4. *(a)* Elaborate the relation between bucket size and overflow in hashing. 3

(b) Explain RAID technology highlighting data mirroring and data striping. 5

UNIT—III

5. *(a)* Explain different data types with example in SQL. 5

(b) Explain keys, foreign key constraints that can be specified on relations. 1+2+2=5

(c) Discuss about outer joins with example. 5

6. *(a)* Consider the following reation :

Project (proj#, proj_name, chief_architect)
Employee (emp#, emp_name)
ASSigned (Proj#, emp#)

Use relational algebra to express the following queries :

(i) Get details of employees working on some projects.

(ii) Get the employee numbers of employees who work on all projects.

(iii) Get details of project on which employee with name "AAA" is working. 2×3=6

(b) What is a join? Explain about EQUI join and natural join with syntax and example. 2+4=6

(c) Define a NULL value. How do you retrieve null values from the database? 1+2=3

UNIT—IV

7. *(a)* Explain normalization of DBMS. 4

(b) Why is BCNF considered a stricter form of 3NF? 2

8. *(a)* Suppose you are given a relation R{A, B, C, D, E} with the following functional dependencies :

$$F = \{AB \rightarrow C, DC \rightarrow AE, E \rightarrow F\}$$

(i) What are the keys of this relation?

(ii) Is this relation in BCNF? If not, explain why by showing one violation. 2+2=4

(b) Explain multivalued dependencies. 2

UNIT—V

9. *(a)* Explain ACID properties of transaction management. 6

(b) What is the difference between a serial schedule and a serializable schedule? When are two operations said to be conflicting? 3+3=6

10. *(a)* Explain the term 'recoverability'. Explain a cascadeless schedule. 2+2=4

(b) Explain deadlock with example. 2

(c) Draw the precedence graph of the following schedule and check its serializibility : 6

Transaction T_1	Transaction T_2	Transaction T_3

	read_item(Z);
	read_item(Y);
	write_item(Y);	

		read_item(Y); read_item(Z);
read_item(X); write_item(X);		
		write_item(Y); write_item(Z);
	read_item(X);	
read_item(Y); write_item(Y);		
	write_item(X);	

★ ★ ★