SIES Graduate School of Technology (Autonomous) RISE WITH EDUCATION NAAC A+ (Affiliated to University of Mumbai)	***End Semester Examination (R-24/R-25) SH 2026 Answer Key with marking scheme***			
Branch: ECS	Course: DBMS			
Year/ Semester: 2/III	Course code:ECC303			
Time: 03 hours	Marks: 80			
		Marks		
Q. 1	All questions are compulsory.			
A.	Explain : 3M. Example: 2M			
B.	Point: 5-6 : 5M			
C.	Define : 2M. Example: 2M Syntax:1M			
D.	Explain : 3M. Example: 2M			
E.	Limitations:5 : 5M			
F.	Explain : 3M. Example: 2M			
Q.2 A	Explain : 7M. Example: 3M			
	OR			
Q.2 B.	Explain : 5M. Example: 2M Diagram: 3M			
Q.3 A	Explain : 7M. Example: 3M			
	OR			
Q.3 B	-- 1. Modify the database so that 'John' now lives in 'Mumbai' UPDATE Employee SET city = 'Mumbai' WHERE empname = 'John'; -- 2. Find all employees who joined in the month of October SELECT empname FROM Employee WHERE EXTRACT(MONTH FROM date_of_joining) = 10; -- 3. Give all employees of 'ABC Corporation' a 10% raise UPDATE Works SET salary = salary * 1.10 WHERE company_name = 'ABC Corporation'; -- 4. Find all employees who earn more than the average salary of their company SELECT W1.empname FROM Works W1 WHERE W1.salary > (SELECT AVG(W2.salary) FROM Works W2 WHERE W2.company_name = W1.company_name);			

	-- 5. List name of companies starting with letter "A" SELECT company_name FROM Company WHERE company_name LIKE 'A%';			
Q.4 A	1. Entity-Relationship (E-R) Diagram This diagram models the core components of a library system: **Book**, **Author**, **Member**, and the **Borrow** relationship.			
	2. Relational Schema Conversion Mapping the E-R diagram objects to tables involves creating relations for entities and foreign keys to track active relationships securely. **BOOK** ● **Columns: ISBN** (PK), Title, Price, *Author_ID* (FK) ● *Note:* Since a book is written by one author (1:M relationship layout here), Author_ID is embedded directly as a foreign key inside the BOOK table. **AUTHOR** ● **Columns: Author_ID** (PK), Name **MEMBER** ● **Columns: Member_ID** (PK), Name, Address **BORROWS (Many-to-Many Bridge Table)** ● **Columns: *ISBN*** (PK, FK), ***Member_ID*** (PK, FK), Due_Date ● *Constraints:* ○ ISBN references BOOK(ISBN) ○ Member_ID references MEMBER(Member_ID)			
	OR			
Q.4 B	========== EER DIAGRAM COMPONENTS FOR HOSPITAL MANAGEMENT SYSTEM ========== 1. SUPERCLASS ENTITY: EMPLOYEE - Attributes: Emp_ID (Primary Key), Name, Job_Type 2. SUBCLASS ENTITIES (Disjoint Specialization: 'd'): - DOCTOR: Inherits from EMPLOYEE. Additional Attribute: Specialty - NURSE: Inherits from EMPLOYEE. Additional Attribute: Shift 3. CORE ENTITIES: - PATIENT: Attributes: Patient_ID (Primary Key), Patient_Name, Medical_History - ROOM: Attributes: Room_No (Primary Key), Room_Type 4. RELATIONSHIPS: - ASSIGNED (1:M between ROOM and PATIENT) * A Room can hold Multiple Patients; a Patient is assigned to One Room. - TREATS (M:N between DOCTOR and PATIENT)			

* A Doctor treats Multiple Patients; a Patient can be treated by Multiple Doctors.
* Relationship Attribute: Treatment_Date

==========

MAPPED RELATIONAL SCHEMA

==========

```
-- 1. EMPLOYEE TABLE
EMPLOYEE (
   Emp_ID [PK],
   Name,
   Job_Type
)

-- 2. DOCTOR TABLE (Subclass table linked via Foreign Key)
DOCTOR (
   Emp_ID [PK, FK references EMPLOYEE(Emp_ID)],
   Specialty
)

-- 3. NURSE TABLE (Subclass table linked via Foreign Key)
NURSE (
   Emp_ID [PK, FK references EMPLOYEE(Emp_ID)],
   Shift
)

-- 4. ROOM TABLE
ROOM (
   Room_No [PK],
   Room_Type
)

-- 5. PATIENT TABLE (Captures the 1:M "ASSIGNED" relationship via Room_No)
PATIENT (
   Patient_ID [PK],
   Patient_Name,
   Medical_History,
   Room_No [FK references ROOM(Room_No)]
)

-- 6. TREATS TABLE (Bridge Table capturing the M:N relationship)
TREATS (
   Doctor_ID [PK, FK references DOCTOR(Emp_ID)],
   Patient_ID [PK, FK references PATIENT(Patient_ID)],
   Treatment_Date
)
```

==========

Q.5 A	========== ========== DATABASE FLUID ANALYSIS: SYSTEM APPOINTMENT SCHEMA ========== ========== 1. PRIMARY DESIGN PITFALL: - UNDER-NORMALIZATION: The relation contains partial functional dependencies, violating Second Normal Form (2NF) criteria and causing data redundancy. - Core Functional Dependencies (FDs): * Patient_ID -> Patient_Name * Doctor_ID -> Doctor_Specialty * {Patient_ID, Doctor_ID, Appointment_Date} -> Entire Tuple (Composite PK) --- 2. OPERATIONAL ANOMALIES: - INSERTION ANOMALY: A new doctor cannot be registered in the system until they have scheduled at least one patient appointment, because key attributes cannot accept NULL values. - UPDATE ANOMALY: If a doctor changes their specialty, that modification must be executed across every historical appointment row for that doctor. Failure to update all rows leads to conflicting and corrupted data states. - DELETION ANOMALY: Deleting a canceled appointment record that happens to be the only appointment for a particular doctor inadvertently purges that doctor's ID and specialty entirely from the system. --- 3. DECOMPOSED RELATIONAL SCHEMA (3NF): -- Table to store patient profiles uniquely PATIENTS (Patient_ID [PK], Patient_Name) -- Table to store doctor profiles uniquely DOCTORS (Doctor_ID [PK], Doctor_Specialty) -- Transaction table capturing individual appointment operations APPOINTMENTS (Patient_ID [PK, FK references PATIENTS(Patient_ID)], Doctor_ID [PK, FK references DOCTORS(Doctor_ID)],			

	Appointment_Date [PK]) ========== ==========			

	OR			
Q.5 B	Need for normalization : 4M. Explain 2NF,3NF and BCNF with examples. : 6M			
Q.6 A	Explain : 7M. Example: 3M			
	OR			
Q.6 B	Explain : 7M. Example: 3M			
	*********** All the Best***********			