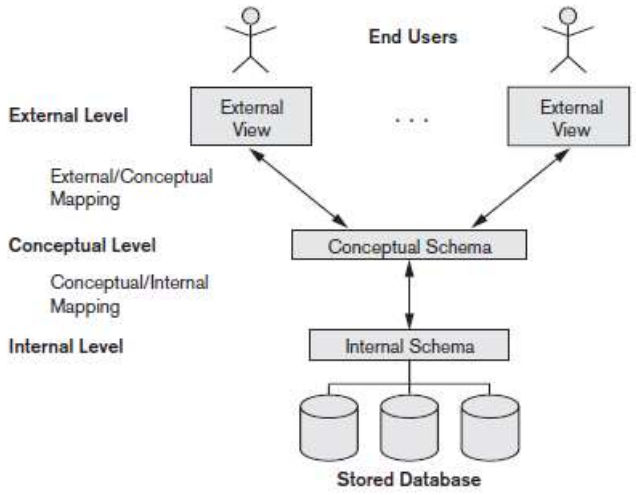
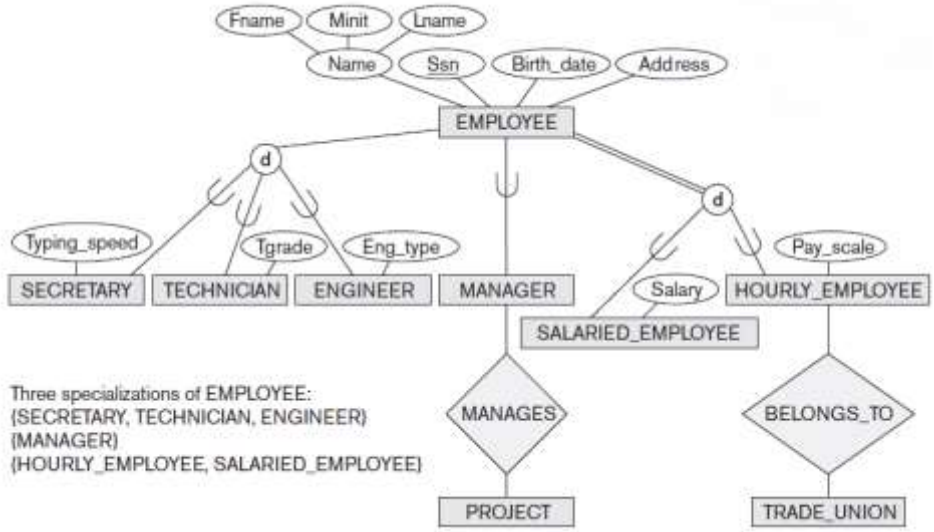
UNIVERSITY SOLUTION 2020-21

SUBJECT : DATABASE MANAGEMENT SYSTEM

SUBJECT CODE: CSC502
CLASS: T.E
SEMESTER: V

SUBJECT INCHARGE: PRIYA KAUL

Q2	Solve any Four out of Six 5 marks each	Rubric
A	Discuss the roles of DBA	
Sol.	**Database Administrator** One of the main reasons for using DBMSs is to have central control of both the data and the programs that access those data. A person who has such central control over the system is called a **database administrator** (**DBA**). The functions of a DBA include: **1. Schema definition**. The DBA creates the original database schema by executing a set of data definition statements in the DDL. **2. Storage structure and access-method definition**. The DBA decides how the data is to be represented in the stored database. **3. Schema and physical-organization modification**. The DBA carries out changes to the schema and physical organization to reflect the changing needs of the organization, or to alter the physical organization to improve performance. **4. Granting of authorization for data access**. By granting different types of authorization, the database administrator can regulate which parts of the database various users can access. The authorization information is kept in a special system structure that the database system consults whenever someone attempts to access the data in the system. **5. Routine maintenance**. Examples of the database administrator's routine maintenance activities are: □ Periodically backing up the database, either onto tapes or onto remote servers, to prevent loss of data in case of disasters such as flooding. □ Ensuring that enough free disk space is available for normal operations, and upgrading disk space as required. □ Monitoring jobs running on the database and ensuring that performance is not degraded by very expensive tasks submitted by some users. **6. Backup and Recovery-** Database should not be lost or damaged. The DBA ensures this periodically backing up the database on magnetic tapes or remote servers. In case of failure, such as virus attack database is recovered from this backup. **7. Specifying integrity constraints** – DBA is responsible for deciding and implementing entity integrity, domain integrity and referential integrity constraints.	4.5 – all functions except backup and recovery 3.5 – all functions except authorization and backup/recovery 2- only listing no explanation
B	Explain data independence and discuss types of data independence	
Sol.	The three-schema architecture of DBMS can be used to further explain the concept of **data independence**, which can be defined as the capacity to change the schema at one level of a database system without having to change the schema at the next higher level.We can define two types of data independence: **1. Logical data independence** is the capacity to change the conceptual schema without having to change external schemas or application programs. We may change the conceptual schema to expand the database (by adding a record type or data item), to change constraints, or to reduce the database (by removing a record type or data item). In the last case, external schemas that refer only to the remaining data should not be affected. For example, the external schema of should not be affected by changing the GRADE_REPORT file (or record type). Only the view definition and the mappings need to be changed in a DBMS that supports logical data independence. After the conceptual schema undergoes a logical reorganization, application programs that reference the external schema constructs must work as before. Changes to constraints can be applied to the conceptual schema without affecting the external schemas or application programs.	5- explaining the two types with examples and diagram 4 - explaining the concept with examples 3 - explaining

	 2. **Physical data independence** is the capacity to change the internal schema without having to change the conceptual schema. Hence, the external schemas need not be changed as well. Changes to the internal schema may be needed because some physical files were reorganized—for example, by creating additional access structures—to improve the performance of retrieval or update. If the same data as before remains in the database, we should not have to change the conceptual schema. Generally, physical data independence exists in most databases and file environments where physical details such as the exact location of data on disk, and hardware details of storage encoding, placement, compression, splitting, merging of records, and so on are hidden from the user. Applications remain unaware of these details. On the other hand, logical data independence is harder to achieve because it allows structural and constraint changes without affecting application programs—a much stricter requirement.	the concept without example
C	Explain Specialization and Generalization in EER with example	
Sol.	**Specialization** is the process of defining a *set of subclasses* of an entity type; this entity type is called the **superclass** of the specialization. The set of subclasses that forms a specialization is defined on the basis of some distinguishing characteristic of the entities in the superclass. For example, the set of subclasses {SECRETARY, ENGINEER, TECHNICIAN} is a specialization of the superclass EMPLOYEE that distinguishes among employee entities based on the job type of each employee entity. We may have several specializations of the same entity type based on different distinguishing characteristics. For example, another specialization of the EMPLOYEE entity type may yield the set of subclasses {SALARIED_EMPLOYEE, HOURLY_EMPLOYEE}; this specialization distinguishes among employees based on the method of pay. 	5- explain two concepts with proper diarams, and showing attributes of all entities 4.5 explain two concepts with proper diagrams 3- only explanation, and no proper examples

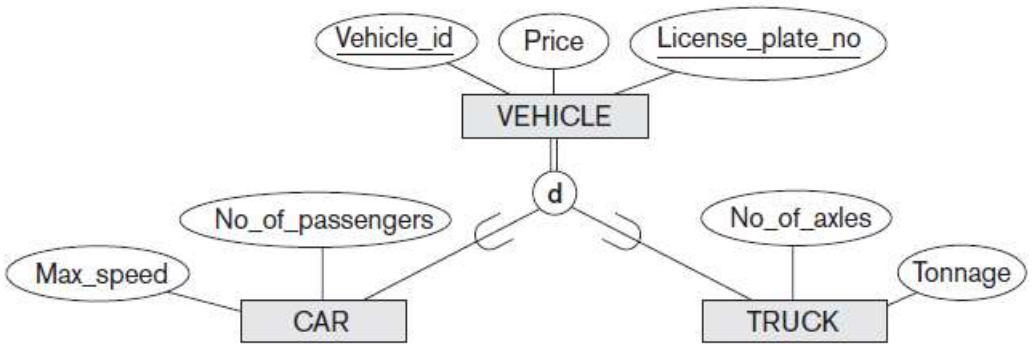
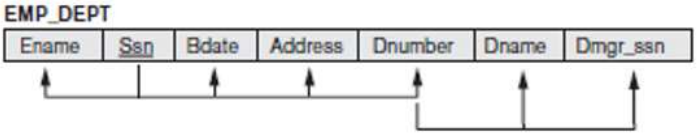
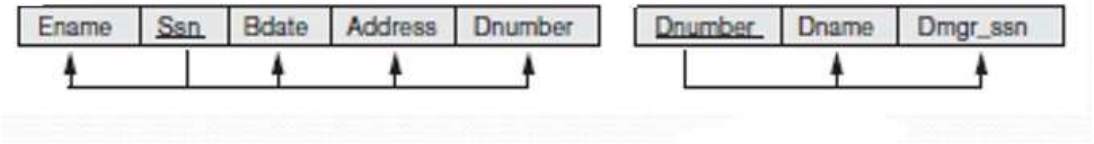
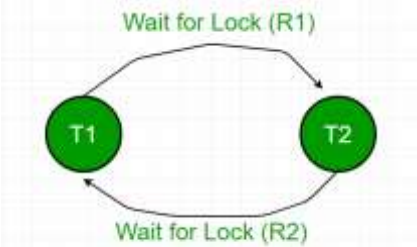
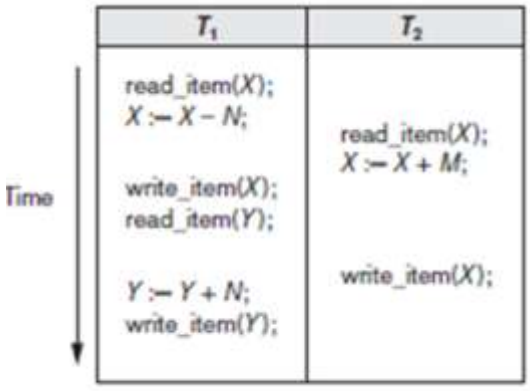
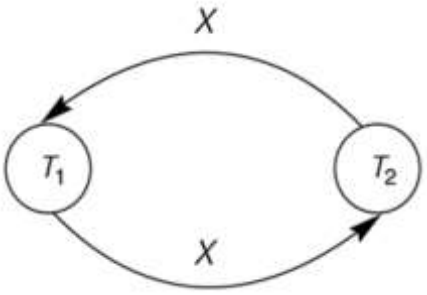
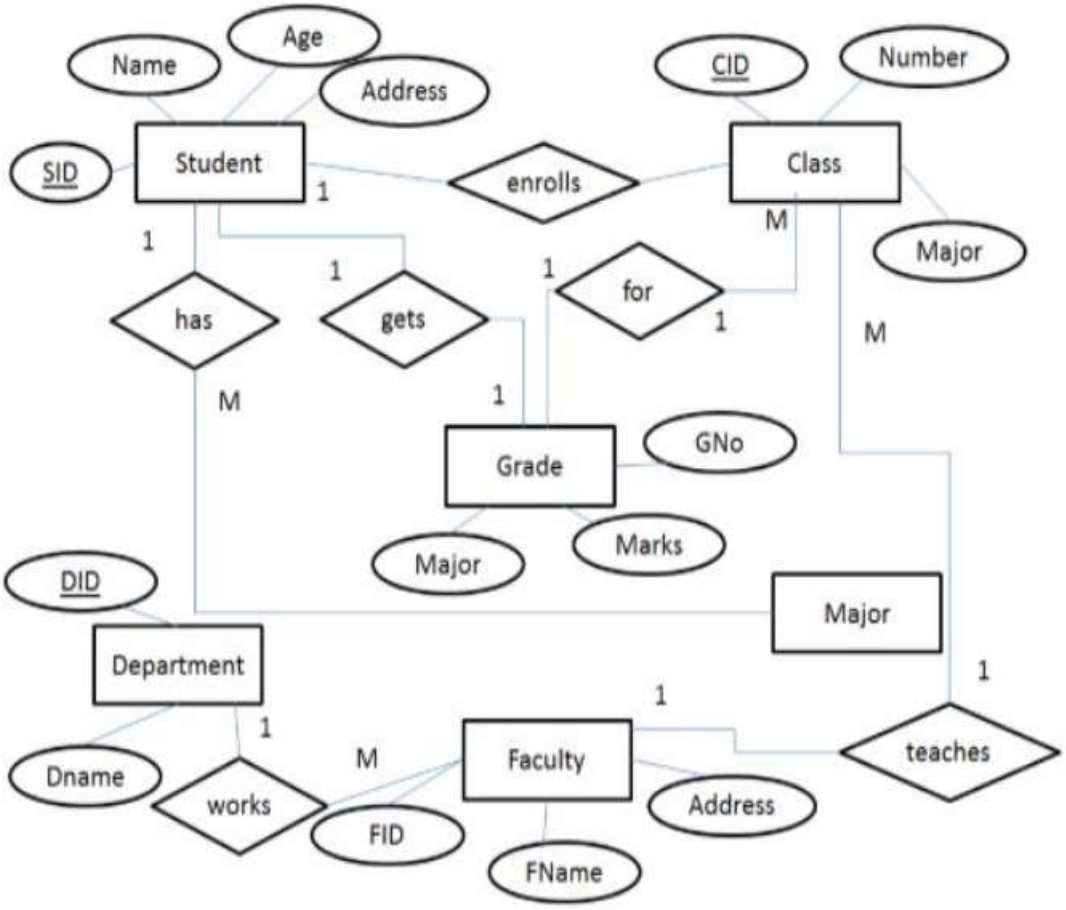
	Generalization is the reverse process of abstraction in which we suppress the differences among several entity types, identify their common features, and generalize them into a single superclass of which the original entity types are special subclasses. For example, consider the entity types CAR and TRUCK. Because they have several common attributes, they can be generalized into the entity type VEHICLE. Both CAR and TRUCK are now subclasses of the generalized superclass VEHICLE. We use the term generalization to refer to the process of defining a generalized entity type from the given entity types. Generalization process can be viewed as being functionally the inverse of the specialization process.  	
D	Explain different integrity constraints	
Sol.	1. Donain Integrity Constraints: NOT NULL, CHECK, DEFAULT a. NOT NULL- The NOT NULL constraint is a column constraint that ensures values stored in a column are not NULL. b. CHECK - The CHECK constraint is used to limit the value range that can be placed in a column. If we define a CHECK constraint on a single column it allows only certain values for this column. Example CREATE TABLE Persons (ID int NOT NULL, LastName varchar(255) NOT NULL, FirstName varchar(255), Age int, CHECK (Age>=18)); 2. Entity Integrity: Primary key, Unique Key a. Primary Key The PRIMARY KEY constraint uniquely identifies each record in a table. Primary keys must contain UNIQUE values, and cannot contain NULL values. b. Unique Key A unique constraint is a single field or combination of fields that uniquely defines a record. Some of the fields can contain null values as long as the combination of values is unique. Example CREATE TABLE Persons (ID int NOT NULL, LastName varchar(255) NOT NULL, FirstName varchar(255), Age int, PRIMARY KEY (ID)); 3. Referential Integrity: Foreign key A FOREIGN KEY is a key used to link two tables together. A FOREIGN KEY is a field (or collection of fields) in one table that refers to the PRIMARY KEY in another	5- listing all constraints, eplaining and example 3.5 – listing all, giving examples of only some, not all 2- partially correct listing with example

	table.The table containing the foreign key is called the child table, and the table containing the candidate key is called the referenced or parent table. A FOREIGN KEY is a key used to link two tables together. CREATE TABLE Orders (OrderID int NOT NULL, OrderNumber int NOT NULL, PersonID int, PRIMARY KEY (OrderID), FOREIGN KEY (PersonID) REFERENCES Persons(PersonID));	
E	Discuss the need of Normalization in Database design. Explain 3NF.	
Sol.	**Need of Normalization:** Normalization is the process of decomposing unsatisfactory "bad" relations by breaking up their attributes into smaller relations. Following the creation of a database, normalization is the next key step, as this process removes any type of error, anomaly or redundancy that might exist in the design of tables and in the links between different sources of information. Normalization is needed to avoid data being replicated in various tables at the same time or unrelated product data being gathered together in the same table. In addition, this technique makes it possible to make your databases more logical and natural, reducing their size and simplifying the structure to make data easier to locate, contrast and retrieve. Certain anomalies that can be solved with the help of Normalization are explained as under: **Insert Anomaly:** Consider the relation: EMP_PROJ(Emp#, Proj#, Ename, Pname, No_hours) We cannot insert a project unless an employee is assigned to it. Conversely we cannot insert an employee unless an he/she is assigned to a project. **Delete Anomaly:** When a project is deleted, it will result in deleting all the employees who work on that project. Alternately, if an employee is the sole employee on a project, deleting that employee would result in deleting the corresponding project. **Update Anomaly:** Changing the name of project number P1 from "Billing" to "Customer-Accounting" may cause this update to be made for all 100 employees working on project P1. **Third Normal Form:** A relation schema R is in **third normal form (3NF)** if it is in 2NF *and* no non-prime attribute A in R is transitively dependent on the primary key Given is the table that is to be converted into 3NF:  Given Functional dependencies are: Ssn -> Ename, Bdate, Address, Dnumber Dnumber -> Dname, Dmgr_ssn Step1: check if in 2 NF Given table is already in 2NF	2.5- explain normalization and anomalies 2.5 – third normal form with example

	Step2: check for transitive dependencies Ssn -> Dnumber -> Dname , therefore Ssn -> Dname is a transitive FD Ssn -> Dnumber -> Dmgr_Ssn, , therefore Ssn -> Dmgr_Ssn is a transitive FD Step 3 Perform decomposition to get final tables in 3NF – Emp1 and Dep1 Emp1 Dep1 	
F	Explain deadlock with wait-for graph	
Sol.	In a database, a deadlock is an unwanted situation in which two or more transactions are waiting indefinitely for one another to give up locks. Example – let us understand the concept of Deadlock with an example : Suppose, Transaction T1 holds a lock on some rows in the Students table and needs to update some rows in the Grades table. Simultaneously, Transaction T2 holds locks on those very rows (Which T1 needs to update) in the Grades table but needs to update the rows in the Student table held by Transaction T1. Now, the main problem arises. Transaction T1 will wait for transaction T2 to give up lock, and similarly transaction T2 will wait for transaction T1 to give up lock. As a consequence, All activity comes to a halt and remains at a standstill forever unless the DBMS detects the deadlock and aborts one of the transactions.  Deadlock Situation Example: Consider the following schedule. With The help of Wait for graph it can be detected that this schedule will lead to a dedlock:  Schedule C Wait-for graph for is a graph with directed edges where an edge is created from Ti to Tj if one of the operations in Ti appears before a conflicting operation in Tj.	2.5 – deadlock explianed, wait for graph explained 2.5- Example of a schedule with wait for graph

	 As wait for graph of the schedule has a cycle. Hence it can be deduced that this schedule will result in a deadlock	

Q3	Solve any Two Questions out of Three 10 marks each	Rubrics
A	Draw an E-R diagram for University database consisting of entities Student, Faculty, Department, Class. A student has a Unique id, the student can enroll for multiple classes and has at most one major. Faculty must belong to department and faculty can take multiple classes. Every student will get a grade for the class he/she was enrolled. Convert E-R diagram into relational schema.	
Sol.	**ER diagram**  **Relational Schema:** 1. Student(Sid, Name, Age, Address) 2. Class(CID, Nmumber, Major, Sid, Gno, Fid) 3. Grade(Gno,Major, Marks) 4. Faculty(Fid, Fname, Address, Did)	**6+4**: ER diagram and Relational modal ER diagram: - relationship - primary key Other attributes Relational modal: - tables for all entities with correct primary and foreign keys **6.5** – correct ER and incorrect

	5. Department(Did, Dname)	Rlational modal **5**- Only correct ER **Less than 5** – incorrect ER, no relational modal
B	Consider the employee database *employee (employeename, street, city,date of join)* *works (employeename, company name, salary)* *company (company name, city)* *manages (employee name, manager name)* Write SQL queries for the following statements 1) Find all the employees who joined in the month of october SELECT employeename FROM employee WHERE to_char(date of join,'mon')='oct'; 2) Modify the database so that 'Anjali' now lives in 'Mumbai' UPDATE employee SET city="Mumbai" WHERE employeename="Anjali" 3) List all the employees who live in the same cities as their managers. SELECT e.employeename FROM employee e, works w, company c WHERE e.employe ename = w.employeename AND e.city = c.city AND w.companyname = c.companyname 4) Find all employees who earn more than the average salary of all the employees of their company SELECT employeename FROM works T where salary > (SELECT AVG(salary) FROM works S WHERE T.companyname = S.companyname) 5) Give all the employees of ABC corporation a 15 percent raise. UPDATE works SET salary = salary * 1.1 WHERE companyname = 'ABC' ;	2 marks each- Correct SQL queries 1 mark – for partially correct query
C	Explain any two concurrency control protocols in database system	
Sol.	Students can explain any of these protocols: 1. Lock based protocol 2. Time-stamp protocol 3. Validation based protocol	

1. Two-Phase Locking Techniques: Shared/ Exclusive

Two locks modes:

(a) shared (read) (b) exclusive (write).

Shared mode: shared lock (X)

More than one transaction can apply share lock on X for reading its value but no write lock can be applied on X by any other transaction.

Exclusive mode: Write lock (X)

Only one write lock on X can exist at any time and no shared lock can be applied by any other transaction on X.

3 OPEARTIONS- Read, Write, Unlock

The following code performs the read operation:

```
read_lock(X)
        B: if LOCK (X) = "unlocked" then
begin LOCK (X)  "read-locked";
        no_of_reads (X)  1;
end
else if LOCK (X)  "read-locked" then
            no_of_reads (X)  no_of_reads (X) +1
         else begin wait (until LOCK (X) = "unlocked" and
                    the lock manager wakes up the transaction);
                    go to B
                end;
```

The following code performs the write operation:

```
write_lock(X)
        B: if LOCK (X) = "unlocked" then
begin LOCK (X)  "write-locked";
                  else begin wait (until LOCK (X) = "unlocked" and
                    the lock manager wakes up the transaction);
                    go to B
                end;
```

The following code performs the unlock operation:

```
unlock(X):
         if LOCK (X) = "write-locked" then
begin LOCK (X)  "unlocked";
         wakes up one of the transactions, if any
end
else if LOCK (X)  "read-locked" then
        begin
            no_of_reads (X)  no_of_reads (X) -1
            if  no_of_reads (X) = 0 then
              begin
                    LOCK (X) = "unlocked";
                    wake up one of the transactions, if any
             end
        end;
```

Lock conversion:

Lock upgrade: existing read lock to write lock

```
                if Ti has a read-lock (X) and Tj has no read-lock (X) (i  j) then
            convert read-lock (X) to write-lock (X)
                else
            force Ti to wait until Tj unlocks X
```

Lock downgrade: existing write lock to read lock

```
                Ti has a write-lock (X)    (*no transaction can have any lock on X*)
```

5+5 : Two phase locking: Explain different locks, and 2 phase shrinking, releasing mode. Explain timestamp protocol with example

2.5+2.5 - Only explain diferent locks. Explain timestamp protocol without example

convert write-lock (X) to read-lock (X)

2. Timestamp based prevention protocols

Suppose that transaction Ti tries to lock an item X but is not able to because X is locked by some other transaction Tj with a conflicting lock. The rules followed by these schemes are:

a) **Wait-die. If TS(Ti) < TS(Tj), then (Ti older than Tj) Ti is allowed to wait;** otherwise (Ti younger than Tj) abort Ti (Ti dies) and restart it later with the same timestamp.

b) **Wound-wait. If TS(Ti) < TS(Tj), then (Ti older than Tj) abort Tj (Ti wounds** Tj) and restart it later with the same timestamp; otherwise (Ti younger than Tj) Ti is allowed to wait.

Example:

Suppose Timestamp(Ti)<Timestamp(Tj). It means Ti is older txn and Tj is younger txn.

Assume that transaction Ti has locked data item X and Tj is requesting lock on X in conflicting mode.

As per Wait-die aproach, Tj dies i.e. it is aborted and restarted later with the same timestamp.

As per Wound-Wait approacj, Tj simply waits till Ti releases the lock on dataitem X.