CBCS SCHEME

USN 2 V D 2 3 C I 0 0 7

BCS403

Fourth Semester B.E./B.Tech. Degree Examination, June/July 2025

Database Management Systems

Time: 3 hrs. Max. Marks: 100

Note: 1. Answer any FIVE full questions, choosing ONE full question from each module.
2. M : Marks , L: Bloom's level , C: Course outcomes.

		Module – 1	M	L	C
Q.1	a.	Explain the types of attributes with example.	4	L2	CO1
	b.	Define database. Explain the main characteristics of the database approach.	8	L2	CO1
	c.	Show the ER diagram for an EMPLOYEE database by assuming your own entities (minimum 4) attributes and relationships, mention cardinality ratios wherever appropriate.	8	L3	CO2
		OR			
Q.2	a.	Describe the three schema architecture.	4	L2	CO1
	b.	Explain the component models of DBMS and their interaction with the help of diagram.	8	L2	CO1
	c.	Design ER diagram for a university database by assuming your own entities (4). Mention primary key , constraints and relationships.	8	L3	CO2
		Module – 2			
Q.3	a.	Explain relational model constraints.	6	L2	CO1
	b.	Explain the characteristics of relations with suitable example for each.	6	L2	CO1
	c.	Considering the following schema : Sailors (sid , sname , rating , age) Boats (bid , bname , color) Reserves (sid , bid , day) Write a relational algebra queries for the following : i) Find the names of sailors, who have reserved red and a green boat. ii) Find the names of sailors who have reserved a red boat. iii) Find the names of sailors who have reserved a red or green boat. iv) Find the names of sailors who have reserved all boats.	8	L3	CO1
		OR			
Q.4	a.	Explain the steps to convert the basic ER model to relational Database schema.	6	L2	CO1
	b.	Explain Unary relational operations with example.	6	L2	CO1

<table>
<tr><td></td><td>c.</td><td>Consider the relation schema Employee database.
EMPLOYEE (Fname ,Minit , Lname , SSn , Bdates , Address , Sex , Salary Super_SSn , Dno)
DEPARTMENT (Dname , Dnumber , Mgr_SSn , Mgr_start_date)
PROJECT (Pname , PNumber , Plocation , Dnum)
WORKS_ON (Essn , Pno , Hours)
DEPENDENT (Essn , Dependent_name , sex, Bdate , Relationship)
Write relational algebra queries for the following .
i) Retrieve the name and address of all employees who work for the 'Research' department.
ii) List the names of all employees with 2 or more dependents.
iii) Find the names of employees who work on all the projects controlled by department number 5.
iv) List the names of employees who have no dependents.</td><td>8</td><td>L3</td><td>CO3</td></tr>
<tr><td colspan="6">Module – 3</td></tr>
<tr><td>Q.5</td><td>a.</td><td>What is the need for normalization? Explain second and third normal form with examples.</td><td>6</td><td>L2</td><td>CO4</td></tr>
<tr><td></td><td>b.</td><td>Outline constraints in SQL.</td><td>6</td><td>L2</td><td>CO1</td></tr>
<tr><td></td><td>c.</td><td>Identify the given Relation R(ABCDE) and its instance, check whether FDS given hold or not. Give reasons
i) A → B ii) B → C iii) D → E iv) CD → E.<table>
<tr><th>A</th><th>B</th><th>C</th><th>D</th><th>E</th></tr>
<tr><td>a_1</td><td>b_1</td><td>c_1</td><td>d_1</td><td>e_1</td></tr>
<tr><td>a_1</td><td>b_2</td><td>c_1</td><td>d_1</td><td>e_1</td></tr>
<tr><td>a_2</td><td>b_2</td><td>c_1</td><td>d_2</td><td>e_3</td></tr>
<tr><td>a_2</td><td>b_3</td><td>c_3</td><td>d_2</td><td>e_2</td></tr>
</table></td><td>8</td><td>L3</td><td>CO4</td></tr>
<tr><td colspan="6">OR</td></tr>
<tr><td>Q.6</td><td>a.</td><td>What is Multivalued dependency? Explain 4NF and 5NF with suitable example.</td><td>6</td><td>L2</td><td>CO4</td></tr>
<tr><td></td><td>b.</td><td>Outline the informal design guidelines for relational schema.</td><td>6</td><td>L2</td><td>CO4</td></tr>
<tr><td></td><td>c.</td><td>Consider relation R with following function dependency :
EMPPROJ (<u>SSn</u> , <u>Pnumber</u> , Hours , Ename , Pname , Plocation)
SSN , Pnumber → Hours,
SSN → Ename
Pnumber → Pname , Plocation.
Is it 2NF? Verify? If no give reason.</td><td>8</td><td>L3</td><td>CO4</td></tr>
</table>

		Module – 4			
Q.7	**a.**	Consider the following schema for a company database : Employee (FName , LName , SSn , Adderss , Sex , Salary , Dno , Super_SSn) Department (Dname , Dnumber , mgr_SSn, mgr_st_date) Project (Pname , Pnumber , Plocation , Dnum) WORKS_on (Essn , Pno , Hours) DEPENDENT (Essn , Dependent name , Sex , Bdate, relationship) Write the SQL queries for the following : i) List the names of managers who have atleast one dependent (use correlated nested). ii) Retrieve the name of each employee who has a dependent with the same first name and is the same sex as the employee. iii) For each project retrieve the project number , project name and the number of employees who work on that project. iv) Retrieve the SSN of all employees who work on project number 1, 2 or 3. (Use IN). v) Find the sum of the salaries of all employees of the 'Research' department as well as maximum salary , minimum salary , average salary in this department.	10	L3	CO3
	b.	Why concurrency control is needed? Demonstrate with an example.	10	L2	CO5
		OR			
Q.8	**a.**	Consider the following schedule. The actions are listed in the order they are scheduled and prefixed with the transaction name. S1 : T1 : R(X) , T2 : R(X) T1 : W(Y) , T2 : W(Y) , T1 : R(Y) , T2 : R(Y) S2 : T3 : W(X) , T1 : R(X) , T1 : W(Y) , T2 : R(Z) , T2 : W(Z) , T3 : R(Z) For each schedule answer the following : i) What is the precedence graph for the schedule? ii) Is the schedule conflict_serializable? If so what are all the conflicts equivalent serial schedules? iii) Is the schedule view serializable? If so what are all the view equivalent serial schedules?	10	L3	CO5
	b.	Explain triggers with example write a trigger in SQL to call a procedure "Inform_Supervisor" whenever an employees salary is greater than the salary of his or her direct supervisor in the COMPANY database.	10	L3	CO5
		Module – 5			
Q.9	**a.**	Describe the two – phase locking protocol for concurrency control provide example to illustrate how it ensures serializability in transaction schedule.	10	L2	CO5
	b.	Explain the characteristics of NOSQL system.	10	L2	CO6
		OR			
Q.10	**a.**	Explain binary locks and shared lock with algorithm.	10	L2	CO5
	b.	Explain MongoDB data model, CRUD operations and distributed system characteristics.	10	L2	CO6

* * * * *

Q.1
b> Define database. Explain the main charecterstics of database approach.

soln. A data base is a collection of related data.
The main charecterstics of the data base application

1> Self describing nature of the data base system.
-> data base system stores both data and its complete definition.
-> DBMS catalog contain structure of each file, Name of the data item, data types, storage formates, size position of each data item, constraints on the data item.

2> Insulation between programs and data and data-abstraction.
-> DBMS based system store the structure of data files seperatly in the DBMS catalog.
-> This seperation between program and data definition is called program data independent.

3> Support multiple views of the data.
-> A data base system typically stores multiple types of users with different roles and requirments
-> Each user may need a different perspective of the same database.
-> A view is a customization representation of the database to a specific user or applicaion.

4> sharing data and multiuser transaction
-> Multiple users can access the same database simultaniously
-> A DBMS include concurrancy control mechanism to manage simultaneous access to a shared data.

Q. 1
c] show a ER diagram for an EMPLOYEE database by assumming own entities, attributes, relationships, mention cordinality ratios, wherever appropriate.

Q 1

a) Explain the types of attribute with example.

① Composite attributes are those which can be divided into smaller subparts, which represent more basic attribute with independent meaning.

Ex: Address
↳ street-address → → Number
→ street
→ Apartment number.
↳ City
↳ State
↳

② Simple attributes: are attributes that are drawn from an atomic value or attribute that are not divisible.

Ex: Apartment number, street.

③ Single valued attributes: which can take more than one value for a given entity from an entity set

Ex: A person who can speak in more than one language.

④ Multivalued attributes: which can take only one value for a given entity from an entity set

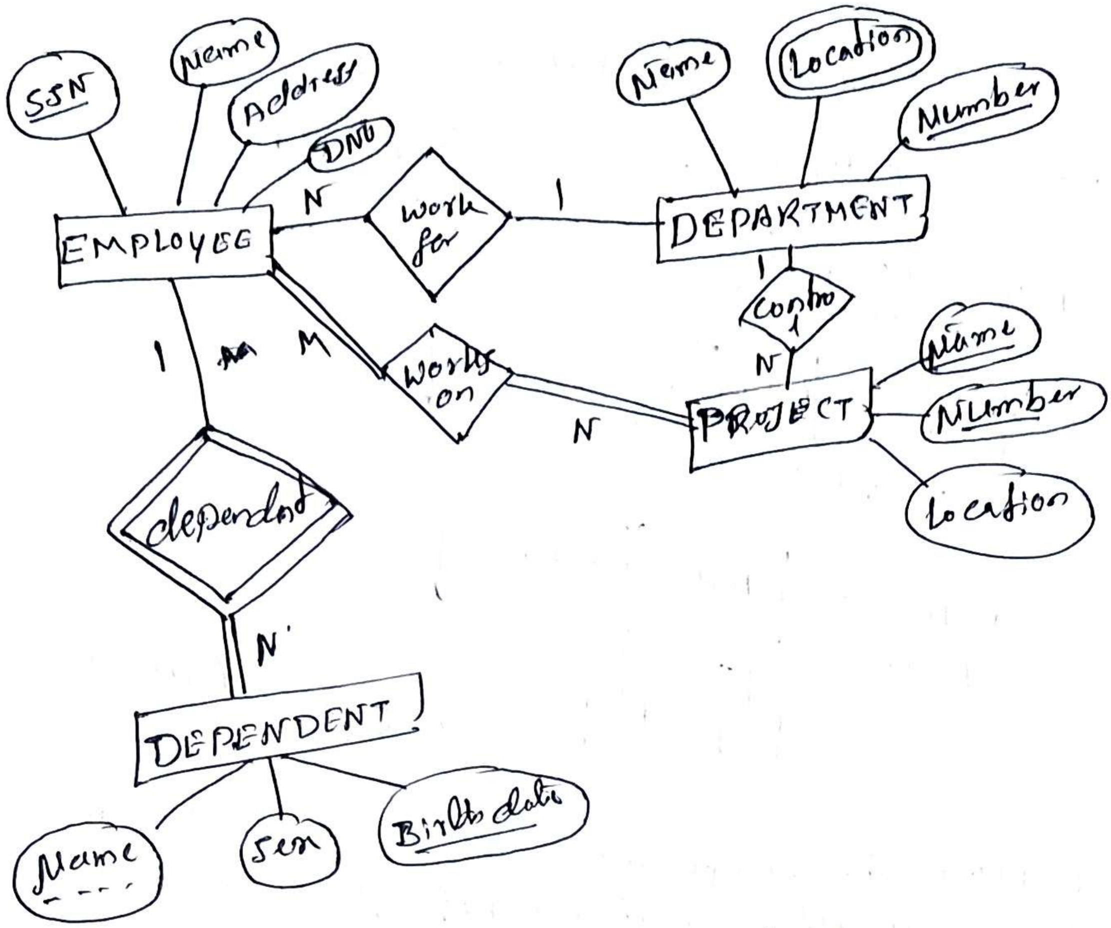
Ex: Age.

⑤ Derived Attribute: which can be derived from other attribute

Ex: Age of student is derived from date of birth.

⑥ Key attributes: which can identify an entity uniquely in a set.

Ex: Roll no. of a student.



ER diagram of employee database

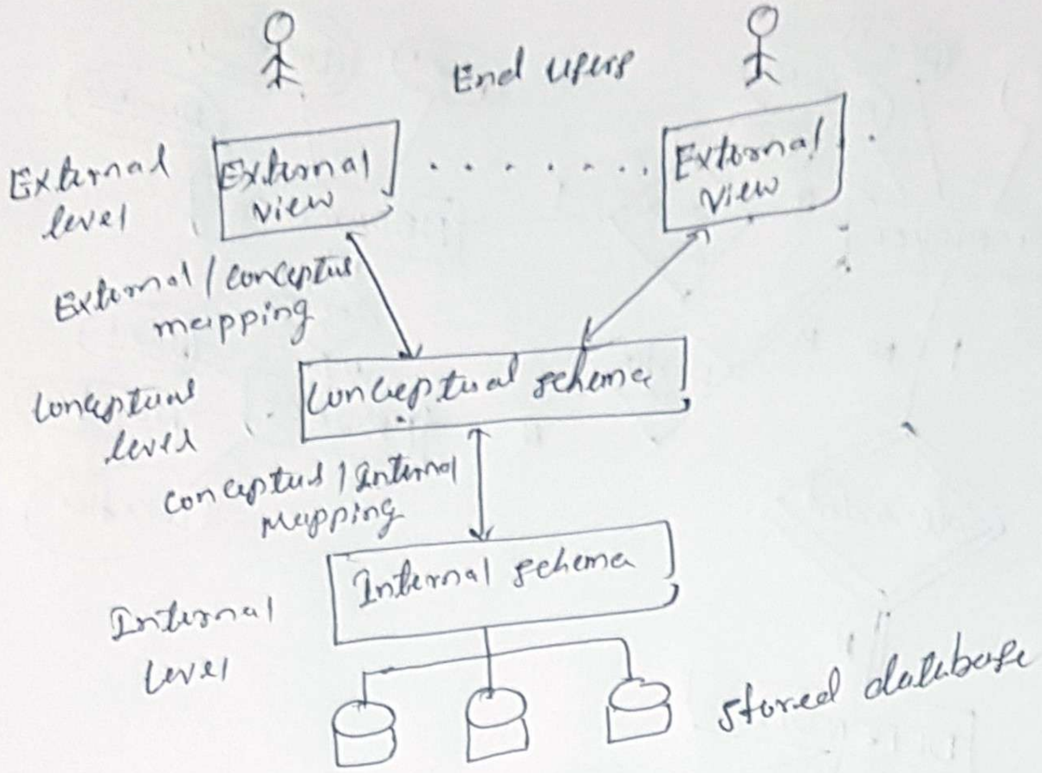
Q. 2
a) Describe the three schema architecture.

Soln The three schema architecture can be defined at the following three levels.

① The internal level has an internal schema, which describe the physical storage structure of the database.

② The conceptual level has a conceptual schema which describe the structure of the whole database for a community of users.
It hides the details of physical storage structure and describing entities, data types relationships. users constraints.

③ External view or view level include a number of external schemas or user views. Each external schema describes a part of the database, that a particular user group is intrested in & hide the rest of the database from the users.



Q 2.
(b) Explain the component models of DBMS by assuming and their interaction with the help of diagram.

Soln → A DBMS a collection of software modules that manage database creation, access and maintenance.

→ DBMS components are divided into two major parts.

→ User level components

→ Internal DBMS components

→ DBA controls and manage the entire database system that define database schema, storage structures, constraints access rights.

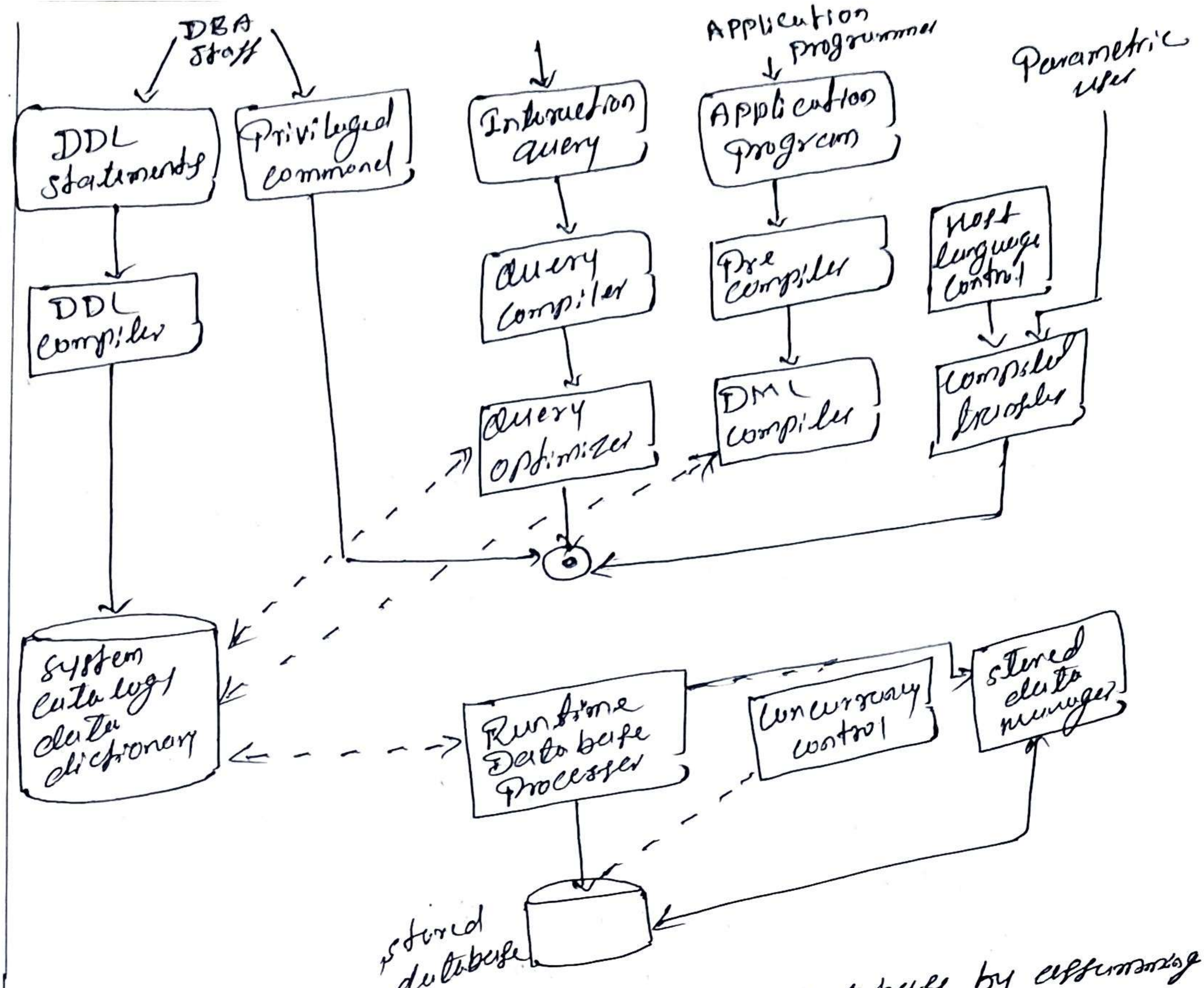
→ Casual users access the database using interactive query interfaces.

→ Application programmers write database programs using host languages ~~[illegible]~~ such as Java, C, Python.

→ parametric user execute predefined or command transaction

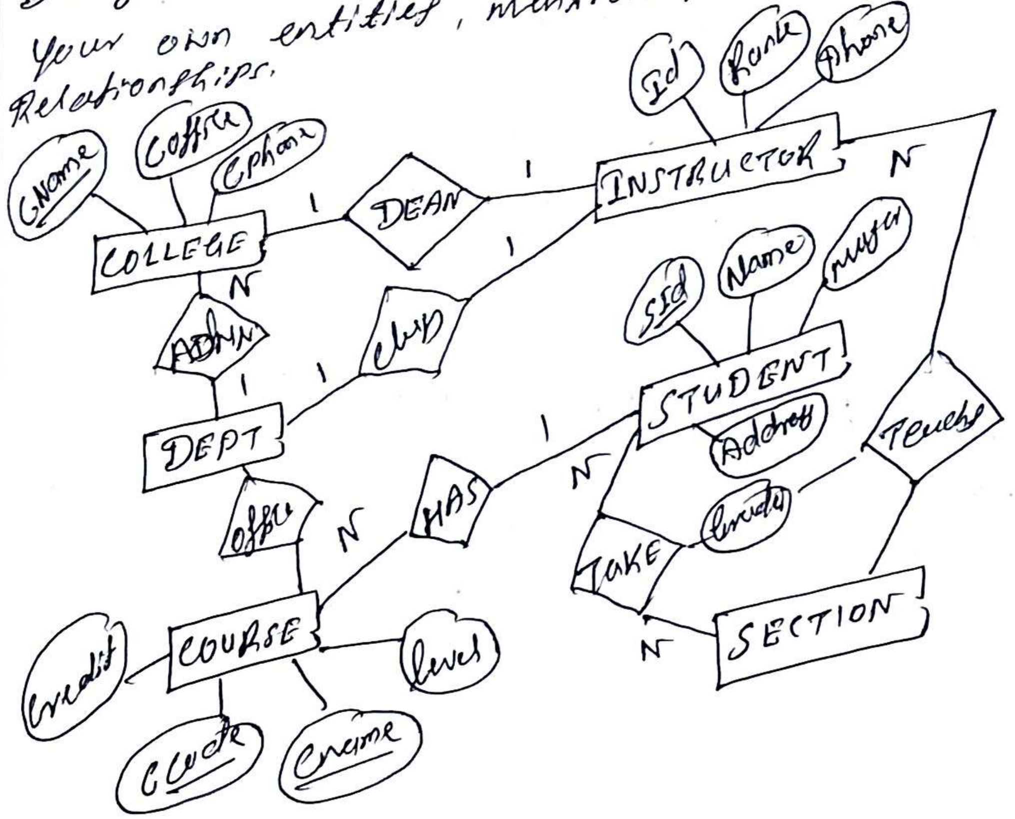
→ DDL compiler processes database schema definition.

→ The query compiler checks SQL query syntax and validates attribute and table names.



Q 2. c) Design ER diagram for a university database by assuming your own entities, mention Primary key, constraints and Relationships.

Sol.



Q3 a) soln

Explain Relational model constraints.

constraints on database can generally be divided into three main categories.

① constraints that are inherent in the data model called implicit constraints.

② constraints that can be directly expressed in the schemas of the data model called data definition language. called explicit constraints.

③ constraints that can't be directly expressed in the schema of the data model, called Application based or constraint rules.

* Domain constraints specify that each attribute value must be an atomic value from its predefined domain. It defines data types, value range, enumerated values, special types.

* Key constraints and constraints on null values.

→ A relation is called as a set of tuples, All elements - set are distinct. No two tuples have the same combination of values for all there attributes.

→ super key defines uniqueness constraints that no two distinct tuples in any state r of R can have the same value for SK.

i.e $t_1[SK] \neq t_2[SK]$

→ A key is a minimal super key with additional property that no two tuple share the same key values and disjoining any attribute remains uniqueness

→ A relation schema may have more than one key. each key is called a candidate key.

→ one of the candidate key is designated as primary key. of the relation.

→ other candidate keys are designated as unique keys.

b) Explain the characterstics of relationships with suitable example for each.

Soln

ⓐ Ordering of tuples in a relation: A relation is defined as a set of tuples. Elements in a relation do not have any particular order.

ⓑ Ordering of values within a tuples and an attribute definition of a relation.
A relation schema $R = \{A_1, A_2 \ldots A_n\}$ is a set of attributes and a relation state $r(R)$ a finite set of mappings $r = \{t_1, t_2, \ldots t_n\}$ where each tuple t_i is a mapping from R to D and D is the union of the attribute domain.

ⓒ Values and NULL in the tuples:- Each value in the tuple is an atomic value, that is it is not divisible into components.

ⓓ Interpretation of Relation:- The relation schema can be interpreted as a declation or a type of assertion.
Each tuple can be interpreted as a fact in the relation

c) Consider the following schema
Sailors (sid, sname, rating, age)
Boats (bid, bname, color)
Reserves (sid, bid, day)

① Find the name of the sailers, who have secured red and green boat.

$Tempred \leftarrow \pi_{sid}(\sigma_{color='red'}(Boats \bowtie Reserves))$

$Tempgreen \leftarrow \pi_{sid}(\sigma_{color='green'}(Boats \bowtie Reserves))$

$\pi_{sname}((Tempred \cap Tempgreen) \bowtie Sailers.$

② Find the name of sailers, who have reserved a red boat

$sailers \leftarrow sailors \bowtie Reserves$

$\pi_{sname}((\sigma_{color='red'} Boats) \bowtie sailres$

③ Find the name of sailors who have reserved a red or green boat.

Temp boats ← ($\sigma_{color='red'}$ Boats) ∪ ($\sigma_{color='green'}$ Boats)

π_{sname} (Temp boats ⋈ (Reserves ⋈ sailors)

④ Find the name of the sailors who have reserved all boats

Tempsids ← ($\pi_{sid, bid}$ Reserves) ÷ (π_{bid} Boats)

Result ← π_{sname} (Tempsids ⋈ sailors).

Q. 4 a) Explain the steps to convert the basic ER model to relational database schema.

Soln <u>Step 1</u>: mapping of regular Entity types:- For each regular entity type E in ER schema. Create a relation R. That include all the simple attribute of E. Choose one of the key attribute of E as the primary key for R

<u>Step 2</u>:- mapping of weak entity type. For each weak entity ~~type~~ ~~type~~ type W in the ER schema with owner entity type E. Create a relation R and include all simple attribute of W as attribute of R

<u>Step 3</u>:- Mapping of binary 1:1 mapping relation types. For each binary 1:1 relationship type R in the ER schema, identify the relations S and T that corresponds to the entity types participating in R

<u>Step 4</u>:- Mapping of the Binary 1:N Relationship type

① Foreign key approach:- For each ~~binary~~ regular binary 1:N relationship type R identify the relation S. that represent the participating entity type at the N side of the relationship type.

② Relationship relation approach: is to use the cross reference option of in the third option for binary 1:1 relationship

Step 5: Mapping of the binary M:N Relationship types:-
In the traditional relational model with no multivalued attributes. the only option for M:N relationship is the relationship relation option.

Step 6: Mapping of the multivalued Attributes
For each multivalued attribute A create a new relation R this relation R will includes an attribute corresponding to A plus the primary key attribute K. as a foreign key in R.

Step 7: Mapping of N-ary relationship types
For each n-ary relationship type R, where $n>2$ Create a foreign key attributes in S., the primary keys of the relations that represends the participating entity.

b) Explain unary relational operations with example
Soln
Unary select operation:-
* The select operation is used to choose a subset of the tuples from a relation that satisfies a selection condition.

* We can consider the select operation to select the tuple in a relation to only those tuples that satisfy the condition.

Ex:- To select the employee tuples where department is 4 or, those whose salary is greater then \$30,000

$\sigma_{Dno=4}$ (Employee)

$\sigma_{salary > 30000}$ (Employee)

* The Project operation: It selects certain columns from the table and discards the other ~~table~~ columns. If you are intrested in only certain attribute of a relation, we use the project operation to project the relation over these attributes.

Ex:- To list each employee first and last name and salary use the project operations as follows.

$\pi_{lname,\ fname,\ salary}$ (Employee).

6) Consider relation schema Employee database

Employee (Fname, Minit, Lastname, <u>SSN</u>, Bdate, Address, Sex, Salary
Super-ssn, Dno)

Department (Dname, Dnumber, mgr-ssn, mgr-start-date)

Project (Pname, Pnumber, Plocation, Dnum)

Works-on (Essn, Pno, hours)

Dependent (Essn, Dependent-name, Sex Bdate, Relationship)

write a relation queries for the following

(a) Retrive the name and address of all employee who work for the Research Department.

$$\Pi_{Fname,\ Lname,\ Address} (\sigma_{Dname = 'Research'} (Department) \bowtie_{Dnumber = D_{no}} Employee)$$

(b) List ~~att~~ the names of all employeer with 2 or more dependent

$$\Pi_{Fname,\ Lname} (Employee \bowtie_{SSN = Essn} \sigma_{Count} > 2 (\gamma_{Essn,\ count <} Department\_name) \rightarrow Count (Dependent))]$$

(c) Find the name of employee who works on all the project controlled by department no = 5

$$\Pi_{Fname,\ Lname} (Employee \bowtie_{SSN = Essn} (\Pi_{Essn,\ Pno} (Works\text{-}on) \div \Pi_{Pno} (\sigma_{Dnum = 5} (Project)))$$

(d) List the name of employees who have no dependents

$$\Pi_{Fname,\ Lname} (Employee) - \Pi_{Fname,\ Lname} (Employee \bowtie_{SSN = Essn} Department)$$

Q 5 a) what is the need for normalization? Explain 2nd and 3NF with examples

Soln Normalization is a systematic process of organizing in a relational database to reduce redundancy and improve data integrity. It provides database designers with the following.

* A formal framework for analyzing relational schemas based on there keys and on the functional dependencies among their attributes.

* A series of normal forms, test that can be carried out individual relation schemas so that the relational database can be managed to any desired degree.

2NF: A relation schema R is in 2NF if every non prime attribute A in R is fully functionally dependent on the primary key of R

It is based on the concept of full functional dependency

A functional dependency X→Y is full functional dependency if removal of any attribute A from X means that the dependency does not hold any more.

Ex. Enrollment (studentid, courseid, Sname, coursename)

Functional Dependency

studentid → student-name

courseid → course-name

conversion into 2NF

Student (studentid, student-name)

course (courseid, course-name)

enrollment (student-id, courseid)

third normal form:: a relation schema R is in 3NF, if it satisfies 2NF and no non prime attribute of R is transitivly dependent on the primary key.

Ex: Employee (Empid, Ename, deptid, dept-name)

conversion to 3NF

Employee (Empid, Emp-name, dept-id)

Dept (Dept-id, Dept-name)

b) Outline the constraints in sql.

① Specifing attribute constraints and attribute default.

→ A constraint NOT NULL may be specified if NULL is not permitted for a particular attribute.

→ Default value for an attribute by appending the clause DEFAULT < value > to an attribute definition.

② Specifying Key and Referencial integrity constraints:-

→ The primary key clause specifies one or more attribute that make up the primary key of a relation.

→ Unique key clause specifies alternate, known as candidate key if it is a single attribute.

→ Referential integrity is specified via the FOREIGN KEY clause

③ Giving names to constraints:-

A constraint name is used to identify a particular constraints in case the constraint must be dropped later or replaced with another constraints.

c) Consider relation R with following function dependency

EMPPROJ (<u>Ssn</u>, Pno, hours, Ename, Pname, Plocation)

Ssn, Pnumber → Hours

Ssn → Ename

Pnumber → Pname, Plocation. Is it in 2NF ? verify.

Soln (a) The value of an employees Ssn uniquely determines the employee name (Ename)

(b) The value of a projects numbers (Pnumber) uniquely determine the project name (Pname) & Plocation.

(c) A combination of SSN & Pnumber uniquely determines the number of ~~hr~~ hours the employee currently works on the project per work.

~~Therefore~~ Therefor the relation is NOT in 2nd normal form.

Q 7

a) i) List the name of managers who have atleast dependent

```
select E. Fname , E. Lname
from Employee e.
where   E. ssn in ( select D. mgr-ssn
                    from Department D
                    where Enists ( select * from
                                   Dependent DP
                                   ~~where DP. Essn = D. mgr(~~
                                   where DP. Essn = D. mgr-ssn ));
```

ii) Retrieve the name of each employee who has a dependent with the same first name & same sex

```
select E. name, E. lname
from Employee E
where Enists ( select *
               from Dependent D
               where D. Essn = E. ssn
               AND D. Dependent_name = E. fname
               AND D. sex = E. sex);
```

iii) For each project, retrieve project no, project name, and number of employee working on it.

```
select P. number, P. Pname, count ( W. Essn) AS No_of_Employees
from project P, works_on W
where P. Pno = W. Pno
Group by P. Pno, Pname;
```

iv) Retrieve the ssn of all employees who work on project no 1, 2 or 3

```
select Distinct Essn
from works_on
where Pno in (1, 2, 3);
```

v) Find sum min, mum, avg Salary of employees in Research dept.

Select sum (E.salary), MAX (E.salary), min (E.salary), AVG (E.salary)
from Employee E, department D
where E.Dno = D.Dnumber AND D.Dname = 'Research';

Q 7
b) Why concurrency is needed? Demonstrate with example.

Soln Several problems can occurs where concurrent transaction execute in an uncontrolled manner

① The lost update problem:- occurs where two transaction that acces the same database items have their operation interleaved in a way that works the value of some database item incorrect.

② The temporary update problem:- occurs when one transaction update a database items and then the transaction fails for some reason. mean while a the updated item is occurred by another transaction before it is changed back to its original value.

③ The incorrect summary problem:- If one transaction is calculating an aggrate summary function on a number of database items while other transactions are updating some value before they are updated and others after they are updated.

④ The unrepeatable Read pattern:- May occur, where a transaction T reads the same item twice and the item is changed by another transaction T' between the two reads, Hence T receives different values for its two reads of the same item.

Q.8>
a> Given schedules

S1
T1 : R(X)
T2 : R(X)
T1 : W(X)
T2 : W(X)
T1 : R(Y)
T1 : R(Y)

S2
T3 : W(X)
T1 : R(X)
T1 : W(Y)
T2 : R(Z)
T2 : W(Z)
T3 : R(Z)

i) what is the Precedance graph for the schedules?

$T_1 : W(X) \rightarrow T_2 : W(Y)$

$T_1 : W(Y) \rightarrow T_2 : R(Y)$ Graph : $T_1 \leftrightarrow T_2$

$T_2 : W(Y) \rightarrow T_1 : R(Y)$

ii) Is the schedule conflict serializable? If so what are the conflict equivalent serial schedules?

conflict operation

① $T_3 : W(X) \rightarrow T_1 : R(X)$

$\Rightarrow$ edge $T_3 \rightarrow T_1$

② $T_2 : W(Z) \rightarrow T_3 : R(Z)$

$\Rightarrow$ edge $T_2 \rightarrow T_3$

The Precedence graph is acyclic and view serializable

The view serial schedule : $T_2 \rightarrow T_3 \rightarrow T_1$

Q. 8>
b) Explain triggers with example. write a trigger in SQL to call a procedure " Inform supervisor" whenever an employees salary is greater than the salary of his or her direct supervisor in the company database.

Soln A trigger is a special kind of stored procedure that is automatically created by the database management system whenever a specified event occurs in the database.

A trigger consist of three main components: an event, a condition, an action

Event specifies the database operation that activates the trigger, such as INSERT, DELETE or UPDATE on a particular relation

A condition is a boolean Expression that is evaluated once the event occurs.

The Action part may consist of one of SQL statement including invocation of stored Procedure.

Consider the company database, where each employee has a direct supervisor and a salary. suppose an employee's salary become grater than the salary of supervisor, the database

System must automatically call a stored procedure named Inform-supervisor.

The trigger can be written as follows

```
Create trigger check-salary
After update the salary on employee
for each row
where (new.salary >
        (Select salary
         from employee
         where ssn = new.super-ssn))

begin
   call inform-supervisor;

end;
```

Q 9 a) Describe the two phase locking protocol for concurrency control provide example to illustrate how it is ensures serializibility in transaction schedule.

Soln It is a concurrency control protocol that ensures conflict serializability of transaction schedules by controlling how locks are acquired and released.

→ A transaction is said to follow the two phase locking protocol with first unlock operation in the transaction.

→ Such transaction can be divided into two phases

① an expanding, during which new locks on items can be acquired with, but none can be released.

② Shrinking, during which existing locks can be released but no how locks can be aquired.

T1	T2
read. lock (y):	read_lock (x);
read-item (y):	read-item (x):
unlock (y):	unlock (x):

initial value x=20, y=30

Result serial-schedule T1 followed T2: x=50, y=80

Result of serial schedule T2 followed by T1: x=70, y=50.

Transaction T_1 and T_2 do not follow the two phase locking protocol because write-lock (x) operation follows the unlock (y) operation in T_1 and similarly the write operation follows the unlock (x) in T_2

T_1	T_2
read-lock (y);	read-lock (y);
read-item (y);	read-item (x);
write-lock (x);	write-lock (y);
unlock (y);	unlock (x);
read-item (x);	read-item (y);
x := x+y;	y := x+y;
write-item (x);	write-item (y);
unlock (x);	unlock (y);

If every one transaction in a schedule follows the two phase locking protocol, the schedule is guaranteed to be serializable

b) Explain the characteristics of NoSQL system.

① Scalability: In NoSQL system horizontal-scalability is generally used, where the distributed system is expanded by adding more nodes for data storage & processing as the volume of data grows.

② Availability, Replication and Eventual consistency. Many applications that use NoSQL system require continuous system availability. To accomplish this data is replicated two or more nodes ~~faits~~ in transparent manner, so that if one node fails, the data is still available on other nodes.

Replication improve data availability and can also improve read performance because read request can often be ~~serviced~~ ~~fro~~ served from any of the replicated data nodes.

(3) Replication models: Two major replication models are used in NoSQL Systems, master-slave and master master replication.

Master-slave replication requires one copy of the master copy. all the write operational must be applied to the copy. and then propogated to the slave copies usually ensures eventual consistency.

(4) Sharing of files: sharing service to distributing the load of accessing the file records to multiple nodes.

(5) High performance data access: It is necessary to find individual records or objects from among the millions of data records or objects in a file. Two techniques used.

(1) Hashing: Hashing h(k) is applied to the key K. and location of the object with key K is determined by value of h(k).

(2) Range partitioning:- The location is determined via a range of key values.

10 a) Explain binary locks and shared lock with algorithms

Soln Binary locks can have two states or values. locked or unlocked

A distinct lock is associated with each data base item x.

If lock on x is 1; item cannot be accessed by database operation. if '0' item can be accessed when requested

```
lock-item (X):
B: if lock (X) = 0
      then Lock (X) ← 1
      else
      begin
      wait (until Lock (X) = 0
      go to B
      end
Unlock-item (X)
      lock (X) ← 0
```

If every transaction are waiting, then wakeup one of the waiting transaction.

Shared / Exclusive locks:- If a transaction is to write item X it must have exclusive access to X. and Shared / exclusive or read / write locks. A lock is associated with an item X. lock (X), has three possible states, read-lock, write locked, or unlocked.

read lock item, because other transactions are allowed to read the item.

write lock item, because a single transactions exclusive loads the locks on the item.

```
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 wakeup the transaction go to B
  end;
```

b) Explain mongoDB data model CRUD operation and distributed system characteristics.

Soln → The mongoDB is a documented oriented NoSQL database that stores data in the form of documents rather than tables and rows

→ documents are grouped into collections which are analogous to relation in database.

→ mongoDB supports embeded documents and arrays allowing complex data structures to be stored within a single document.

CRUD Operations :

① Create operation : data is inserted into a collection. using insert operation, which add new document to a collection.

② Read operation : Data retrieval is performed using query operation that select documents based on specified condition.

③ update operation : Existing documents can be modified using update operation.

④ Delete operation : Documents can be deleted using delete operation based on given criteria.

⑤ MongoDB supports rich query expression like indexing, aggregation for efficient data access.

(Bheerappa.S.)