CBCS SCHEME

USN | 2 | V | 0 | 1 | 9 | C | S | 0 | 5 | 0 |

18CS53

Fifth Semester B.E. Degree Examination, Feb./Mar. 2022
Database Management System

Time: 3 hrs. Max. Marks: 100

Note: ***Answer any FIVE full questions, choosing ONE full question from each module.***

Module-1

1 a. List and discuss advantages of Database Management System over File Processing System. **(06 Marks)**

b. Explain three Schema Architecture and reason for need of mapping among schema level. **(08 Marks)**

c. Explain different types of attributes that occur in an E – R diagram model with example. **(06 Marks)**

OR

2 a. Explain characteristics of the Database approach. **(06 Marks)**

b. Discuss the different types of User friendly interfaces. **(06 Marks)**

c. Draw an ER diagram for an AIRLINES database schema with atleast five entities. Also specify primary key and structural constraints. **(08 Marks)**

Module-2

3 a. What are the basic operations that can change the states of relations in the database? Explain how the basic operations deal with constraints violations. **(06 Marks)**

b. Explain the terms Super key , Candidate key and Primary key. **(04 Marks)**

c. Given the following schema :
emp (fname, Lname , SSN , Bdate, address, gender , salary , superSSN , Dno)
dept (Dname , Dnumber , MgrSSN , mgrstartdate)
dept_loc (Dnumber , Dloc)
project (Pname, Pnumber, Ploc, Dnum)
works_on (ESSN, Pno , hours)
Dependent (ESSN , dependent_ name, gender , bdate , relationship)
Give the relation algebra expression for the following :
i) Retrieve the name of the manager of each department.
ii) For each project retrieve the project number , project name and number of employee who worked on that project.
iii) Retrieve the names of employees who work on all the project controlled by department 5.
iv) Retrieve the name of employees who have no dependents.
v) Retrieve number of Male and Female employee working in the Company. **(10 Marks)**

OR

4 a. Describe the steps of an algorithm for ER to Rational mapping with example. **(06 Marks)**

b. Write command that is used for table creation. Explain how constraints are specified in SQL during table creation, with suitable example. **(04 Marks)**

c. Given the following schema
Emp (Fname, Lname , SSN , bdate , address, gender, salary , superSSN , dno)
dept (dname , dnumber, mgrSSN , mgrstartdate)
dept_loc (dnumber, dloc)
project (Pname, Pnumber, Ploc , dnum)
works_on (ESSN, Pno, hours)
dependent (ESSN , dependent_name, gender, bdate, relationship)
Give the relation algebra expression for the following :
i) Retrieve the name and address of all employees who work for 'sports' department.
ii) Retrieve each department number, number of employers and their average salary.
iii) List the project number, controlling department number and department manager's last name , address and birthdate.
iv) Retrieve the name of employees with 2 or more dependents.
v) List female employees from dno = 20 earning more than 50000. **(10 Marks)**

Module-3

5 a. Define Database stored procedure. Explain creating and calling stored procedure with example. **(06 Marks)**

b. What is SQLJ and how is it different from JDBC? **(06 Marks)**

c. Consider the following schema :
Sailors (Sid , Sname , rating , age)
Boats (bid, bname, color)
Reservers (Sid , bid , day)
Write queries in SQL
i) Find the ages of sailors whose name begins and ends with A and has atleast three characters.
ii) Find the age of the youngest sailor who is eligible to vote (i.e. is atleast 18 years old) for each rating level with atleast two such sailors.
iii) Find the names of sailors who have not reserved a red boat. (use nested query).
iv) Compute increments for the rating of persons who have sailed two different boats on the same day. **(08 Marks)**

OR

6 a. What is CGI? Why was CGI introduced? What are the disadvantages of an architecture using CGI script? **(06 Marks)**

b. What is Dynamic SQL and how is it different from embedded SQL? Explain. **(06 Marks)**

c. Consider the following schema :
Sailors (Sid, Sname, rating , age)
Boats (bid, bname, color)
Reserves (Sid, bid, day).
Write queries in SQL.
i) Find the names of sailors who have reserved at least one boat.
ii) Find sailors whose rating is better than some sailors called 'Jennifer'. (Use nested query)
iii) Find the average age of sailor for each rating level that at least two sailors.
iv) Find the name and age of the oldest sailor. **(08 Marks)**

Module-4

7 a. Which normal form is based on 6 transitive functional dependencies and full functional dependency? Explain the same with example. **(08 Marks)**

b. A relation R satisfies the following : FDS : $A \rightarrow C$, $AC \rightarrow D$, $E \rightarrow AD$, $E \rightarrow H$. Find the cover for this set of FDS. **(06 Marks)**

c. Consider the universal relation : R = {A, B, C, D, E, F, G, H, I, J} and the set of functional dependencies. F = {$AB \rightarrow C$, $A \rightarrow DE$, $B \rightarrow F$, $F \rightarrow GH$, $D \rightarrow IJ$}.
Determine whether each decomposition has the loss less join property with respect to F.
$D_1 = \{R_1, R_2, R_3\}$; $R_1 = \{A, B, C, D, E\}$; $R_2 = \{B, F, G, H\}$; $R_3 = \{D, I, J\}$. **(06 Marks)**

OR

8 a. Write an algorithm to check whether decomposed relations are in 3NF with dependency preservation and non – additive join property. Consider universal relation R = (U, C, L, A) and the set of functional dependencies. F = {$P \rightarrow LCA$, $LC \rightarrow AP$, $A \rightarrow C$}. Decompose the relation R into 3NF with dependency preservation and non – additive join property. **(06 Marks)**

b. Define Normal Form. Explain 1NF, 2NF and 3NF with suitable examples for each. **(08 Marks)**

c. Consider two set of functional dependencies F = {$A \rightarrow C$, $AC \rightarrow D$, $E \rightarrow AD$, $E \rightarrow H$} and G = {$A \rightarrow CD$, $E \rightarrow AH$}. Are they equivalent? **(06 Marks)**

Module-5

9 a. What are the anomalies occur due to interleave execution? Explain them with example. **(08 Marks)**

b. Explain different types of locks used in concurrency control. **(06 Marks)**

c. Explain how shadow paging helps to recover from transaction failure. **(06 Marks)**

OR

10 a. Explain ACID property of transaction and system log. **(06 Marks)**

b. When deadlock and starvation problem occurs? Explain how these problems can be resolved. **(06 Marks)**

c. Explain ARIES recovery algorithm with example. **(08 Marks)**

* * * * *

Q.1)
a) List and discuss advantages of Database Management System over file processing system.

1) Controlling Redudancy
2) Restricting Unauthorized Access.
3) Providing Persistent storage for Program objects.
4) Providing storage structures & search Techniques for Efficient Query Processing
5) Providing Backup & Recovery.
6) Providing Multiple User Interface.
7) Representing complex Relationships among Data.
8) Enforcing Integrity Constraints.
9) Permitting Inferencing and Actions Using Rules & Triggers.
10) Additional Implications of using the Database Approach.

1) -> Reedudancy in storing the same data multiple times leads to several problems.
i) There is the need to perform a single logical update such as entering data on a new student - multiple times: one for each file where student data is recorded. This leads to
ii) duplication of effort.
ii) Storage space is wasted when the same data is stored repeatedly, and this problem may be large serious for large databases.
iii) Files that represent the same data may become inconsistent. In the DB approach, the views of different users groups are integrated during the DB design. Ideally we should have a dB design that stores each logical data item such in only one place in the dB. This is known as data normalization, and it ensures consistency & saves storage space

2) → When multiple users share a large DB, it is likely that most users will not be authorized to access all information in DB.

3) → DB's can be used to provide persistent storage for program objects & data structures. An object is said to be persistent, since it survives the termination of program execution & can later be directly retrieved by another program.

4) → The query processing & optimization module of the DBMS is responsible for choosing an efficient query execution plan for each query based on the existing storage structures.

5) A DBMS must provide facilities for recovering from H/W & S/W failure. The backup & recovery subsystem of the DBMS is responsible for recovery.

6) Many types of users with varying levels of technical knowledge use a DB, a DBMS should provide a variety of user interfaces. Ex: Query language for casual users, menu-driven interfaces for stand alone users.

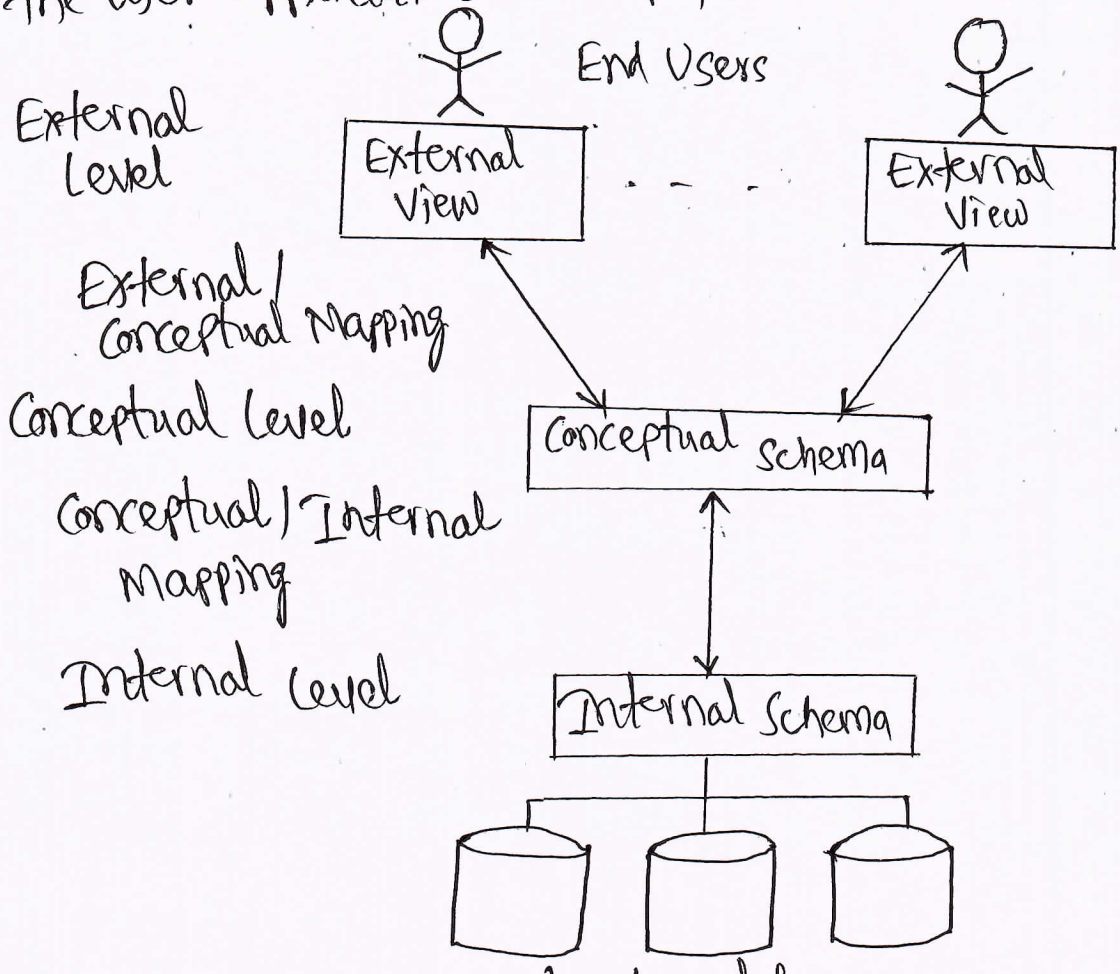
7) Representing Complex: A DBMS must have the capability to represent a variety of complex relationships among the data, to define new relationships as they arise & to retrieve and update related data easily & efficiently.

8) → Most DB applications have certain integrity constraints that must hold for the data. The simplest type of integrity constraint involves specifying a data type for each data item.

9) In a deductive DB system, one may specify declarative rules that allow to infer new data. Active DB systems go one step further by allowing "active rules" that can be used to initiate actions automatically.

b) Explain three schema architecture and reason for need of mapping among schema level.

→ The goal of the three-schema architecture is to seperate the user applications from physical dB.



In this architecture, Schemas can be defined at the following three levels:

1) The internal level has an internal schema,
   - Describes the physical storage structure of the dB.
   - Uses a physical data model & describes the complete details of data storage & access paths for the DB.

2) The conceptual level has a conceptual schema,
   - Describes the structure of the DB for a community users.
   - Hides the details of physical storage structures and concentrates on describing entities, relationships, user operations, & constraints.
   - Representational data model is used to describe the conceptual schema when a dB system is implemented.
   - This implementation conceptual schema is often based

only conceptual schema design model.

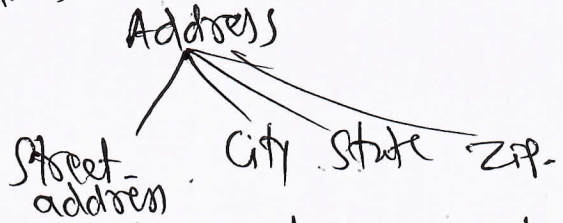
3) The External or view level includes a number of external schemas or user views,
- describes the part of the DB that a particular user group is interested in & hides the rest of the database from that user group.
- As in the previous level, each external schema is typically implemented using a representational data model possibly based on an external schema design in a high-level data model.

The DBMS must transform a request specified on an external schema into a request against the conceptual schema, & then into a request on the internal schema for processing over the stored DB.
If the request is a DB retrieval, the data extracted from the stored DB must be reformatted to match the user's external level. The process of transforming requests & result between levels are called mapping.

c) Explain different types of attributes that occur in an E-R diagram model with example.

→ Types of attributes occur in the ER model.
1) Simple Vs Composite.
2) Single valued Vs. multivalued
3) Stored Vs. Derived.

* Composite attributes can be divided into smaller subparts.
eg:- The address attribute of the Employee entity can be divided into street_address, City, State & Zip

*


* Attributes that are not divisible are called simple or atomic attributes.
eg: Age

* Single-valued :- Most attributes have a single value for a particular entity.
Eg:- Age is a single-valued attribute of a person.
* Multivalued :- An entity having multiple values for that attribute.
Eg:- Color of a car = {black, red}.
* Stored & Derived Attribute:
- Two attribute values are related. Eg:- The age & Bdate attributes of a person.
- The value of age can be determined from the current date & the value of that person's Bdate.
- The age attribute is hence called a derived attribute
- Bdate attribute is called a stored attribute

2)
a) Explain characteristics of the database approach.
→ 1) Self-Describing Nature of a DB System.
DB Approach :- * A fundamental characteristics of the DB approach is that the DB system contains not only the DB itself but also a complete definition or description of the DB structure & constraints.
* This Definition is stored in the DBMS catalog, which contains info such as the structure of the file, the type & storage format of each data item, & various constraints on the data.
* The information stored in the catalog is called meta-data, & it describes the structure of the primary DB.
Traditional File Processing:-
* In this data definition is typically part of the application programs themselves. Hence these programs are constrained to work with only one specific DB, whose structure is declared in the application programs.
2) Insulation between programs & data, & data Abstraction
* In traditional file processing, the structure of data files is embedded in the application programs so any changes to the structure, if we want to add another piece of data require changing all pgms that access that file.

* By contrast, DBMS access pgms do not require such c
- ges in most cases. The structure of data files is stored in the dbms catalog seperately from the access pgroms we call this property program-data independence.

3) Support of multiple views of the Data:

→ A DB typically has many types of users, each of whom r require a different perspective or view of the DB.

* A view may be a subset of the DB or it may contain virtual data i.e, derived from the DB files but is not explicitly stored.

* A multiuser DBMS whose users have a variety of distinc applications must provide facilities for defining multiple views.

4) Sharing of Data & Multiuser Transaction Processing

* A multiuser DBMS, as its name implies, must allow multiple users to access the DB at the same time.

* This is essential if data for multiple applications is to be integrated & maintained in a single DB.

* A fundamental role of multiuser DBMS s/w is to ensure that concurrent transactions operate correctly & efficiently.

* A transaction is an executing program or process that includes one or more DB accesses, such as reading or updating of DB records

b) Discuss the different types of User friendly interfaces.

→ 1) Menu-based Interfaces for web clients or Browsing.

These interfaces present the user with list of options (called menus) that lead the user through the formulation of a request.

2) Form-based interfaces displays a form to each user. Users can fill out all the form entries to insert new data or they can fill out only certain entries, in which case the DBMS will retrieve matching data for remaining entries.

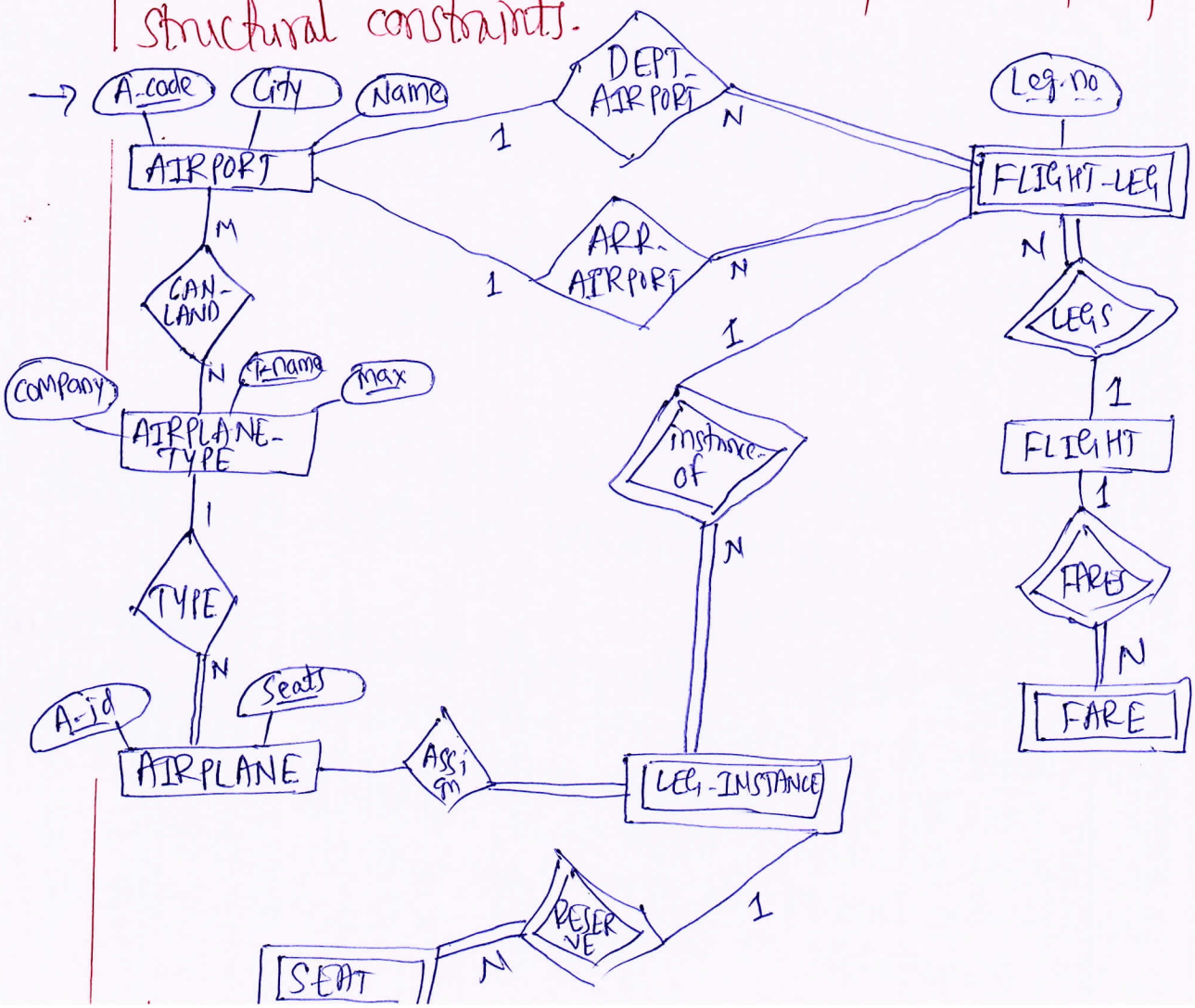
3) Graphical user Interfaces :- displays a schema to the user in diagrammatic form. The user then can specify a query by manipulating the diagram. GUIs utilize both menus & forms.

4) Natural language Interfaces :- accepts requests written in English or some other language & attempt to understand them.

5) Speech input & output use of speech as an input query & speech as an answer to a question or result. The speech input is detected using a library of predefined words & used to set up the parameters that are supplied to the queries.

6) Interfaces for parametric users such as bank tellers, often have a small set of operations that they must perform repeatedly.

c) Draw an ER diagram for an AIRLINES DB schema with atleast five entities. Also specify primary key and structural constraints.



Q5/Q4 What are the basic operations that can change the state of relations in the DB? Explain how the basic operations deal with constraint violations.

→ * Three basic operations that can change the states of relations in the DB.

i) Insert ii) Delete iii) Update

The Insert Operation:- The insert operation provides a list of attribute values for new tuple t that is to be inserted into a relation R.

* Insert can violate domain, key, entity and referential integrity constraints.

Operation:-

1) Insert <'Cecilia', 'F', 'Kolonsky', NULL, '1960-04-05', '6357 Wim Lane, Katy, TX', F, 28000, NULL, 4> into Employee.

Result: It violates the entity integrity constraint, so it is rejected.

2) Insert <'Alicia', 'J', 'Zelaya', '999887777', '1960-04-05', '6357 Windy Lane, Katy, TX', F, 28000, '987654321', 4> into EMP.

Result:- This insertion violates the key constraint, so it is rejected.

The Delete Operation:- The delete operation can violate only referential integrity will be violated.

Operation:-

Delete the Employee tuple with SSN = '999887777'.

Result:- This deletion is not acceptable, as it violates referential integrity.

The Update operation:- The update operation is used to change the values one or more attributes in a tuple of some relation R. It is necessary to specify a condition on the attributes of the relation to select the tuple to be modified.

Update the Dno of the emp tuple with SSN = '99988' to 7.

→ Not acceptable, as it violates referential integrity.

b) Explain the terms Superkey, Candidate Key & Primary Key

→ *Superkey is a single key or a group of multiple keys that can uniquely identify tuples in a table.

* A candidate key is a subset of Super keys & is devoid of any unnecessary attributes that are not important for uniquely identifying tuples.

* The primary key uniquely identifies each record in a table. Primary keys must contain unique values, & cannot contain NULL values.

c) Given the following schema:

emp (fname, Lname, SSN, Bdate, address, Gender, Salary, SuperSSN, Dno)

dept (Dname, Dnumber, MgrSSN, mgrstartdate)

dept_loc (Dnumber, Dloc)

project (Pname, Pnumber, Ploc, Dnum)

works_on (ESSN, Pno, hours)

Dependent (ESSN, dependent_name, gender, bdate, relationship)

Give the relational algebra expression for the following.

i) Retrieve the name of the manager of each department.

$$R_1 \leftarrow emp \underset{SSN = MgrSSN}{\bowtie} dept$$

$$R_2 \leftarrow \pi_{fname, Lname}(R_1)$$

ii) For each project retrieve the project number, project name & no of employees who worked on that project.

$$R_1 \leftarrow Project \underset{Pnumber = Pno}{\bowtie} works\_on$$

$$R_2 \leftarrow \pi_{pnumber, Pname, COUNT(*)}(R_1)$$

iii) Retrieve the names of employees who work on all the project controlled by department 5.

$$R_1 \leftarrow \rho_{(Pno)}(\pi_{pnumber}(\sigma_{Dnum=5}(PROJECT)))$$

$$R_2 \leftarrow \rho_{(SSN, Pno)}(\pi_{ESSN, Pno}(WORKS\_ON))$$

$$R_3 \leftarrow EMP - R_1 \div R_2$$

$$R_4 \leftarrow \pi_{Lname, Fname}(R_3 * EMP)$$

iv) Retrieve the name of the employees who have no depen

→ $R_1 \leftarrow \Pi_{SSN}(Emp)$

$R2(SSN) \leftarrow \Pi_{ESSN}(Dependent)$

$R3 \leftarrow R_1 - R_2$

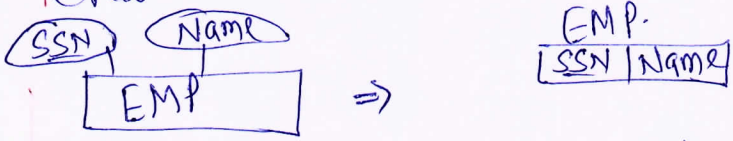
$R4 \leftarrow \Pi_{Lname, Fname}(R_1 * R_2)$

v) Retrieve number of male & female employees working in the company.

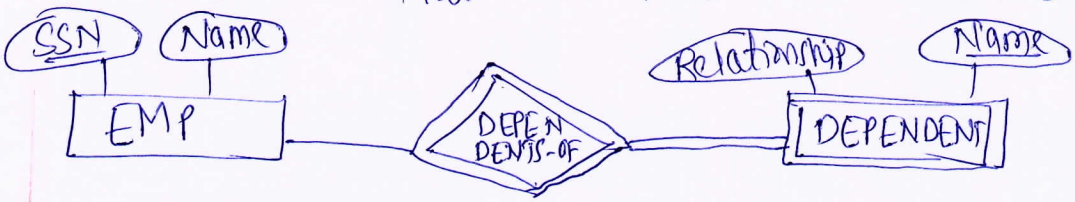
$\mathfrak{F}_{COUNT\ gender}(Emp)$.

Q. 4) a) Describe the steps of an algorithm for ER to Relational mapping with example.

→ Step 1: For each regular entity type E in the ER schema, create a relational R that includes all the simple attributes



Step 2: For each weak entity type W in the ER schema with owner entity type E, create a relation R, & include all simple attributes of W as attributes. In addition, include as foreign key attributes of R the primary key attribute of the relation that correspond to the owner entity type.



DEPENDENT

SSN	Relationship	Name

Step 3: For each binary 1:1 relationship type R identify the relation (N) relation S. The primary key of T as a FK of S. Simple attributes of R map to attributes of S.

EMP. DEPARTMENT

Mgr. SSN	Start. Date

Step 4: For each regular binary 1:N relationship type R identify the relation (N) relation S. The primary key of T as a FK of Simple attributes of R map to attributes of S.

Emp
| SuperSSN. |

Step 5: For each binary M:N relationship type R, create a relation S. Include the PK's of participant relations as FK in S. Their combination will be the PK for S. Simple attributes of R became attributes of S.

WORKS_FOR
| ESSN | Dnumber |

Step 6: For each multivalued attribute A, Create a new relation. This relation will include an attribute corresponding to A, plus the PK K of the parent relation as a FK in R. The PK of R is the combination of A & K.

DEPT_LOC.
| DLoc | Dnumber |

Step 7: For each n-ary relationship type R, where $n \geq 2$, Create a new relation S to represent R.

b) Write command that is used for table creation. Explain how constraints are specified in SQL during table creation, with suitable example.

→ The create table command is used to specify a new relation by giving its name & specifying its attributes and initial constraints.

i) NOT NULL : It is specified if NULL is not permitted for particular attribute.

Create table EMP(Fname varchar(20) NOT NULL, Lname varchar(20))

ii) DEFAULT :- It is possible to define a default value for an attribute by appending the clause Default to an attribute definition.

Create table DEPT (....,
Mgr_SSN Char(9) NOT NULL DEFAULT '88866555S'
----)

(Y Given the following schema:

Emp(Fname, Lname, SSN, bdate, address, gender, salary, SuperSSN, dno)

dept(dname, dnumber, mgrSSN, mgrstartdate)

dept_loc(dnumber, dloc)

project(Pname, Pnumber, Ploc, dnum)

works-on(ESSN, Pno, hours)

dependent(ESSN, dependent_name, gender, bdate, relationship)

Give the relational algebra expression for the following

i) Retrieve the name & address of all employees who work for 'Sports' department

$$R_1 \leftarrow \sigma_{Dname='Sports'}(Dept)$$

$$R_2 \leftarrow R_1 \bowtie_{Dnumber=Dno} EMP$$

$$R_3 \leftarrow \pi_{Fname, Lname, Address}(R_2)$$

ii) Retrieve each dept number, number of of employees & their average salary.

$$Dno\ \mathcal{F}\ COUNT(SSN), AVERAGE\ salary(EMP)$$

iii) List the project number, controlling department number & department manager's last name, address & birthdate.

$$R_1 \leftarrow PROJECT \rtimes_{Dnum=Dnumber} DEPT$$

$$R_2 \leftarrow R_1 \bowtie_{Mgr\text{-}SSN=SSN} EMPLOYEE$$

$$R_3 \leftarrow \pi_{Pnumber, Dnum, Lname, Address, Bdate}(R_2)$$

iv) Retrieve the name of the employees with 2 or more dependents.

$$T_1(SSN, No.\ of\ dependents) \leftarrow ESSN\ \mathcal{F}\ count\ dependent\_name(DEPENDENT)$$

$$T_2 \leftarrow \sigma_{No\ of\ dependents > 2}(T_1)$$

$$T_3 \leftarrow \pi_{Lname, Fname}(T_2 \bowtie EMP)$$

Q 5) a) Define database stored procedure. Explain creating & calling stored procedure with example.

→ * A stored procedure is a pgm that is executed through a single SQL statement that can be locally executed & compiled within the process space of the database server.

* The results can be packaged into one big result & returned to the application, or the application logic can be performed directly at the server, without having to transmit the results to the client.

Creating a simple stored Procedure

```
CREATE PROCEDURE ShowNumberOfOrders
  SELECT  C.Cid, C.Cname, COUNT(*)
    FROM  CUSTOMERS C, ORDERS O
    where C.Cid = O.Cid
    Group by C.Cid, C.Cname
```

Calling Stored Procedures:

* Stored procedures can be called in interactive SQL with the CALL statement

```
CREATE PROCEDURE RankCustomers( IN number INTEGER
  Language Java
  External Name 'file:///C:/StoredProcedures/rank.jar
CALL StoredProcedureName (argument 1, argument 2, ....
                          Argument N);
```

* Calling Stored Procedures from JDBC.

```
Callable Statement cstmt = con.prepareCall("{call
                    ShowNumberOfOrders}");
ResultSet rs = cstmt.executeQuery()
while (rs.next())
```

b) what is SQLJ and how it is different from JDBC?

→ * SQLJ was developed to complement the dynamic way of creating queries in JDBC with static model.

* Unlike JDBC, having semi-static SQL queries allows the compiler to perform SQL syntax checks, strong type checks, of the compatability of the host variables with the respective SQL attributes.

*. In SQLJ, variables in the host language allows are bound statically to the same arguments, whereas in JDBC, we need seperate statements to bind each variable to an argument & to retrieve the result

* SQLJ pgms require fewer lines of code than JDBC pgms. They are shorter & hence easier to debug

* SQLJ provides strong type checking of query results & other return parameters, while JDBC values are passed to & from SQL without having been checked at compile time.

* SQLJ fills a complementary role for static SQL, while JDBC provides a complete dynamic SQL interface from Java to relational DBs.

* SQLJ is used to write a pgm when you want to write an application that you can deploy to another DB. while, JDBC is used when you do not want to have a SQLJ layer during deployment

* SQLJ is used when you want to be able to check your pgm for errors at translation time rather than at run time while JDBC is used when your pgm uses dynamic SQL.

c) Consider the following schema:

Sailors (Sid, Sname, rating, age)
Boats (bid, bname, color)
Reserves (Sid, bid, day)
Write queries in SQL

ii) Find the ages of sailors whose name begins & ends with A & has atleast three characters.

```
Select age
From Sailors
where Sname like 'A_ _ %.A';
```

iii) Find the age of the youngest sailor who is eligible to vote for each rating level with atleast two such sailors.

```
Select S.rating, MIN(S.age)
From Sailors S
where S.age >= 18
Group by S.rating
Having 1 < (select count(*)
            From Sailors S2
            where S.rating = S2.rating).
```

iii) Find the names of sailors who have not reserved a red boat

```
Select S.name
From Sailors S
where S.id not in ( Select Sid
                    From Sailors S, reserves r, boats b
                    where S.id = r.sid & r.bid = b.bid).
```

iv) Compute increments for the rating of persons who have sailed two different boats on the same day.

```
Select rating+1 !
From sailor
where Sid in (Select sid
              From Reserves R
              group by day, sid
              Having count(*) = 2)
```

Q6) a) what is CGI ? why was CGI introduced ? what are the disadvantages of an architecture using CGI script ?

→ * The CGI connects HTML forms with application pgms.
* It is protocol that defines how arguments from forms are passed to pgms at the server side

* Pgms that communicate with web server via CGI are often called CGI scripts.
* CGI was introduced to transmit arguments from HTML forms to application pgms running at the middle tier
* Interaction with standalone application was costly.

The disadvantages of CGI are

1) Each page request results in the creation of a new process.
2) Data can not be easily cached beth page loads.
3) It uses a lot of processing time.
4) No resource sharing beth application pgms.

b) What is Dynamic SQL & how is it different from embedded SQL? Explain.

→ * we may not be able to executed, even though there is some algorithm by which the application can construct the necessary SQL statements once a users command is issued. SQL provides some facilities to deal with such situations. These are referred to as Dynamic SQL
* It is a programming technique that enables you to build SQL statements dynamically at runtime.
* The preparation of a Dynamic SQL command occurs at runtime & is runtime overhead whereas Embedded SQL commands can be prepared once at compile time & re-executed as often desired.
* Dynamic SQL less efficient but Embedded SQL is more efficient.
* Dynamic SQL is more flexible than Embedded SQL
* In Dynamic SQL, EXECUTE, PREPARE statements are used. In embedded SQL, EXECUTE, PREPARE statements are not used.

c) Consider the following schema:

Sailors (Sid, Sname, rating, age)

Boats (bid, bname, color)

Reserves (Sid, bid, day)

write queries in SQL.

i) Find the names of sailors who have reserved atleast one boat.

```
Select Distinct S.Sname
From Sailors S, Reserves R
where S.Sid = R.Sid.
```

ii) Find sailors whose rating is better than some sailors called 'Jennifer'.

```
Select S.Sname, S.rating
From sailors
where S.rating > ALL( select S1.rating
                      From sailors S1
                      where S1.sname = 'Jennifer')
```

iii) Find the average age of sailor for each rating level that atleast two sailors.

```
Select S.rating, AVG(S.age)
From Sailors S
group by S.rating
Having COUNT(*) > 1.
```

iv) Find the name & age of the oldest sailor.

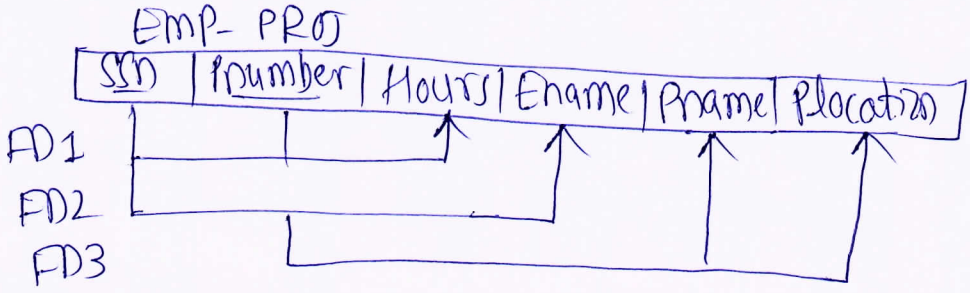
```
Select S.Sname, S.age
From Sailors S
where S.age = (Select MAX(S2.age)
               From Sailors S2).
```

Q.7) a) Which normal form is based on 6 transitive FD's & full functional dependency? Explain the same with ex.

→ 2NF is based on a transitive FD's & full functional dependency.

Second Normal Form

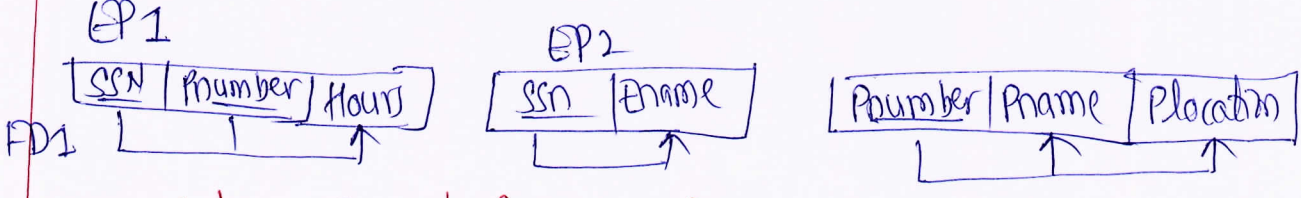
* A FD $X \rightarrow Y$ is a full FD if removal of any attribut from X means that the dependency does not hold any m
* A FD $X \rightarrow Y$ is a partial dependency if some attribute $A \in X$ can be removed from X & the dependency still hold



* In the above figure, {SSN, Pnumber} → Hours is a full dependency.
* SSN → Ename & Pnumber → {Pname, Plocation} is a partial dependency.

Defn:- A relation schema R is in 2NF if every nonprime attribute A in R is a fully functionally dependent on the PK of R.

↓ 2NF Normalization.



b) A relation R satisfies the following FD's:
$A \rightarrow C$, $AC \rightarrow D$, $E \rightarrow AD$, $E \rightarrow H$. Find the cover for this set of FDs.

→ i) $A \rightarrow C$
ii) $AC \rightarrow D$
iii) $E \rightarrow A$ } Decomposition.
iv) $E \rightarrow D$
v) $E \rightarrow H$

Remove $E \rightarrow D$ Redudant

$AC \rightarrow D$

$A^+ = \{A, C\}$

∴ C is extraneous & is removed.

Minimal cover $G := \{A \rightarrow CD, E \rightarrow AH\}$.

c) Consider the universal relation: $R = \{A, B, C, D, E, F, G, H, I, J\}$ & the set of FD's, $F = \{AB \rightarrow C, A \rightarrow DE, B \rightarrow F, F \rightarrow GH, D \rightarrow IJ\}$. Determine whether each decomposition has loss less join property wrt F. $D_1 = \{R_1, R_2, R_3\}$; $R_1 = \{A, B, C, D, E\}$; $R_2 = \{B, F, G, H\}$; $R_3 = \{D, I, J\}$.

→ $F = \{AB \rightarrow C, A \rightarrow DE, B \rightarrow F, F \rightarrow GH, D \rightarrow IJ\}$

$D_1 = \{R_1, R_2, R_3\}$

$R_1 = \{A, B, C, D, E\}$

$R_2 = \{B, F, G, H\}$

$R_3 = \{D, I, J\}$

$R_1 \cup R_2 \cup R_3 = R$ (universal relation).

$R_1 \cap R_2 = \phi$

$R_2 \cap R_3 = \phi$

$R_1 \cap R_3 = \phi$

∴ Hence each decomposition has loss less join property with respect to F.

8) Write an algorithm to check whether decomposed relations are in 3NF with dependency preservation & non-additive join property. Consider universal relation $R = (U, C, L, A)$ & the set of FD's. $F = \{P \rightarrow LCA, LC \rightarrow AP \rightarrow A \rightarrow C\}$. Decompose the relation R into 3NF with dependency preservation & non-additive join property.

Algorithm for Testing for Non-additive Join property

Input: A universal relation R, a decomposition $D = \{R_1, R_2 \ldots R_m\}$ of R, & set F of FD's.

1) Create an initial matrix S with one row i for each relation R_i in D & one column j for each attribute A_j in R.

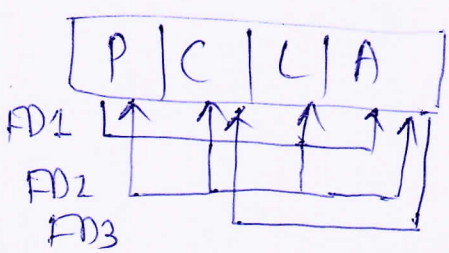
2) Set $S(i,j) := b_{ij}$ for all matrix entries.

3) for each row i representing relation schema R_i
   { for each column j representing attribute A_j
      { for (relation R_i includes attribute A_j) then set
         $S(i,j) := a_j$; }; };

4) Repeat the loop until a complete loop execution results in no changes to S.

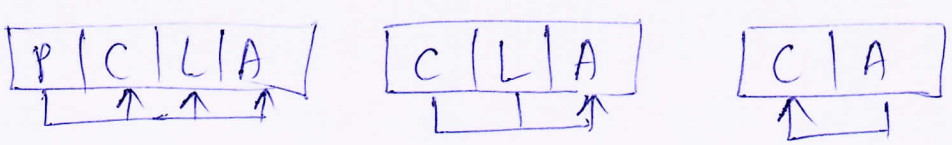
5) If a row is made up of entirely of a symbols, then the decomposition had the non additive join property; otherwise it does not.

$R = (V, C, L, A)$



From the Diagram it is clearly visible that it is not 3NF as it has transitive dependency.

Decompose table into 3 subtables as



Now it is 3NF.

b) Define Normal form. Explain 1NF, 2NF, 3NF with suitable example for each.

→ The normal form of a relation refers to the highest normal form condition that it meets, & hence indicates the degree

which it has been normalised.

1NF :- 1NF disallows relations within relations or relations as attributes values within tuples.

* The only attribute values permitted by 1NF are single atomic values.

DEPT

Dname	Dnumber	Dmgr_SSN	Dlocations

↓ 1NF

DEPT

Dname	Dnumber	Dmgr_SSN	Dlocations
Research	5	111	{B, C, D}
CS	4	222	{C}
EC	1	444	{B}

↓ 1NF

Dname	Dnumber	Dmgr_SSN	Dlocations
Research	5	111	B
Research	5	111	C
Research	5	111	D
CS	4	222	C
EC	1	444	B.

2NF :→ 2NF is based on the concept of full functional dependency.

* A FD X→Y is a full functional dependency if removal of any attribute A from X means that the dependency does not hold any more, i.e, for any attribute A ∈ X, (X − {A}) does not functionally determine Y.
* A relation schema R is in 2NF if every nonprime attribute A in R is fully functionally dependent on the PK of R.

EMP_PROJ

SSN	Pnumber	Hours	Ename	Pname	Plocation

FD1
FD2
FD3

↓ 2NF

EP1

SSN	Pnumber	Hours

FD1

EP2

SSN	Ename

FD2

EP3

Pnumber	Pname	Plocation

FD3.

* 3NF :- A relation schema R is in 3NF if it is 2NF & no non-prime attribute A in R is transitively dependent on the PK

EMP-DEPT

Ename	SSN	Bdate	Address	Dnumber	Dname	Dmgr-SSN

↓ 3NF

ED1

Ename	SSN	Bdate	Address	Dnumber

ED2

Dnumber	Dname	Dmgr-SSN

c) Consider two set of FD's $F = \{A \to C, AC \to D, E \to AD, E \to H\}$ & $G = \{A \to CD, E \to AH\}$. Are they equivalent?

→

F: $A \to C$ $AC \to D$ $E \to AD$ $E \to H$	G: $A \to CD$ $E \to AH$
$A^+ : ACD$ $(AC)^+ : ACD$ $(E)^+ : ACDHE$ $F \subseteq G$	$A^+ : ACD$ $E^+ : ACDEH.$ $G \subseteq F$

$F = G.$

Q.9) a) What are the anomalies occur due to interleave execution? Explain them with example.

1) The Lost update Problem:- Occurs when two transactions that access the same DB items have their operations interleaved in a way that makes the value of some DB item incorrect

T_1	T_2
r(X)	
X = X-N	
	r(X)
	X = X+M
w(X)	
r(Y)	
	w(X)
Y = Y+N	
w(Y)	

2) The dirty read problem:- occurs when one transaction updates a db item & the transaction fails for some reason.
* Meanwhile the updated item is accessed by another transaction before it is changed back to its original value.

T_1	T_2
r(X)	
X = X-N	
w(X)	
	r(X)
	X = X+M
	w(X)
r(Y)	

3) The incorrect summary problem: If one transaction is calculating an aggregate summary function on a number of db items while other transactions are updating some these items.

T_1	T_2
	sum = 0
	r(A)
	sum = sum+A
	⋮
r(X)	
X = X-N	
w(X)	
	r(X)
	sum = sum+X
	r(Y)
r(Y)	sum = sum+Y
Y = Y+N	

4) The Unrepeatable Read Problem: Transaction T reads the same item twice & gets different values on each read, since the item was modified by another transaction T' betn the two reads.

b) Explain different types of locks used in concurrency control.

→ 1) Binary locks :- A binary lock can have two states: locked & unlocked.

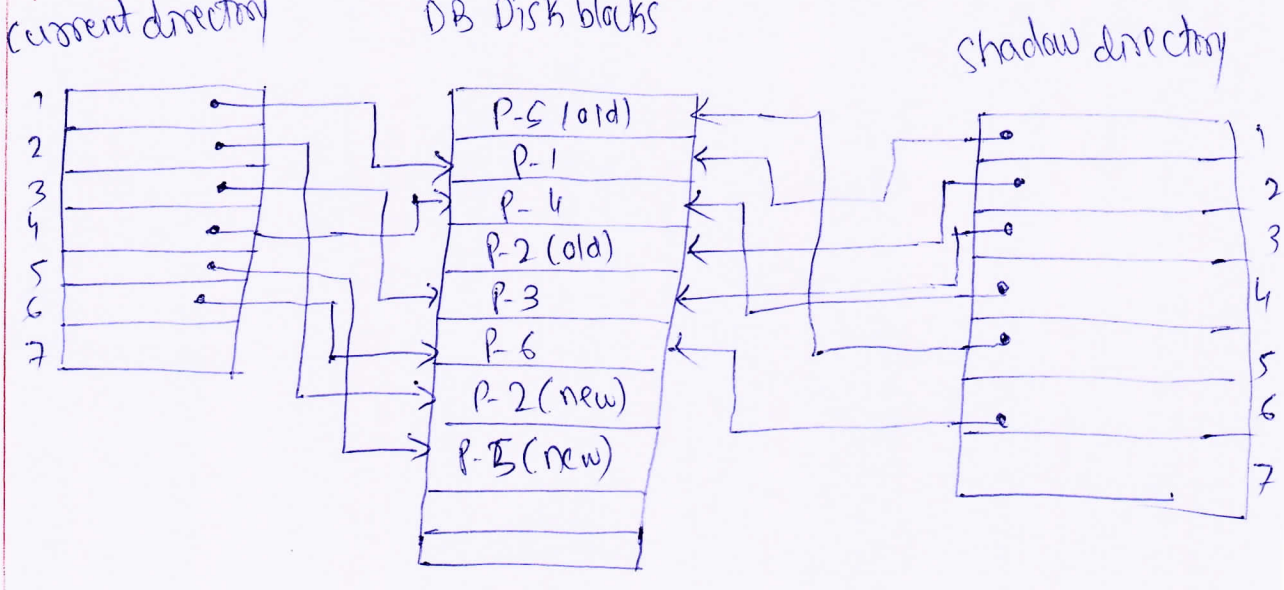
* If the value of the lock on x is 1, item x can not be accessed by a db operation that requests the item.
* If the value of the lock on x is 0, the item can be accessed when requested, the lock value is changed to 1

2) Shared / Exclusive Locks :-

* In this scheme there are 3 locking operations: read_lock(x), write_lock(x), and unlock(x).
* A read_locked item is called shared_locked because other transactions are allowed to read the item.
* A write_locked item is called exclusive_locked because a single transaction exclusively holds the lock on the item.

c) Explain how shadow paging helps to recover from transaction failure.

→



* This recovery scheme does not require the use of a log in a single-user environment.
* Shadow paging considers the db to be made up of a no of fixed size disk pages - say, n - for recovery purposes.
* A directory with n entries is constructed, where the ith entry points to the ith db page on disk.
* The directory is kept in main directory if it is not too large & all references - reads or writes - to database pages on disk go through it.
* When a transaction begins executing, the current directory - whose entries point to the most recent or current db pages on disk - is copied into a shadow directory.
* The shadow directory is then saved on disk while the current directory is used by the transaction.
* During the transaction execution, the shadow directory is never modified.
* The old version is referenced by the shadow directory & the new version by the current directory.

Q. 10) Explain ACID property of transaction and system log.

→ a) Atomicity:- A transaction is an atomic unit of processing & it is either performed entirely or not at all.

b) Consistency Preservation:- A transaction should be consistency preserving that is it must take the db from one consistent state to another.

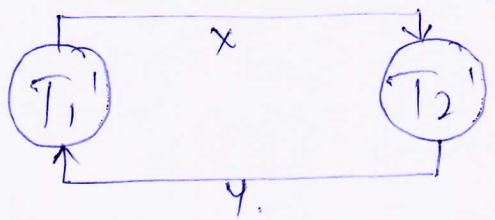
c) Isolation:- A transaction should appear as though it is being executed in isolation from other transactions.

d) Durability:- If a transaction changes the db and is committed the changes must never be lost because of any failure.

The system log:- It keeps track of all transaction operations that affect the values of db items.
* This info may be needed to permit recovery from transaction failures.

b) When deadlock and starvation problem occurs? Explain how these problems can be resolved.

→ * Deadlock occurs when each transaction T in a set of two or more transactions is waiting for some item that is locked by some other transaction T' in the set.



* Starvation may occur when we ~~use~~ a transaction can not proceed for an indefinite period of time while other transactions in the system continume normally.

* Deadlock can be resolved by.
  1) Deadlock prevention Protocols.
     a) Conservative two-phase locking
     b) ordering of all items.
     c) wait-die
     d) wound-wait
     e) No waiting algorithm.
     f) cautious waiting
  2) Deadlock Detection
     a) wait-for graph
     b) Timeouts.

* Starvation problem can be resolved by:
  a) A Fair waiting queue.

c) Explain ARIES recovery algorithm with example.

→ * ARIES uses a steal/no-force approach for writing, & it is based on three concepts.
  1) Write-ahead logging
  2) Repeating history during redo, &
  3) logging changes during undo

* The ARIES recovery procedure consists of 3 main steps.

1) Analysis.

2) REDO

3) UNDO.

* The Analysis step:-

a) Identifies the dirty pages in the buffer & the set of transactions active at the time of the crash.

b) The appropriate point in the log where the REDO operation should start is also determined.

* The REDO phase:-

- reapplies updates from the log to the db.

- Certain info in the ARIES log will provide the start point for REDO, from which REDO operations are applied until the end of log is reached.

* The UNDO phase:-

- The log is scanned backward & the operations of transactions that were active at the time of the crash are undone in reverse order.

(Shivachalesh S.G.)

HOD

HOD
Computer Science & Engineering
KLS Vishwanathrao Deshpande
Institute of Technology, Haliyal.

Dean, Academics