2073 Bhadra

Q1) Why is data independence important in data modeling? Differentiate between schema and instances. [4]

Ans The ability to modify a schema definition in one level without affecting a schema definition in the next higher level is called Data Independence. It is independence between the programs and the data.

There are two levels of data independence

1) Physical Data Independence
   It is the ability to modify the physical schema without causing application programs to be re-written.

2) Logical Data Independence
   It is the ability to modify the logical schema without causing application programs to be rewritten.

Data independence is important in data-modeling because of following reasons.

i) Improve performance
ii) Change in data structure do not require change in application program.
iii) Hide implementation details from the users.
iv) Security can be improved.
v) Standards can be enforced.
vi) Better service to the users.
vii) Flexibility in system improvement.
viii) Cost of developing & maintaining systems is lower.
ix) Integrity, consistency, security and availability can be ensured.

2nd Part.

Differences between schema and instances are as follows:

Schema : Design of database is called schema. Schema is of three types: Physical, logical & view schema.

i) Physical schema: The design of a database at physical level where how the data stored in blocks of storage is described.

ii) Logical schema: The design of a database at logical level where programmer and database administrators work.

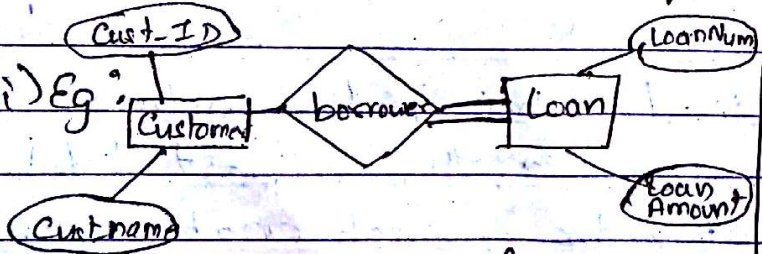
iii) View schema: The design of database at view level which describes end user interaction with database systems.

Instances : The data stored in database at a particular moment of time is called instance of database. Database schema defines the variable declarations in tables that belong to a particular database; the values of these variables at a moment of time is called the instance of that database.

Q2) Differentiate total and partical participation with suitable example and draw an ER diagram for the airport database. Be sure to indicate the various attributes of each entity. Every airplane has a registration number and each airplane is of a specific model. The airport accommodates a number of airplane

models and each model is identified by a model number (eg DC-10) and has a capacity and a weight. A number of technician works at the airport. You need to store the name, SSN, address, phone number and salary of each technician. Each technician is an expert on one or more plane model(s) and his or her expertise may overlap with that of other technicians. This information about technicians must also be recorded. Traffic controllers have an annual medical examination. For each traffic controller you must store the data of the most recent exam. [4+8]

Ans

Total Participation	Partial Participation
i) The participation of an entity set E in a relationship set R is said to be total if every entity E participates in at least one relationship R.	i) If only some entities in entity set E participates in relationships in R, the participation of entity set E in relationship set R is said to be partial.
ii) ~~Eg. Consider~~ It is represented by double line connecting entities in relationship	ii) It is represented by single line connecting entities in relationship.
iii) Eg: 	iii) Eg: In the given figure, the participation of customer in borrower is partial.

Here, A double line from loan to borrower indicates that each loan must have at least one associated customer, where the relationship borrower between customers and loans is considered.

2nd Part

ER diagram for the Airport database

Airplane (registration_no , model_name)

~~Airport (model_no , model_name , capacity , weight)~~

Technician (name , SSN , address , phone_no , salary)

~~Employee Airport (Registration_no~~

~~Airport (Name~~

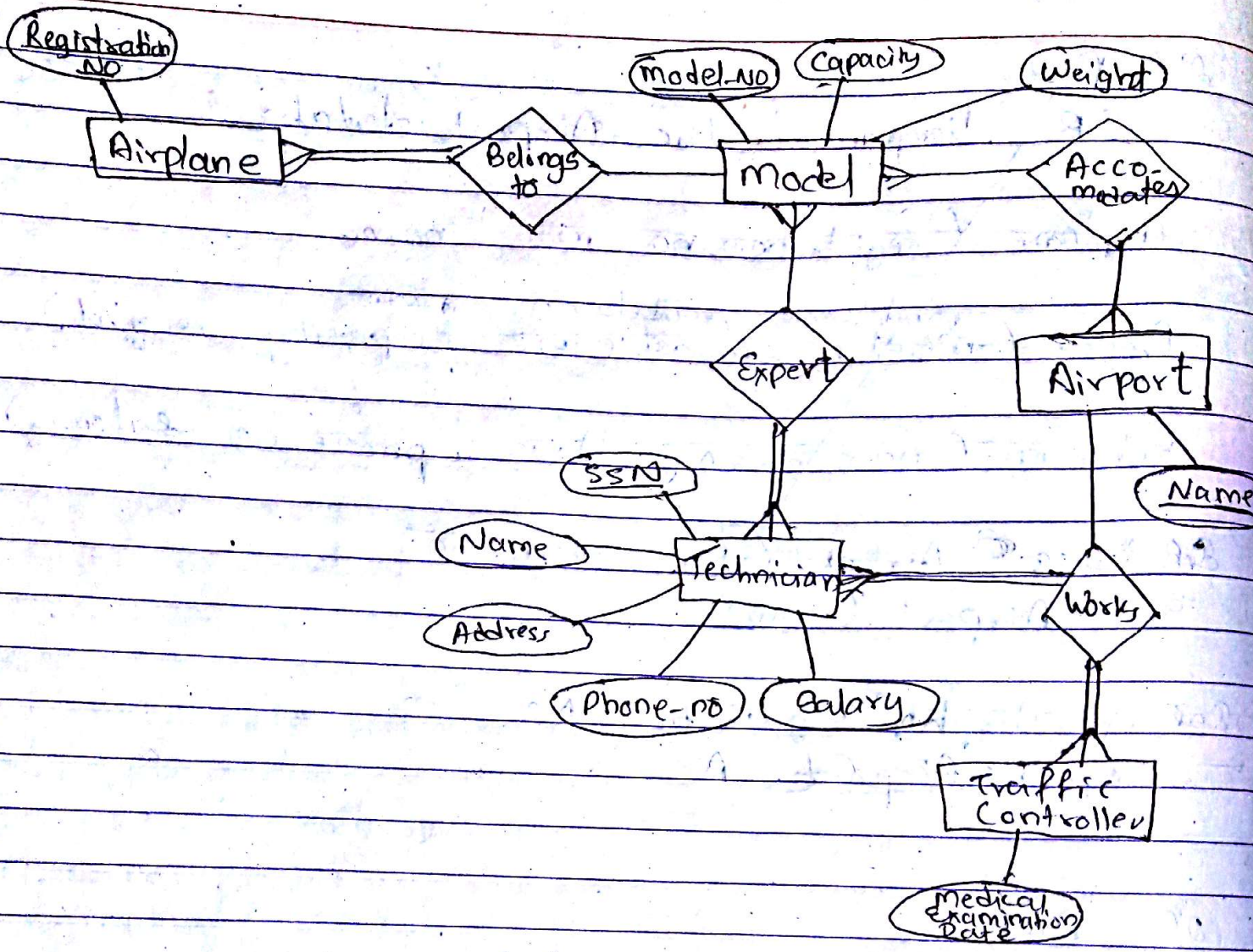
1) ~~Traffic~~ Airplane { Registration_No }

2) ~~Airplane~~ ~~Airport~~ A.

2nd Part:

ER diagram for Airport database

1) Airplane { Registration_No }
2) Airport { Name }
3) Model { Model_No , Capacity , Weight }
4) Technician { SSN , Name , address , phone_No , Salary }
5) Traffic Controller { Medical Examination_Date }



03) Consider the following relational schema.

Employee (Ename, street, city)
Works (Ename, company_name, salary)
Company (company_name, city)
Manages (Ename, manager_name)

a) Write the queries in Relational Algebra.

i) Find all the employees name who work in 'NMB Bank'

⇒ $\Pi_{Ename}(\sigma_{company\_name = 'NMB\ bank'}(Employee \bowtie Works$

ii) Find all the employees names who live in the same city as their company is located.

⇒ $\Pi_{Ename}(\sigma_{Employee.city = Company.city}(Employee \bowtie Company))$

iii) Find the name and city of those employees whose salary is greater than 30000 and lives in 'ktm' city.

⇒ $\Pi_{Ename, city}(\sigma_{salary > 30000 \text{ AND } city = 'ktm'}(Employee \bowtie Works))$

b) Write SQL queries for the following.

i) Create Employee and Works relation with primary key and foreign key constraints.

⇒ ~~Alter table Employee~~
~~Add constraint f-key~~
~~Foreign key (Ename Works) references Work (Ename)~~

⇒ Alter table Employee add constraint p_key primary key (Ename)
Alter table Works add constraint w_key primary key (Ename)
Alter table Employee add constraint f_key foreign key (Ename) references Works (Ename)

b) Write SQL queries for the following,

i) Create Employee and Works relation with primary key and foreign key constraints.

→ Create table Employee
(Ename varchar(50) primary key,
street varchar(50),
city varchar(20),
works integer
);
Create table Works
(Ename varchar(50) primary key,
Company-name varchar(30),
Salary int not null
);
Alter table Employee add constraint f-key
foreign key (Works) references Works (Ename)

ii) Find the employee name, their company name and city name which ends with 'pur' as substring.

Select Ename, company-name, city
from Employee, Company
where

iii) Increase the salary of each employees by 25% where salary is less than 30000.

→ Update ~~salary worker. Employee~~ Works
set salary = salary * 1.25
where salary < 30000
Update employee
set salary = salary * 1.25
where salary < 30000

4a) What do you mean by functional dependencies? Define formally. What is BCNF? [3+3]

Ans → Functional dependencies describes relationship between attributes.
→ It is an important concept associated with normalization.

Ans. Functional dependency (FD) is a set of constraints between two attributes in a relation. Functional dependency says that if two tuple have same values for attributes A1, A2, ..., An, then those two tuples must have to have same values for attributes B1, B2, ..., Bn.

Functional dependency is represented by an arrow sign (→) that is, $X \rightarrow Y$, where X functionally determines Y. The left-hand side attributes determine the values of attributes on the right hand side.

Formally:
If column A of a table uniquely identifies the column B of same table then it can be represented as $A \rightarrow B$
(Attribute B is functionally dependent on attribute A).

Types of Functional Dependency
i) Trivial functional dependency
ii) Non-trivial functional dependency
iii) Multivalued Dependency
iv) Transitive Dependency.

2nd Part

BCNF (Boyce-Codd Normal Form)
It is an advance version of 3NF that's why it is also refered as 3.5NF. BCNF is stricter than 3NF. A table complies with BCNF if it is in 3NF and for every functional dependency $X \rightarrow Y$, X should be the superkey of the table.

OR,

A relational schema R is considered to be in Boyce Codd Normal Form (BCNF) if, for every one of its dependencies $X \rightarrow Y$, one of the following conditions holds true.

(i) $X \rightarrow Y$ is a trivial functional dependency (i.e. Y is a subset of X)
(ii) X is a superkey for schema R.

b) What is Normalization? Explain 1NF, 2NF, 3NF and 4NF. [2+4]

Ans Normalization is a process of organizing the data in database to avoid data redundancy, insertion anomaly, update anomaly & deletion anomaly. It is a multi-step process that puts data into tabular form by removing duplicated data from the relation tables. It is used for mainly two purposes.

i) Eliminating redundant (useless) data.
ii) Ensuring data dependencies make sense ie. data is logically stored.

Normalization rule are divided into following normal form.

1) First Normal Form (1NF)
2) Second Normal Form (2NF)
3) Third Normal Form (3NF)
4) BCNF

2nd Part

1NF : A database is in first normal if it satisfies the following conditions:

i) contains only atomic values
ii) There are no repeating groups.

It sets the basic rules for an organized database as follows:

i) Eliminate duplicate columns from the same table.
ii) Create separate tables for each group of related data and identify each row with a unique column or set of columns.

2NF: A database is in second normal form if it satisfies the following conditions:
i) It is in ~~first~~ 1NF.
ii) All non-key attributes are fully functional dependent on the primary key.

Converting 1NF to 2NF
- Identify the primary key for the 1NF relation
- Identify the functional dependencies in the relation
- If partial dependencies exist on the primary key remove them by placing them in new relation along with a copy of their determinant.

3NF: A database is in third normal form if it satisfies the following conditions
i) It is in 2NF
ii) There is no transitive functional dependency.

Converting 2NF to 3NF
- Identify the primary key in 2NF relation
- Identify functional dependencies in the relation.
- If transitive dependencies exist on the primary key remove them by placing them in a new relation along with a copy of their dominant.

4NF: A database is in fourth normal form, if it satisfies the following conditions
i) It should meet all the requirement of BCNF or 3NF
ii) Attribute of one or more rows in the table should not result in more than one rows of the same table leading to multivalued dependencies.

5) Explain the basic steps in query processing. Make distinctions between cost based optimization and heuristic optimization [4+4]

Ans Query processing refers to the range of activities involved in extracting data from a database. The activities include transaction of queries in high-level database languages into expressions that can be used at the physical level of the system, a variety of query-optimizing transformations, and actual evaluation of queries.

The steps involved in query processing are as follows:

i) Parsing and translation
ii) Optimization
iii) Evaluation.

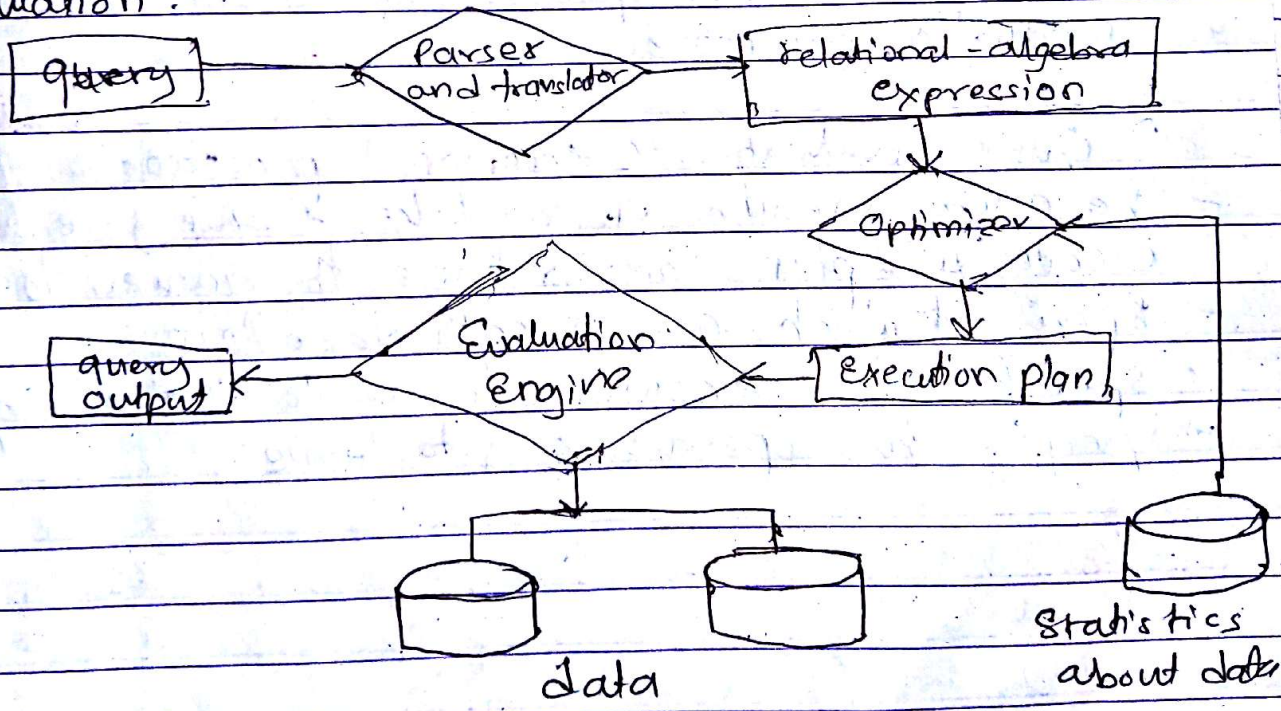


Fig: Steps in query processing

i) Query parsing and translation (query compiler)
- check the syntax. (e.g. SQL for relational DBMS)
- verify that the mentioned relations do exist and replace views.
- transform the SQL query to a query plan represented by a relational algebra expression.

ii) Query Optimization (Query Optimizer)
- transform the initial query plan into the best possible query plan based on the given data set.
- specify the execution of single query plan operations (evaluation primitives).
  E.g. which algorithms and indices to be used.
- The query execution plan is defined by a sequence of evaluation primitives.

iii) Query evaluation (command processor)
- The query-execution engine takes a query-evaluation plan, executes that plan, and returns the answers to the query.
- Specify which access path to follow
- specify which algorithm to use to evaluate operator
- specify how operators interleave.

Query Optimization is of two types.
1) Heuristic (Logical) query optimization.
2) Cost-based (physical) query optimization.

1) Cost-based Optimization:
→ Cost of physical plans includes processor time and communication time. The most important factor to consider is disk I/Os because it is the most time consuming action.
→ Steps in cost-based query optimization
   i) Generate logically equivalent expressions using equivalence rules.
   ii) Annotate resultant expressions to get alternative query plans.
   iii) Choose the cheapest plan based on estimated cost.
→ Estimation of plan cost based on
   - statistical information about relations.
   ⇒ Statistical estimation for intermediate results to compute cost of complex expressions.
   - Cost formulae for algorithms, computed using statistics.

Heuristic Optimization
→ Cost based optimization is expensive, even with dynamic programming.
→ Systems may use heuristics to reduce the number of choices that must be made in cost-based fashion.

→ Heuristic Optimization transforms the query-tree by using a set of rules that typically improve execution performance.

- Perform selection early (reduces the no. of tuples)
- Perform projection early (reduces the no. of attributes)
- Perform most restrictive selection and join operations (i.e. with smallest result size) before other similar operations.
- Some systems use only heuristics, others combine heuristics with partial cost-based optimization.

6 a) What is the use of RAID storage device? How is a record searched from a sparse sequential index? [2+8]

Ans. RAID (Redundant Arrays of Independent Disks) is a disk organization techniques that manage a large number of disks providing a view of a single disk of

- high capacity and high speed by using multiple disks in parallel
- high reliability by storing data redundantly, so that data can be recovered even if a disk fails.

Sparse Index

Sparse index contains index records for only some search-key values. It is applicable when records are sequentially ordered on search-key.

To search a record from a sparse sequential index following steps are followed.

~~i) First, proceed by index record and reach at the actual location of the data.~~

i) Find index record with largest search-key value $< k$.

To locate a record with search-key value k we:

i) Find index record with largest search-key value $< k$.

ii) Search file sequentially starting at the record to which the index record points.

OR.

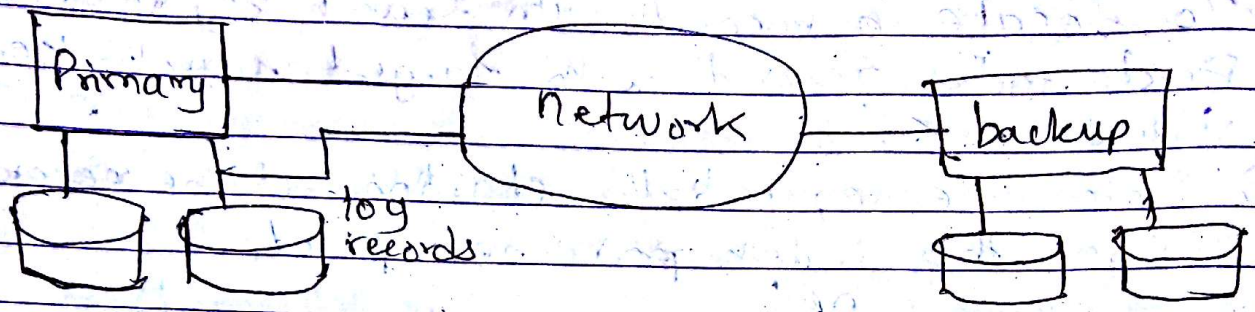
i) First, proceed by index record and reach at the actual location of the data.

ii) If the data we are looking for is not where we directly reach by following the index, then the system starts sequential search until the desired data is found.

b) Explain about the remote backup system with diagram. [3]

Ans Remote Backup System

→ Remote Backup system provide high availability by allowing transaction proceming to continue even if the primary site is destroyed.



→ Detection of Failure: Backup site must detect when primary site has failed.

→ Transfer of contra'.

- To take over control backup site first perform recovery using its copy of the database and all the long records it has received from the primary. Thus, completed transactions are redone and incomplete transactions are rolled back.
- When the backup site takes over processing it becomes the new primary
- To transfer control back to old primary when it recovers, old primary must receive redo logs from the old backup and apply all updates locally.

→ Time to recover: To reduce delay in takeover, backup site periodically processes the redo log records, ... perform a checkpoint and can then delete earlier parts of the log.
→ Hot-Spare configuration permits very fast takeover:
- Backup continually processes redo log record as they arrive, apply updates locally.
- When failure of the primary is detected the backup rolls back incomplete transactions, and is ready to process new transactions.

7a) What are schedules? Describe the concept of view serializability for concurrent execution of transactions. [2+4]

Ans Schedules are a sequences of instructions that specify the chronological order in which instructions of concurrent transactions are executed. It is of two types

i) Serial schedule
→ A schedule where the operations of each transaction are executed consecutively without any interleaved operations from other transactions.
→ There is no interference between transactions.

ii) Non serial Schedule
→ A schedule where the operations from a set of concurrent transactions are interleaved.

View Serializability

→ A schedule is view serializable if it is view equivalent to any serial schedule.

→ Every conflict serilizable schedule is also view serilizable

→ Below is a schedule which is view-serializable but not conflict serializable

T_{27}	T_{28}	T_{29}
read(Q)		
	write(Q)	
write(Q)		
		write(Q)

→ Let S and S' be two schedules with the same set of transactions. S and S' are view equivalent if the following three conditions are met, for each data item Q,

1) If in schedule S, transaction T_i reads the initial value of Q, then in schedule S' also transaction T_i must read the initial value of Q.

2) If in schedule S transaction T_i executes read(Q), and that value was produced by transaction T_j (if any), then in schedule S' also transaction T_i must read the value of Q that was produced by the same write(Q) operation of transaction T_j.

3) If the transaction (if any) that performs the final write(Q) operation in schedule S must also perform the final write(Q) operation in schedule S'.

As can be seen, view equivalence is also based purely on reads and writes alone.

b) How deadlocks arise while processing of transactions? Explain the deadlock prevention strategies. [2+4]

Ans In a multi-process system, deadlock is an unwanted situation that arises in a shared resource environment, where a process indefinitely waits for a resource that is held by another process. System is deadlocked if there is a set of ~~trans-~~ ~~act~~ transactions such that every transaction in the set is waiting for another transaction in the set.

~~Q8) Write the~~ Deadlock prevention strategies

~~⇒ to prevent any de~~

i) Predeclaration

→ Require that each transaction locks all its data items before it begins execution.

ii) Impose partial ordering of all data forms.

→ Require that a transaction can lock data items only in the order specified by the partial order.

iii) Timeout-Based Schemas.

→ a transaction waits for a lock only for a specified amount of time.

→ After the wait time is out and the transaction is rolled back.

→ simple to implement ; but starvation is possible.

→ difficult to determine good value of the timeout interval.

iii) Use time stamping:

a) Wait -die scheme - non-preemptive
- Older transaction may wait for younger one to release data item.
- Younger transactions never wait for older ones; they rolled back instead.
- A transaction may die several times before acquiring needed data item.

b) Wound -wait scheme - preemptive
- Older transaction wounds (forces rollback of) younger transaction instead of waiting for it.
- Younger transactions may wait for older ones.
- May be fewer roll backs than wait -die scheme.

8) Write two different types of failures that may occurs in system. Differentiate between shadow paging and log-based recovery.

Ans The different types of failures that may occurs in system are as follows:

i) Transaction failure

→ logical errors: transaction can't complete due to some internal error condition. Eg. Bad input, Data not found, Overflow.

ii) System Crash

→ Fail -stop assumption: non volatile storage contents are assumed to not be corrupted by system crash.

→ Database system have numerous integrity checks to prevent corruption of disk data.

iii) Disk failure:
→ A head crash or similar disk failure destroys all or part of disk storage.
→ Destruction is assumed to be detectable: disk drives use checksums to detect failures.

Shadow Paging Recovery	Log-based Recovery
i) It is an alternative to log-based recovery; this scheme is useful if transactions execute serially.	i) log is a sequence of records, which maintains the records of actions performed by a transaction.
ii) Here, the overhead of log-record output is eliminated, and recovery from crashes is significantly faster since no undo or redo operations are needed.	i) The search process is time consuming since most of the transaction are redone so, recovery take longer time.
iii) The commit of a single transaction requires multiple blocks to be output. The actual data block, the current page table, the disk address of current page table.	iii) Log based schemes need to output only the log-records, which, for typical small transactions, fit within one block.
iv) ~~Its~~ complexity is high.	iv) Complexity is low.
v) Garbage collection is necessary.	v) Not necessary
vi) Locality property of the page is lost.	vi) Locality of page is not lost