PREPARATION DBMS

Unit 1

1. Explain database and its components in detail.

2. Explain file and its types.

3. Explain operations on a file.

There are mainly two types of operations on file –

1. Retrieval Operation :-

When we want to extract the data from file for reading purpose then we perform retrieval operation on the file. Retrieval operations do not change the contents (data) of file; it only locates the records in file. Before we go ahead, we will see what is record??? In database management systems, a complete set of information is called as a ‘record’.

2. Update Operation :-

Update operation on the other hand; change the file by modifying the records, deleting the records and inserting new records. That is we actually modify the contents in file in update operation. The operations for locating, modifying, deleting and inserting records will vary from system to system

4. Explain a. Entity b. Attribute c. Domain d. Instance e. Tuple

**5. Explain data and its types **

unit 2

1. Write any 4 differences between DBMS and RDBMS

DBMS	RDBMS
DBMS stores data as file.	RDBMS stores data in tabular form.
Data elements need to access individually.	Multiple data elements can be accessed at the same time.
No relationship between data.	Data is stored in the form of tables which are related to each other.
Normalization is not present.	Normalization is present.
DBMS does not support distributed database.	RDBMS supports distributed database.

2 What is metadata?

Metadata is loosely defined as data about data. For example: a web page may include metadata specifying what language it's written in, what tools were used to create it, and where to go for more on the subject, allowing browsers to automatically improve the experience of users.

Metadata is defined as data providing information about one or more aspects of the data, such as:

1. Means of creation of the data
2. Purpose of the data
3. Time and date of creation
4. Creator or author of data
5. Placement on a computer network where the data was created
6. Standards used

For example, a digital image may include metadata that describes how large the picture is, the color depth, the image resolution, when the image was created, and other data. A text document's metadata may contain information about how long the document is, who the author is, when the document was written, and a short summary of the document.

Metadata is data about data.

As such, metadata can be stored and managed in a database, often called a registry or repository. However, it is impossible to identify metadata just by looking at it because a user would not know when data is metadata or just data.

3. Explain the structure of DBMS

4 What are the different types of users role in database environment.

5. Explain different services provided by DBMS over conventional file system.

6. What is RDBMS? Explain different relational db components.

7. Discuss different applications of DBMS.

---

unit 3

1. What is data model? Explain basic types of data model.

2. What is set operator? Explain set operations with example.

**3. What is foreign key? Explain foreign key ir referential intergrity constraint.
**A FOREIGN KEY in one table point to a PRIMARY KEY in another table. It is used to make sure referential integrity of the data. When we want to generate foreign key we must have two tables one is parent table & child table. Both tables have relationship with each other based on common column. A common column in both tables relates parent table & child table.

Syntax: - CREATE TABLE ChildTableName
(Column_Name Size,
Foreign Key (Common Key)
references ParentTableName (Common_Key));

4. Explain record based model with example.

5. What are joins ? List down the types of joins.

6. Explain types of integrity constraints.

**7. Explain hierarchical model with an example.
** Hierarchical data files permits records to be grouped together. This allows a parent-child relationship (a single one-to-many relationship) to be defined between records. In simple forms, the parent records are used to collect information that is common to all child records of the same group. This effect reduces redundancy with the database.

Therefore here organization of the records is as a collection of trees.

Figure2.7 shows a sample hierarchical database that is the equivalent of the relational database of Figure2.6.

Name	Street	City	Member
Jiten	Maple	Mumbai	990
Jolly	West	Nashik	336
Sudha	Sidehill	Pune	800
Jolly	West	Nashik	644
Sudha	Sidehill	Pune	644

Member	Balance
990	10000
336	25000
644	15000
800	40000

Figure: 2.6 Relational database

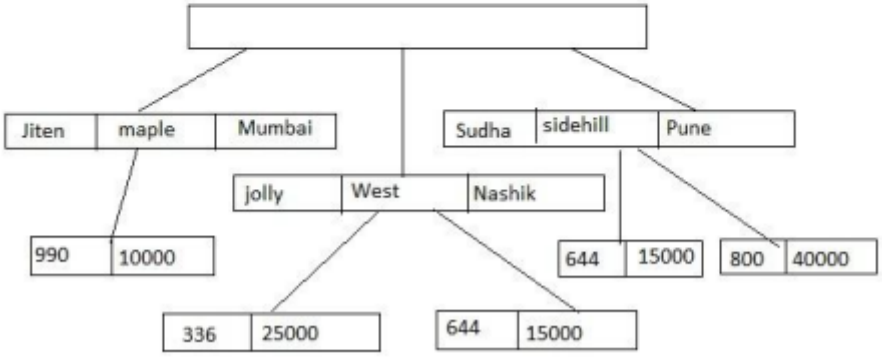


Figure2.7: A sample hierarchical database

In this example the data is sorted hierarchically, using a downward tree.

This model uses pointers to navigate between stored data. It was the first DBMS model.

---

unit 4

---

1. Whqt is relationship? Explain types of relationship.

**2. Explain with example weak and strong entity.
**An entity, which is dependent on one or more entities for its existence, is known as Weak entity. A weak entity is existence dependent because entity existence depends on one or more entities. The existence of a weak entity is indicated by a double rectangle in the ER diagram. An entity set that does not contain enough attributes to form a primary key is known as a weak entity set. In weak Entity set it is vital to identify attributes of weak entity set so we require a key it is known as discriminator or partial Key. The discriminator (or partial key) is used to categorize other attributes of a weak entity set.

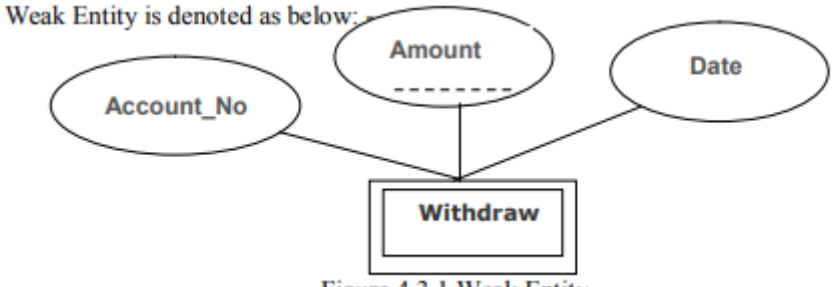


Figure.4.3.1 Weak Entity

4.3.2. Strong entity: -

An entity set that has a primary key is known as strong entity set. A strong entity is independent of other entities and can exist on its own.

Example: Student is strong entity type because it associates with Primary key Roll_No.

Figure. 4.3.2 Strong Entity set

3. Explain the concept of candidate key, super key and primary key with example.

a. Super Key: -

A Super Key is a set of one or more attributes that are taken collectively and can identify all other attributes uniquely. Super key stands for superset of a key.

Example: Student (RollNo, Name, Class, Address, Phone_No)
<RollNo, Name>
<RollNo, Name, Address> all these are super Key.

b. Candidate Key: -

Candidate Keys are super keys for **which no proper subset is a super key**.
Example: let suppose we have a Student Table with attributes
('RollNo', 'Name',' Address', 'Class','Phone_No')

In above table RollNo Key can identify the record uniquely and similarly combination of Name and Address can identify the record uniquely, but neither Name nor Address can be used to identify the records uniquely as it might be possible that we have two Students with similar name or two Students from the same Address. So for above table we have only two Candidate Keys used to identify the records from the table uniquely.

1. RollNo
2. Name, Address. These are Candidate Key

h. Primary Key: -

Primary key is an attribute in table that uniquely identifies each record in table and we cannot put primary key as null as well as duplicate. In the ER diagram, primary key is represented by underlining the primary key attribute. Usually primary key is composed of a single attribute

Example: Order (OrderID, Quantity, Price, Date).

In this case we will consider OrderID as a Primary Key .By using OrderID we can easily identify individual details of Orders uniquely.

Primary Key is Shown Below: -

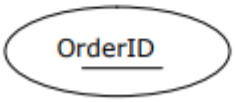


Figure. Primary Key OrderID

4. What is attribute? Explain any two attributes with example.

4.4.1. Simple Attribute: -

When attribute consist of a single atomic value it is Simple attribute.
i.e. Simple attribute cannot be subdivided.
Example: - The attributes age, Gender etc is simple attributes.

Figure. 4.4.1. Simple Attribute

4.4.2. Composite Attribute: -

When attribute value is not atomic it is composite attribute. A composite attribute is an attribute that can be further subdivided.

Example: The attribute ADDRESS can be subdivided into street, city, state, and zip code.

Address: Street: City: State: Zip Code
Name: 'First Name: Middle Name: Last Name'

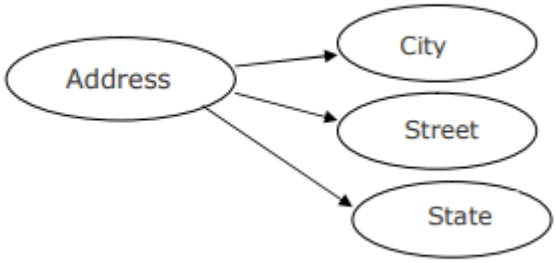


Figure. 4.4.2.Composite Attribute

5. What is keys? Explain any keys of relationship sets.

A key is an attribute or a combination of attributes that is used to identify records. Sometimes we might have to retrieve data from more than one table, in those cases it is required to join tables with the help of keys. The purpose of the key is to bind data together across tables without repeating all of the data in table

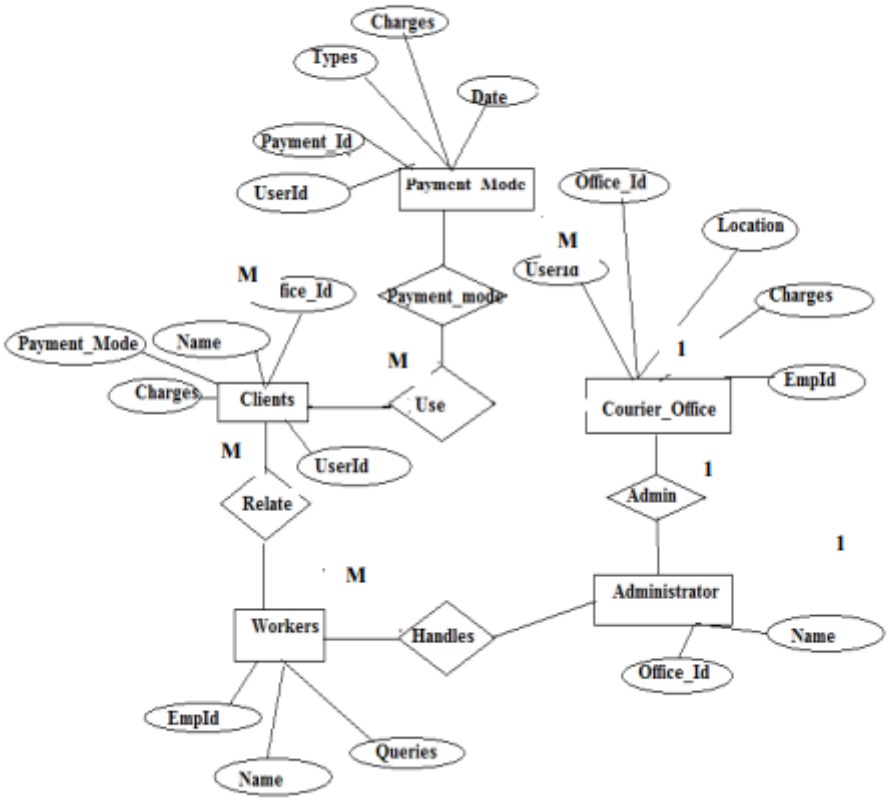
6. What is ER model? Explain the steps in ER model.

The Entity-Relationship model is developed by Chen in 1976 to design database more effectively. A Conceptual data model explains the structure of a database. The fundamental concept of ER model includes entity types, relationship type and attributes. Entity type represents a set of object in actual world with similar properties. Let's see all components of ER Model.

Steps in ER Modeling: -

1. Identify the Entities.
2. Find relationships.
3. Identify the key attributes for every Entity.
4. Identify other relevant attributes.
5. Draw complete E-R diagram with all attributes including Primary Key

7). draw ER diagram for courier service system



unit 5

1.What is normalization? Why we need normalization?

1. Normalization is the process of structuring and handling the relationship between data to minimize redundancy in the relational table and avoid the unnecessary anomalies properties from the database like insertion, update and delete.
2. It helps to divide large database tables into smaller tables and make a relationship between them. It can remove the redundant data and ease to add, manipulate or delete table fields.
3. The main reason for normalizing the relations is removing these anomalies.
4. Failure to eliminate anomalies leads to data redundancy and can cause data integrity and other problems as the database grows.
5. Normalization consists of a series of guidelines that helps to guide you in creating a good database structure

**

2. What is functional dependency and its types? Explain with example.

1. Full Functional Dependency:-

If we have Functional Dependency P->Q then it can be Full Functional Dependency only when we cannot remove any attribute of Q from P.In other words Q is fully functionally dependent on P and there should not be any R->Q, where R is a proper subset of P.

Example: -Determine functional dependency in Mark sheet table given below.

RollNo	Subject_Code	Marks
101	C110	82
101	C112	45
102	C122	65
103	C123	70

Solution: -Mark Sheet (RollNo, Subject_Code, Marks).

In this case **(RollNo, Subject_Code)** both form composite key (primary key with more than one attribute) means if we want to find marks for particular subject we need RollNo and Subject_ Code.

I.e. Functional Dependency exists in this relation as follows:-

Marks->RollNo, Subject_Code

**

**3. What is denormalization?

**De-normalization is the process of converting normalized form into unnormalized form. It is a technique to move from higher to lower normal forms of database. We used denormalization for following purposes: -

To optimize the performance of a database.
To speed up database access
Sometime users want to access data in different ways.
In database many critical queries and reports exist which depends upon data from more than one table.
Sometime users want to apply many calculations to one or many columns for expected result.

4. What is decomposition or small schema with an example.Explain its types.

In general decomposition means to break down particular thing into many pieces. Similar to that we decompose relation into several relations to avoid redundancy. When we decompose relation it is required to check that decomposition does not lead to bad design.

Definition: -Let R be a relation schema

A set of relation schemas {R1, R2, …, Rn} is a decomposition of R if

- R = R1 U R2 U …..U Rn
- Each Ri is a subset of R (for i = 1,2…n)

Example: Consider relation R (x, y, z) :
We decompose R into two subset R1 and R2 as below

R1 (x, z) and R2 (y, z)

If we union R1 and R2, we get R

R = R1 U R2

In decomposition, if we decompose tables then it may happen that, we may not be able to reconstruct the corresponding instance of the original relation it means **information loss** in decomposition appear**.**

Example: -Consider Following Table for decomposition
Give Relation R we decompose it into two subset R1 and R2 .If we try to reconstruct R from R1 U R2 ,we cannot get original R so there is information loss in decomposition.

RollNo	Fees	Grade
A11	15000	A+
S20	4000	B
A70	1500	A+

Decomposition

RollNo	Grade
A11	A+
S20	B
A70	A+

R1

Fees	Grade
15000	A+
4000	B
25000	A+

R2

**5.

Example: Determine the functional dependencies. Remove partial dependency and transitive dependencies in Figure 5.7.4.3 i.e. convert it into 3NF.

Student = (RollNo, Name, Course_Code, Course_Name, Fees)

RollNo	Name	Course_Code	CourseName	Fees
123	Ravi	C102	C	2500
123	Ravi	C103	C++	1200
123	Ravi	C104	OOPs	3200
124	Sumit	C102	C	2500
124	Sumit	C103	C++	1200
125	Trupta	C102	C	2500
125	Trupta	C103	C++	1200
125	Trupta	C104	OOPs	3200

Figure. 5.7.4.3 Student table

Solution: -Remove transitive dependencies.

In above example Name is depend upon Primary key RollNo and other attribute Fee is depend upon CourseName, which is depend upon Course_Code.

RollNo -> Name

Transitive dependencies: - Course_Code->CourseName->Fees

First check is their partial dependents or transitive dependencies exist in given relation? If transitive dependencies exist then remove it by divide the relation into new relations.

Personal (RollNo, Name, Course_Code)

Course (Course_Code, CourseName, Fees)

**

**6. What are the advantages and disadvantages of normalization.
**

7. Explain types of normalization with example.
Types of Normalization: -

**

Un-Normalized Form :- **It is a relation that contains non-atomic values. A non-atomic value means it can be decomposed .To create UN normalize form we must list attribute of entity, identify repeating groups of attributes, their keys, identify main keys.

First Normal Form (1 NF) :- For 1NF it is must that each cell of the table must have single value i.e. atomic. Every row of table contains exactly one value for each attribute. No two rows in a table may have similar values. For converting table in 1NF we must follow rules: - 1. Remove duplicate columns from the same table. 2. After removing duplicate columns, you have to make separate tables for each group of related data and recognize each row with a unique column (the primary key)

Second Normal Form (2 NF) :- After producing first normal form we found redundant data in tables so we use second normal form to normalize it again. Second normal form is use to reduce the amount of

redundant data in a table. This is done by extracting it and placing that redundant data in new table(s) and creating relationships between those tables. We can say that a relation R is said to be in 2NF if relation is in 1NF and every attribute is depend on Primary key.

**

Third Normal Form (3 NF) :- **In 2NF we found some columns that are not fully dependent upon primary keys exist in table so we need to convert relation into 3NF .In 3NF we first check that relation satisfies 1NF and 2NF, after that we remove columns that are not fully dependent upon the primary key. i.e. we remove transitive dependencies. So we can say that a relation R is in 3NF, if it is in 2NF and it has no transitive dependency.

---

unit 6

1.Explain any five datatypes in SQL.

A data type is a set of data with values having predefined characteristics. A data type is a type of data. In SQL a wide range of data types is present. By using different data types we can store data in different forms, including character strings, numbers, file and date.
Example: -For name of student, we may use a string data type for the student's first and last name.

char(n)	char(n)	Max 8,000 Char	Fixed-length character string.
varchar(n)	varchar(n)	Max 8,000 Char	Variable-length character string.
text	text	Max 2GB of text data	Variable-length character string.
bit	bit(x)		Allows 0, 1, or NULL
date	date	8 bytes	Stores year, month, and day values.
datetime	datetime	8 bytes	Store Date & Time in numeric format Eg. 2007-10-08 12:35:30.123

2. Write a short note on primary key with an example.

A primary key is used to uniquely identify each row in a table. It can be used either when the table is created using create table statement or by changing the existing table structure using alter table. It consists of one or more fields on a table. When multiple fields are used as a primary key, they are called a composite key.

Example: - CREATE TABLE Student
****** (RollNo integer PRIMARY KEY,**
****** FirstName varchar (30),**
LastName varchar (30));

3. What is database constraints.Explain check constraint with an example.

4. What is DDL? Explain any two DDL with example.

DDL: Data Definition Language

The Data Definition Language (DDL) is used to create and destroy databases and database objects. Database administrators will primarily use these commands during the setup and removal phases, of a database project.

DDL statements are used to define the database structure or schema. In below given example there are some DDL commands:

CREATE - to create objects in the database
ALTER - alters the structure of the database
DROP - delete objects from the database
TRUNCATE - remove all records from a table, including all spaces allocated for the records are removed
COMMENT - add comments to the data dictionary
RENAME - rename an object

5. Define with example and syntax group by clause.

6. Explain any six types of aggregate functions with example.

```
Select Count (Amount) From Student
Or                                          Output: - 5
Select Count (*) from Student

Select First (Amount) From Student          Output: - 15000
```



```
Select Last (Amount) From Student           Output: - 18000

Select Max (Amount) From Student            Output: - 25000

Select Min (Amount) From Student                    Output: - 15000
```

Syntax: -Now ()

```
SELECT NOW () FROM table_name
```

Example: -

```
SELECT RollNo, Amount, Now () as Date
FROM Student where RollNo=1021
```

Output: -

RollNo	Amount	Date
1021	15000	20/1/2010 1:35:02 AM

7. What is string operations? Explain any 5 string operations with example.

A string is a finite sequence of symbols that are chosen from a set or alphabet.

4. REPLACE: -
Replaces all occurrences of the string2 in the string1 with string3.

Syntax - REPLACE ('string1', 'string2', 'string3')

Example: -SELECT REPLACE ('Last_Name '),'til','tel')
FROM Students WHERE Last_Name = ' Patil

Output: -Patel

**

**

6.LEFT: -
Returns left part of a string with the specified number of characters.

Syntax - LEFT (string, integer)

Example: -SELECT LEFT ('Last_Name ', 3)
FROM Students WHERE Last_Name = ' Shah' Output: - Sha

7. RIGHT: -
Returns right part of a string with the specified number of characters.

Syntax - RIGHT (string, integer)

Example: -SELECT RIGHT ('Last_Name ', 3)
FROM Students WHERE Last_Name = ' Shah

Output: -hah

8. REPLICATE: -
Repeats string for a specified number of times.

Syntax - REPLICATE (string, integer)

Example: -SELECT REPLICATE ('Last_Name ', 2)
FROM Students WHERE Last_Name = ' Shah'

Output: -ShahShah

10.UPPER: -
Convert string to Uppercase.

Syntax - UPPER (string)

Example: -SELECT UPPER ('Last_Name ')
FROM Students WHERE Last_Name = ' Shah'
Output: -SHAH

unit 7

1. Explain transaction state by giving an example.

6.4 Transaction States

The transaction may be in one of the following states while executing:

1) **Active**: This is the initial state. The transaction is in this state while it is executing.
2) **Partially Committed**: The entire transaction has been executed, but not yet committed.
3) **Failed**: The transaction cannot execute normally.
4) **Aborted**: The transaction is rolled back and the database is stored to its prior state.
5) **Committed**: After a successful completion.

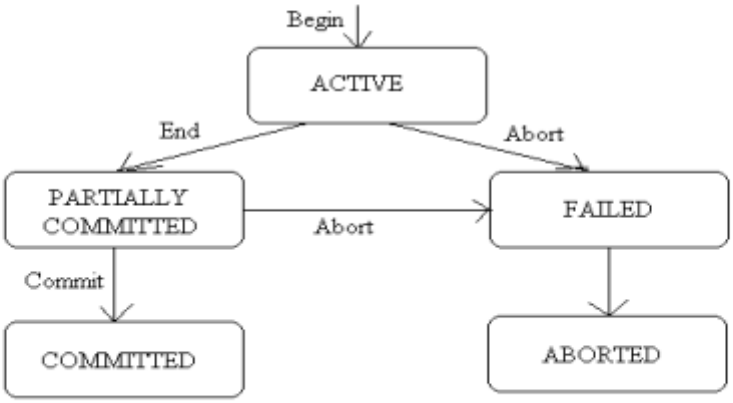


Figure 6.3: Transaction States.

**

2. Explain ACID properties in detail.**

- **ACID Properties**
  - **Atomicity**: A transaction is an atomic unit of processing; it is either performed in its entirety or not performed at all.
  - **Consistency Preservation**: A correct execution of a transaction must take the database from one consistent state to another.
  - **Isolation**: A transaction should not make its updates visible to other transactions until it is committed.
  - **Durability**: Once a transaction changes the database and the changes are committed, these changes must never be lost.

3. What is schedule? Explian recoverable schedule.

Schedules: Schedules are used to represent the sequence in which the instructions are executed in the system

Recoverable Schedule: A Recoverable Schedule is a schedule in which, if a transaction Tj reads a data item previously written by a transaction Ti, then the commit operation of Ti should appear before the commit operation of Tj.

**4. Write the following operations on transactions Read-Item(X) Write-Item(X)
**

1) Read-item (X): Reads the database item named X into a program variable. The Read-item (X) includes the following steps:

Find the address of disk block that contains item X.

Copy the contents of disk block into a buffer in main memory.

Copy the item X from the buffer to the program variable named X.

2) Write-item (X): Writes the value of a program variable X into the database item named X. Write-item (X) includes the following steps:

Find the address of the disk block that contains item X.

Copy the contents of disk block into a buffer in main memory.

Copy item X from program variable named X into its correct location in the buffer.

Store the updated block from the buffer back to disk.

5 Explain in detail concurrency control multiple update problems in detail.

**6. Explain the terms

---

1. Cascadeless rollback :- A single transaction failure leads to a series of transaction rollbacks. This is called as Cascading Rollback.
2. Strict schedule :- In this schedule, transactions cannot read or write a data item 'X' until the last transaction that wrote the data item 'X' has committed or aborted.
3. Cascadeless schedule :- For each pair of transactions Ti and Tj such that Tj reads a data item previously written by Ti, then the commit operation of Ti should appear before the read operation of Tj. This schedule is called as Cascadeless Schedule. **

7. Explain recoverabiliry in detail.

If a transaction fails we have to undo the effect of the transaction and bring the database to the consistent state. So when a transaction aborts, all the transactions dependent on the aborted transaction should also be aborted and rolled back. To achieve this, there are certain restrictions imposed on the type of schedules permitted in the system. To bring the database back to a consistent state, there is a limit on the acceptable schedules. We will now see the types of schedules:
