NOTES LINK

unit-3,4-dbms

1. What do you mean by functional dependency?

The functional dependency is a relationship that exists between two attributes. It typically exists between the primary key and non-key attribute within a table.

X → Y

The left side of FD is known as a determinant, the right side of the production is known as a dependent.

Assume we have an employee table with attributes: Emp_Id, Emp_Name, Emp_Address.

Here Emp_Id attribute can uniquely identify the Emp_Name attribute of employee table because if we know the Emp_Id, we can tell that employee name associated with it.

2. What do you mean by transitive functional dependency?

When an indirect relationship causes functional dependency it is called Transitive Dependency.

If P -> Q and Q -> R is true, then P-> R is a transitive dependency.

3. What is partial functional dependency?

Partial Functional Dependency :
A functional dependency X->Y is a partial dependency if Y is functionally dependent on X and Y can be determined by any proper subset of X.
For example, we have a relationship AC->B, A->D, and D->B.
Now if we compute the closure of {A+}=ADB
Here A is alone capable of determining B, which means B is partially dependent on AC.

4. What is full functional dependency?

Fully Functional Dependency :
If X and Y are an attribute set of a relation, Y is fully functional dependent on X, if Y is functionally dependent on X but not on any proper subset of X.
Example –
In the relation ABC->D, attribute D is fully functionally dependent on ABC and not on any proper subset of ABC. That means that subsets of ABC like AB, BC, A, B, etc cannot determine D.

5. Explain 2NF with example.

Second Normal Form (2NF):

Second Normal Form (2NF) is based on the concept of full functional dependency. Second Normal Form applies to relations with composite keys, that is, relations with a primary key composed of two or more attributes. A relation with a single-attribute primary key is automatically in at least 2NF. A relation that is not in 2NF may suffer from the update anomalies.

To be in second normal form, a relation must be in first normal form and relation must not contain any partial dependency. A relation is in 2NF if it has No Partial Dependency, i.e., no non-prime attribute (attributes which are not part of any candidate key) is dependent on any proper subset of any candidate key of the table.

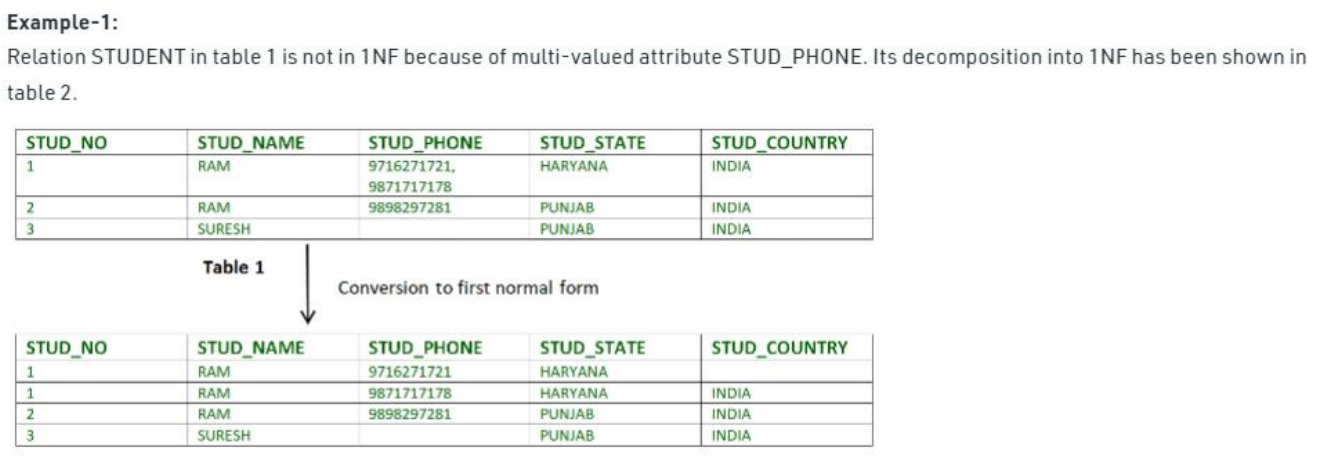
6. Explain 1NF with example.

First Normal Form (1NF):

If a relation contains a composite or multi-valued attribute, it violates the first normal form, or the relation is in first normal form if it does not contain any **composite** or **multi-valued attribute**. A relation is in first normal form if every attribute in that relation is singled valued attribute.

A table is in 1 NF iff:

1. There are only Single Valued Attributes.
2. Attribute Domain does not change.
3. There is a unique name for every Attribute/Column.
4. The order in which data is stored does not matter.



Example-1:

Relation STUDENT in table 1 is not in 1NF because of multi-valued attribute STUD_PHONE. Its decomposition into 1NF has been shown in table 2.

STUD_NO	STUD_NAME	STUD_PHONE	STUD_STATE	STUD_COUNTRY
1	RAM	9716271721, 9871717178	HARYANA	INDIA
2	RAM	9898297281	PUNJAB	INDIA
3	SURESH		PUNJAB	INDIA

Table 1

$\downarrow$ Conversion to first normal form

STUD_NO	STUD_NAME	STUD_PHONE	STUD_STATE	STUD_COUNTRY
1	RAM	9716271721	HARYANA	
1	RAM	9871717178	HARYANA	INDIA
2	RAM	9898297281	PUNJAB	INDIA
3	SURESH		PUNJAB	INDIA

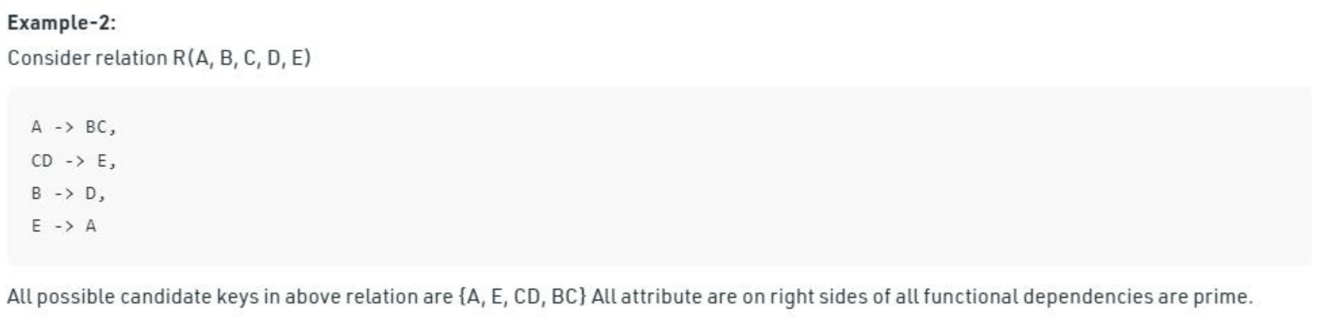
7. Explain 3NF with example.

Third Normal Form (3NF):

A relation is in third normal form, if there is no transitive dependency for non-prime attributes as well as it is in second normal form.

A relation is in 3NF if at least one of the following condition holds in every non-trivial function dependency X –> Y:

1. X is a super key.
2. Y is a prime attribute (each element of Y is part of some candidate key).


Example-2:
Consider relation R(A, B, C, D, E)

```
A -> BC,
CD -> E,
B -> D,
E -> A
```

All possible candidate keys in above relation are {A, E, CD, BC} All attribute are on right sides of all functional dependencies are prime.

8. Explain BCNF with example.

BCNF (Boyce Codd Normal Form) is the advanced version of 3NF. A table is in BCNF if every functional dependency X->Y, X is the super key of the table. For BCNF, the table should be in 3NF, and for every FD. LHS is super key.

Example

Consider a relation R with attributes (student, subject, teacher).

Student	Teacher
Jhansi	P.Naresh
jhansi	K.Das
subbu	P.Naresh
subbu	R.Prasad

F: { (student, Teacher) -> subject

(student, subject) -> Teacher

Teacher -> subject}

9. Explain 4NF with example.

A relation will be in 4NF if it is in Boyce Codd normal form and has no multi-valued dependency.

For a dependency A → B, if for a single value of A, multiple values of B exists, then the relation will be a multi-valued dependency

Example

STUDENT

STU_ID	COURSE	HOBBY
21	Computer	Dancing
21	Math	Singing
34	Chemistry	Dancing
74	Biology	Cricket
59	Physics	Hockey

The given STUDENT table is in 3NF, but the COURSE and HOBBY are two independent entity. Hence, there is no relationship between COURSE and HOBBY.

In the STUDENT relation, a student with STU_ID, **21** contains two courses, **Computer** and **Math** and two hobbies, **Dancing** and **Singing**. So there is a Multi-valued dependency on STU_ID, which leads to unnecessary repetition of data.

So to make the above table into 4NF, we can decompose it into two tables:

STUDENT_COURSE

STU_ID	COURSE
21	Computer
21	Math
34	Chemistry
74	Biology
59	Physics

STUDENT_HOBBY

STU_ID	HOBBY
21	Dancing
21	Singing
34	Dancing
74	Cricket
59	Hockey

10. Explain the difference between 3NF and BCNF.

S.NO.	3NF	BCNF
1.	In 3NF there should be no transitive dependency that is no non prime attribute should be transitively dependent on the candidate key.	In BCNF for any relation A->B, A should be a super key of relation.
2.	It is less stronger than BCNF.	It is comparatively more stronger than 3NF.
3.	In 3NF the functional dependencies are already in 1NF and 2NF.	In BCNF the functional dependencies are already in 1NF, 2NF and 3NF.
4.	The redundancy is high in 3NF.	The redundancy is comparatively low in BCNF.
5.	In 3NF there is preservation of all functional dependencies.	In BCNF there may or may not be preservation of all functional dependencies.
6.	It is comparatively easier to achieve.	It is difficult to achieve.
7.	Lossless decomposition can be achieved by 3NF.	Lossless decomposition is hard to achieve in BCNF.

11. What is Multivalued dependency?

Multivalued Dependency

- Multivalued dependency occurs when two attributes in a table are independent of each other but, both depend on a third attribute.
- A multivalued dependency consists of at least two attributes that are dependent on a third attribute that's why it always requires at least three attributes.

Example: Suppose there is a bike manufacturer company which produces two colors(white and black) of each model every year.

BIKE_MODEL	MANUF_YEAR	COLOR
M2011	2008	White
M2001	2008	Black
M3001	2013	White
M3001	2013	Black
M4006	2017	White
M4006	2017	Black

Here columns COLOR and MANUF_YEAR are dependent on BIKE_MODEL and independent of each other.

In this case, these two columns can be called as multivalued dependent on BIKE_MODEL. The representation of these dependencies is shown below:

1. BIKE_MODEL → → MANUF_YEAR
2. BIKE_MODEL → → COLOR

This can be read as "BIKE_MODEL multidetermined MANUF_YEAR" and "BIKE_MODEL multidetermined COLOR".

12. Find the Normal Form

R(ABCDE)

AB->C

C->E

D->A

Video link

13. Find the Normal

formR(ABCDE)

AB->C

C->DE

D->A

14. Find Normal Form

Emp(ename, did, empid, salary)
ename, did->salary
did, empid-> salary
ename -> empid
empid->ename

15. Find the Normal

R(A,B,C,D,E)

A->BC

BC->E

E->DA

16. Find the normal

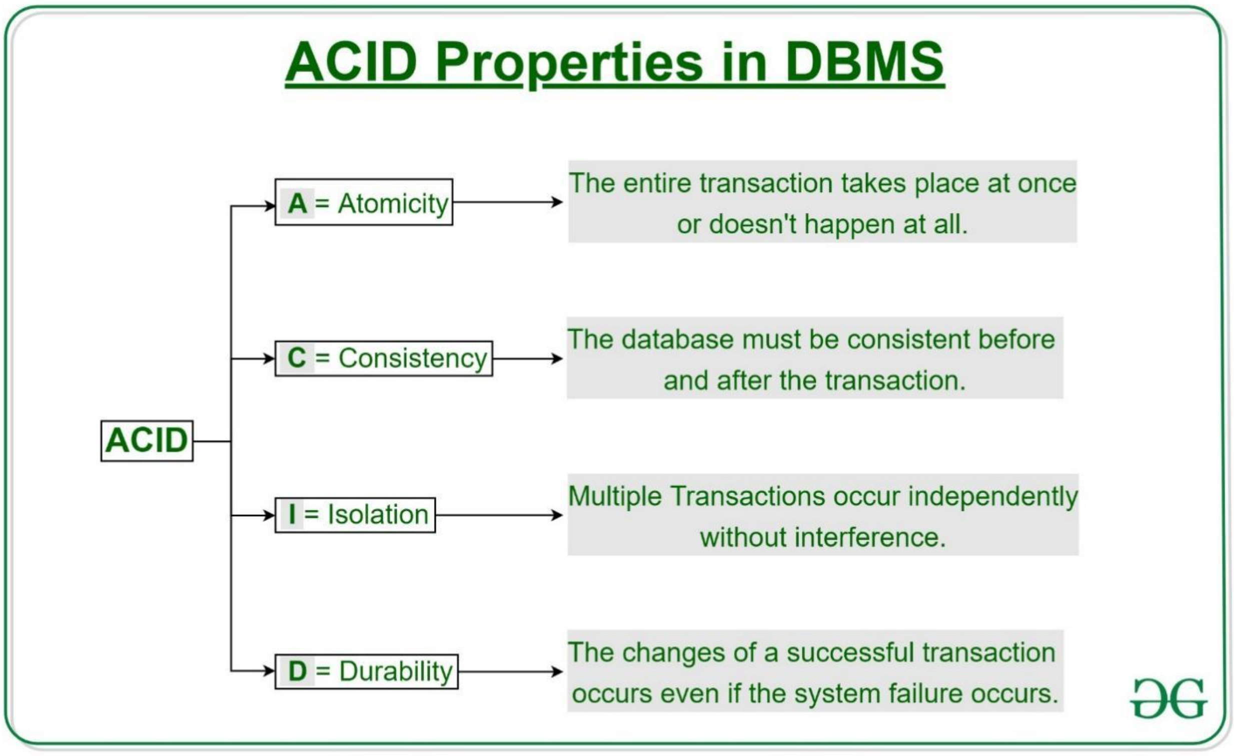
R(ABCDE)

A->D

BC->E DE->A

17. R(C, S, Z)
CS→Z
Z→C
Find the list of candidate key.

18. Explain ACID Property.


ACID Properties in DBMS
ACID
A = Atomicity
The entire transaction takes place at once or doesn't happen at all.
C = Consistency
The database must be consistent before and after the transaction.
I = Isolation
Multiple Transactions occur independently without interference.
D = Durability
The changes of a successful transaction occurs even if the system failure occurs.

19. What is serializability?

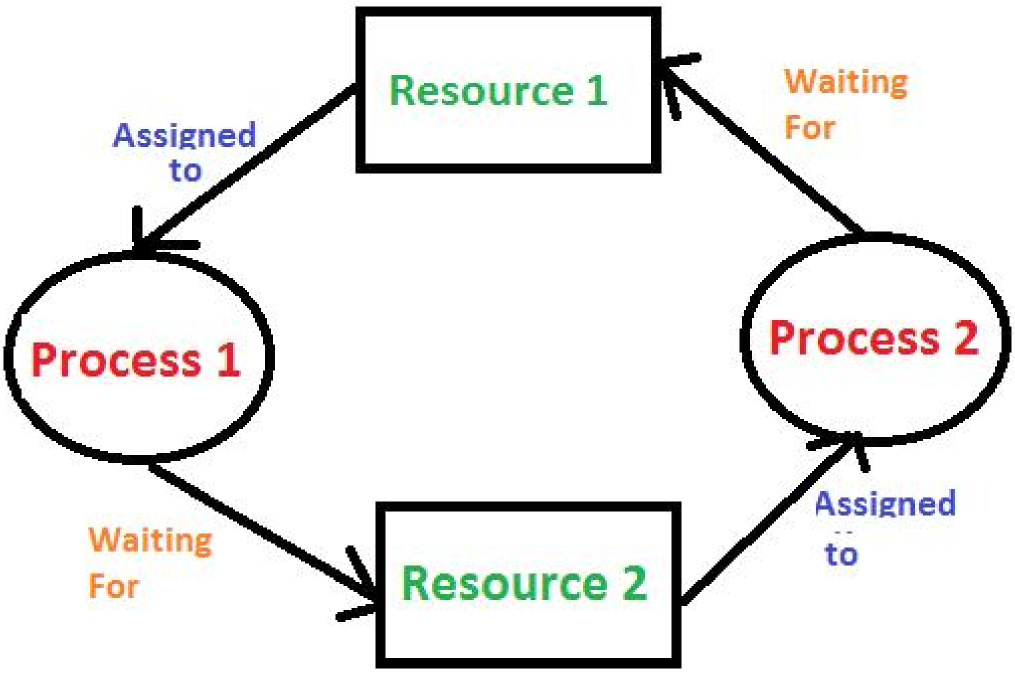
Serializability of schedules ensures that a non-serial schedule is equivalent to a serial schedule. It helps in maintaining the transactions to execute simultaneously without interleaving one another. In simple words, serializability is a way to check if the execution of two or more transactions are maintaining the database consistency or not.

Types of Serializability

1. Conflict Serializability
2. View Serializability

20. What do you mean by deadlock?

Deadlock is a situation where a set of processes are blocked because each process is holding a resource and waiting for another resource acquired by some other process.
Consider an example when two trains are coming toward each other on the same track and there is only one track, none of the trains can move once they are in front of each other. A similar situation occurs in operating systems when there are two or more processes that hold some resources and wait for resources held by other(s). For example, in the below diagram, Process 1 is holding Resource 1 and waiting for resource 2 which is acquired by process 2, and process 2 is waiting for resource 1.



Deadlock can arise if the **following four conditions hold simultaneously (Necessary Conditions)**
Mutual Exclusion: Two or more resources are non-shareable (Only one process can use at a time)
Hold and Wait: A process is holding at least one resource and waiting for resources.
No Preemption: A resource cannot be taken from a process unless the

process releases the resource.

Circular Wait: A set of processes are waiting for each other in circular form.

21. Define transaction with example.

A transaction can be defined as a group of tasks. A single task is the minimum processing unit which cannot be divided further.

Let's take an example of a simple transaction. Suppose a bank employee transfers Rs 500 from A's account to B's account. This very simple and small transaction involves several low-level tasks.

A's Account

```
Open_Account(A)
Old_Balance = A.balance
New_Balance = Old_Balance - 500
A.balance = New_Balance
Close_Account(A)
```

B's Account

```
Open_Account(B)
Old_Balance = B.balance
New_Balance = Old_Balance + 500
B.balance = New_Balance
Close_Account(B)
```

22. Explain conflict serializable schedule.

A schedule is called conflict serializability if after swapping of non-conflicting operations, it can transform into a serial schedule.

The schedule will be a conflict serializable if it is conflict equivalent to a serial schedule.

23. R(A, B, C)

$A \rightarrow B$
$B \rightarrow C$

Find the closure of each attribute.

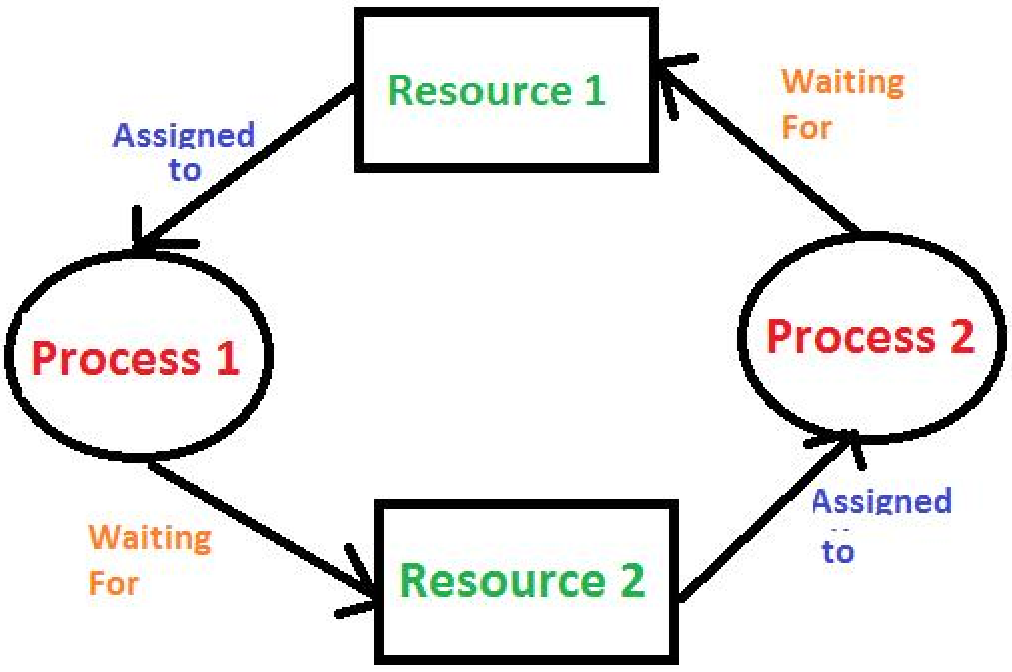
24.What do you mean by deadlock?
ans-20

25. Explain Deadlock detection.

Deadlock Detection :

1. If resources have a single instance –

In this case for Deadlock detection, we can run an algorithm to check for the cycle in the Resource Allocation Graph. The presence of a cycle in the graph is a sufficient condition for deadlock.



2. In the above diagram, resource 1 and resource 2 have single instances. There is a cycle R1 → P1 → R2 → P2. So, Deadlock is Confirmed.

3. If there are multiple instances of resources –

Detection of the cycle is necessary but not sufficient condition for deadlock detection, in this case, the system may or may not be in deadlock varies according to different situations.

Deadlock Recovery :

A traditional operating system such as Windows doesn’t deal with deadlock recovery as it is a time and space-consuming process. Real-time operating systems use Deadlock recovery.

1. **Killing the process –**
   Killing all the processes involved in the deadlock. Killing process one by one. After killing each process check for deadlock again keep repeating the process till the system recovers from deadlock. Killing all the processes one by one helps a system to break circular wait condition.
2. **Resource Preemption –**
   Resources are preempted from the processes involved in the deadlock, preempted resources are allocated to other processes so that there is a possibility of recovering the system from deadlock. In this case, the system goes into starvation.

26. What is serial execution in transaction?

__Serial execution__ – In serial execution, the second transaction can begin its execution only after the first transaction has completed. This is possible on a uniprocessor system.

27. Explain conflict serializable schedule.

- A schedule is called conflict serializability if after swapping of non-conflicting operations, it can transform into a serial schedule.
- The schedule will be a conflict serializable if it is conflict equivalent to a serial schedule.

Conflicting Operations

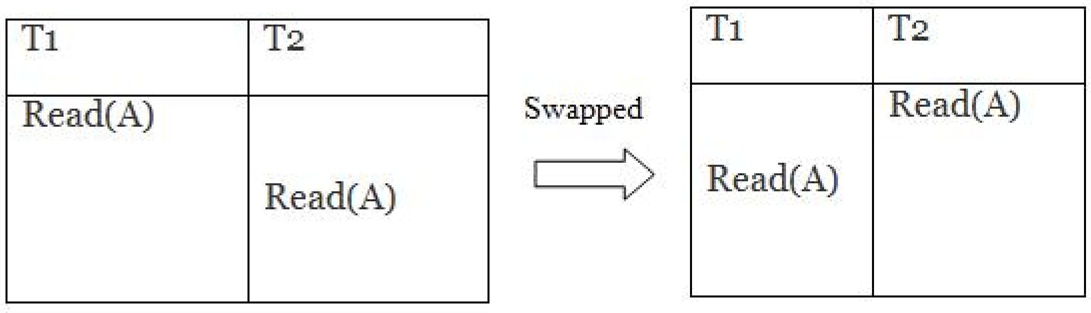
The two operations become conflicting if all conditions satisfy:

1. Both belong to separate transactions.
2. They have the same data item.
3. They contain at least one write operation.

Example:

Swapping is possible only if S1 and S2 are logically equal.

1. T1: Read(A)   T2: Read(A)



T1	T2
Read(A)	
	Read(A)

T1	T2
	Read(A)
Read(A)	

Schedule S1　　　　**Schedule S2**

Here, S1 = S2. That means it is non-conflict.

28. What is concurrent execution in transaction?

Concurrent transaction or execution includes multiple transactions which are executed concurrently or simultaneously in the system.

Advantages

The advantages of the concurrent transactions are as follows −

- Increases throughput which is nothing but number of transactions completed per unit time.
- It reduces the waiting time.

Example

T1= 90sec

T2= 500sec

T3= 5sec.

If we execute serially by T1->T2->T3 then transaction T3 waits for 590 sec, so we go for non-serial or concurrent transactions to reduce waiting time.

i.e. T3 -> T1 -> T2.

Disadvantage

The disadvantage is that the execution of concurrent transactions may result in inconsistency.

29. check the given schedule is serializable or not.
Consider the following schedule for transaction T1, T2, T3

T1	T2	T3
r(x)		
	r(y)	
		r(y)
	w(y)	
w(x)		
		w(x)
	r(x)	
	w(x)	

30?. Check the give schedule is serializable or not. S:- R1[X] R2[X] R2[Y] W1[X] W1[Y] W2[Y]