Copy of DatabasesThis discussion will be at 8.30 on 30/12 on voice.

- ER-model, Relational model (relational algebra, tuple calculus)
- Database design (integrity constraints, normal forms)
- Query languages (SQL)
- File structures (sequential files, indexing, B and B+ trees)
- Transactions and concurrency control

procedure. : today will be a voice discussion and handwritten notes will be uploaded. Notes will be explained on voice first then we may solve questions on them.

Plz don't ask random question during explanation. i will try to cover all the possible questions.

first we will cover the ER diagram. As no expert is available after completion of ER diagram. Any other interested volunteer may . explain further. **or i will explain as much i can . the problem i will not be sure about will be added to Unsolved tag and if anyone of u knows can help in solving them.**

please don't write between notes and follow the procedure of first come first serve ,

What i suggest is something is better than nothing. So we should start on our own.

How to convert ER to Relational model?
Is there a unique relational model for a given ER model?
While converting from ER to relational model can we find the minimum no. of tables unless we are given a particular normal form?

Normalization- how to find FDs?. Converting from 1NF to 2NF, 3NF, BCNF.

Lossless decomposition:

Dependency preserving decomposition:

Multi-valued dependency:

Foreign key - must be present in parent table.

SQL:

Joins:

Divide operation:

Group By/Having:

NULL issues:

Relational Algebra: Same power as SQL. Projection vs Select.

Tuple calculus- Revising First Order Logic.

Sequential Files

B+ trees- Advantages over B-Trees

Types of Indexes.

Transactions

ACID Property:

concurrency Control:

ER DIAGRAM .
so e r diagram is the graphical representation of what the requirement is .
we just go to client and take the requirement and make the er diagram as t is a good way to make the client understand in a easy way . and it is also easy for the developers to go and do it after the er diagram, is finalized .

so as we all know ER model stands for Entity Relationship model .
entity as we have read is an object in real world .

entity can be a physical or virtual like the car . car is an physical entity while the bank account is an virtual entity . as we don't have bank account in real world. can we see our bank account . answer is no , we can't that's why the entity can be classified in two groups tangible and intangible .
tangible - PHYSICAL one
intangible - virtual one .

so we have various types of attribute in er diagram . we can have multivalued attribute . well i m not going to explain the obvious things . this everyone will be knowing . if u not know then just google it out .
multivalued attribute - the attribute that have more than one value .
like phone number .
single value . as date of birth .
derived attribute . like age . the derived attributes are the one that can be derived from other attributes like the age can be derived from date of birth . by subtracting your date of birth from present day . we do not store derived attributes in database because the derived attribute can be derived at any point of time so why store them . (storing might give performance)

2- some change regularly
like the age . let's suppose i stored your age then i have to update it everyday . as it will change everyday , so better don't save it . derive it as u need it .

@ ravi sir Is that also the reason that we don't make separate table for derived attributes While converting from ER to relational model ?

so after making the er diagram we start making the tables from it . so first of all before going to the tables. again review some good and important point about er diagram like the cardinality of a an entity . so i am uploading an image, then we should see what this is

ok i m n a slow connection
so let me tell in words only

like suppose there is an entity is employee and a department . verb in real world can be used to say the relationship .

like here the relationship can be works for .
so employee(entity) ------- works for -------- department .

so if i say that any employee can only work in one department while many same department can have a large number of employee.
so cardinality of an entity means what is the maximum number of attributes in the second set it can connect .
so here the cardinality of employee is 1 while there can be a case that all the employee work in one department so cardinality of set department will be n .
now the participation is the minimum number of attribute from set a to b .
as there is no condition on the minimum number of employee we can say that the participation of the employee will be 0 . and also we have not specified that whether there should be ne employee in every department .
so the participation of the department will also be zero .
now suppose the case if i specify that every employe has to work in one and only one department . so there will be o one which will not be connected to set B .
so the participation then will become . 1 for the employee . while still no condition on department . department participation is still 0;

so if a set has a participation of 1 then we call it total participation and it is represented as double line .

so in er diagram actually we represent the cardinality ,while participation is also represented , in some way but not all the participation are represented .

all participation are not represented but only the total participation whose definition is told earlier is represented by double line.
and the cardinality are represented on the edge .
NOTE ::::
cardinality in er diagram are represented on opposite side like if it is like this

employee --------n ------ works for ---------1-------- department
this means that the cardinality of employee is 1 while the cardinality of the department is n means , an employee can only work in one department while in a department n can work .

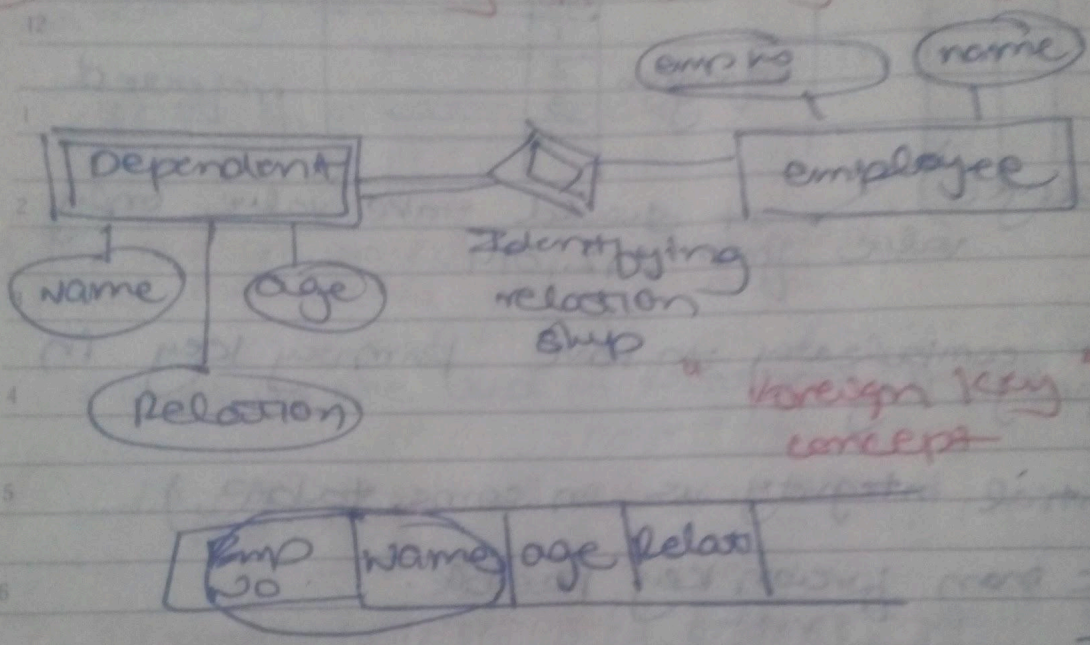
of so now we are good to go with the conversion of relational modal .
lets see some things about weak entity and

so the weak entity is a one which does not have a primary key .
but to convert a er diagram in a relational model every table should have a primary key . so we just take the primary key of an entity on which it is dependent and put that primary key with any attribute of the weak entity so that it can then uniquely identify the attributes like . this
here is an example .

20 MONDAY — Conversion of E-R Diagram to Relational Schema

(1) Create a table for all entity.
(2) Composite attribute are used directly. No multivalued attribute will be used in table

(3) weak entity to strong entity



Foreign key concept

Emp no	name	age	Relat

(Name must be the partial key.)

But given a part of table you can identify it in colm emp no.

Notes

(3) → (Relationship to Relation)

s here dependent is a weak entity and thats why we just added the strong entity primary key with it . and there is always a total participation from the weak entity side.

so we have just made a weak entity a strong entity .
now the rules of conversion to an er diagram are .

- first convert every weak entity into strong entity.
- make a new table for every entity .
- make a new table of a relationship if the degree of relationship is greater than 2 .

Degree of relationship means " number of entity participating with that relation. "
like the above example the degree of the relationship . works for was 2 as employee and department (2 entities were participating)

- No need for a new table if the degree is 1.
- now we have somewhat difficult cases for degree 2 .
- if a relationship has a degree 2 then these cases arise .
- the cardinality can be 1 : 1
- or 1 : N
- or N : 1
- or N:N
- in the first case 1.:1 we just transfer the primary key of one table into second table , if relationship contain any attributes transfer them too . and do this from where there is a total participation. if not total participation han can transfer to any side.
- N:1 take the primary key of the entity who has N cardinality and take relationship attributes and transfer them to the one side.
- (remember cardinality are on opposite sides.)
- easy way just take the primary key of where there is a 1 in er diagram and send it to n side.
- like in the case of 1:N take the primary key of lhs and send to rhs ,
- in case of N:1 take the primary key of rhs nd sed to lhs .
- in the case of M:M we have to make a new table and take primary key of both the table and add put them there with relationship attributes.

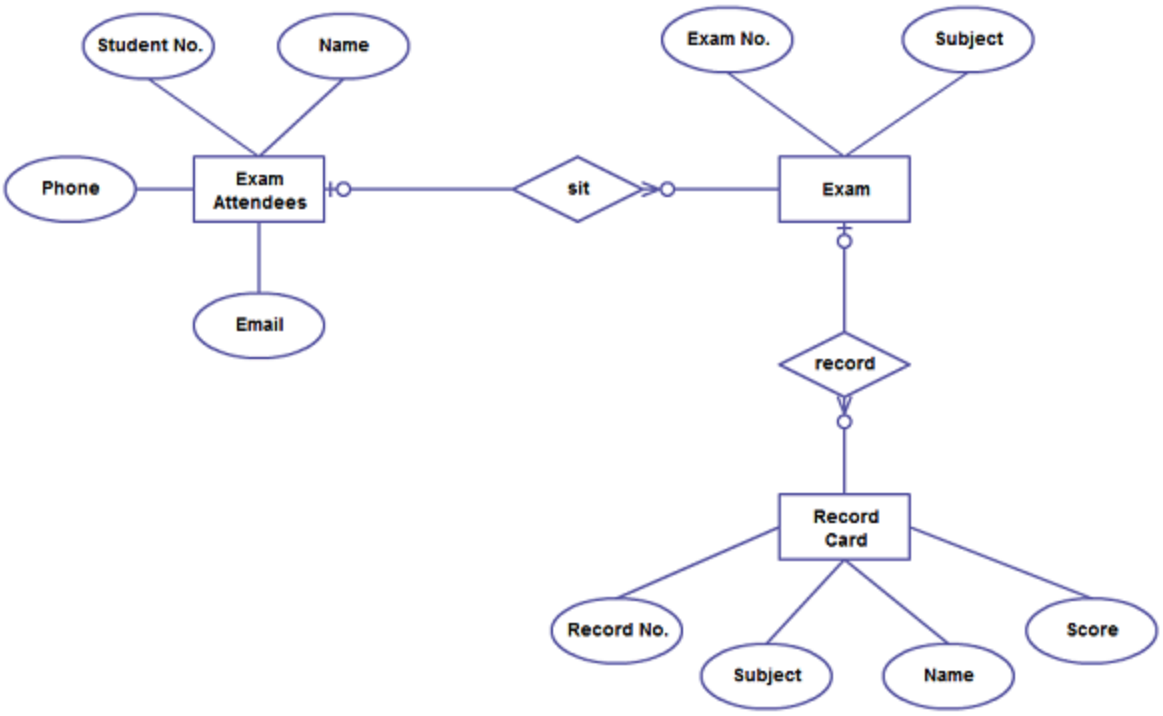
And we are done . making the relational model .
any question . ???

Q. sir how degree of a relationship could be 1?
degree of a relationship can be one if there is only one entity .
it's seems illogical ok , we can have one . but an relationship can't be possible without any entity s it can't be zero.

Following are the degrees of Relationships.
1. Single Entity - Unary
2. Double Entities - Binary0
3. Triple Entities - Ternary
4. N Entities - N- ary

https://msdn.microsoft.com/en-us/library/bb399189.aspx

Q. sir there is also something called as a partial key in weak entities..
Ans. yes. the partial key is the attribute with which we add the primary key of the strong entity, actually the partial key is able to uniquely identify every record with the help of strong entity primary key . so any attribute which is able to do so is called partial key or discriminator attribute .



Anyone wanna ask something related to ER diagram could ask..

Q.1 - How to convert ER dia to relational model and find the minimum no. of tables to satisfy a particular normal form?
yes

Q.2 - yeah, converting from er to relational, minimum tables required?? That depends on various things. Just google it.www.cs.sjsu.edu/faculty/lee/cs157/26Presentation_Jung_T_Chang.ppt

above link not opening.
It works. But is it the only way? There can be multiple ways to form relations from ER model rt? So, suppose the question is to minimize the no. of relations can we assume 1NF as that will give min no. of relations? (Also is there any GATE question which asks this without specifying a normal form?) Yes they ask it without mentioning the normal form..

Q.3 After converting from ER to relational, which normal form do we get?
1nf..sure?
ya, 1nf will be there always, because we convert to relational keeping in mind what is required in 1nf. ok

Q.4 calculating total no. of schedules, like recoverable schedules, cascadeless schedules out of a given schedule is sometimes where i got struck.. any easy way to approach these questions?

calculating can be done using the same theory .

procedure to find recoverability and cascade lessness
1 look for the dirty read . if no dirty read then schedule is recoverable and as well as cascadeless.
2-
if there is still a dirty read , schedule is not cascadeless and go to step no 3
3
if the transition are committing in the same order in which dirty read is done then it is recoverable otherwise not .
recoverability is a mandatory property but cascadeless is not.

What is fan Trap ??

fan trap is when we have two many to one relationship coming out of an entity in different direction .
like
this
entity ------ m--- <> ------1 ------ entity -------1 -----<> -----m-----entity
if we have such a situation we can't uniquely get a value for the first entity in the set of last entity. mapping can't be done so we have to resolve it .
we can do it by turning any 1-m to m to 1 . just turn one .

Question: How many minimum relational database table required, which satisfy 4NF?
4nf says that it should be in bcnf and no multivalued dependency . so i can make only one table with the property. but given a relation i have to see,

what is multivalued dependency?
in a functional dependency like A->B we check that for every unique A there should be same B .
but if we get for same A more than one value then it is multivalued . like for a u are getting 3 , 4 then multivalued.
remember one point
A->B
can be written as
A->>B a multivalued b but the converse is wrong
like if it is given A →>b we cannot say that A->b holds,
as it is saying we have multiple values for a then we can't say we will have unique value a.

Is explained below:
Consider the following relation..

Name	Phone	Game Liked

Suppose Len plays cricket and football and has a phone numbers 9847123641 and 9621748362. So, we will get

Name	Phone	Game Liked
Len	9847123641	Cricket
Len	9621748362	Football
Len	9847123641	Football
Len	9621748362	Cricket

Is there a FD above? Nopes.

Is there a MVD- yes. Name - -> Phone and Name - -> Game.

This is because the **Game a person likes is independent of his Phone number**. So, for each game one likes, we must **repeat** his phone numbers (also for each phone number one adds we must repeat all games he likes) in the table, thus causing redundancy which is bad. Such a scenario is called Multi-valued dependency. (A FD is a special case of MVD- why?)

SIR NOT GETTING…. :)

The issue here is phone number and Game Liked are independent.. So, say I got a new phone number.. So, in the table I must now add new tuples - not once- but for each game I like. And this redundancy is not captured by FD - the above relation is in BCNF.

Like if Len has got a new number 9999. We have to make two entries (i.e., 2 tuples)..one for fb, and other for cricket.

yes, correct. And this is the simplest case. Consider the case when he likes 100 games. So, this dependency is bad.. got it.. :)

4NF- there should not be any non-trivial MVD. Non-trivial means, for X - -> Y,

- Y not a subset of X, and
- X union Y should not be the complete attribute set.

How to decompose in case of MVD- For X - ->Y, make

- XY a relation and
- all attributes except (Y-X) the other relation.

Got this?

Unlike for FD we cannot always split RHS of a MVD. Please see below link..

[Must Read (just takes a few minutes if you read above)]

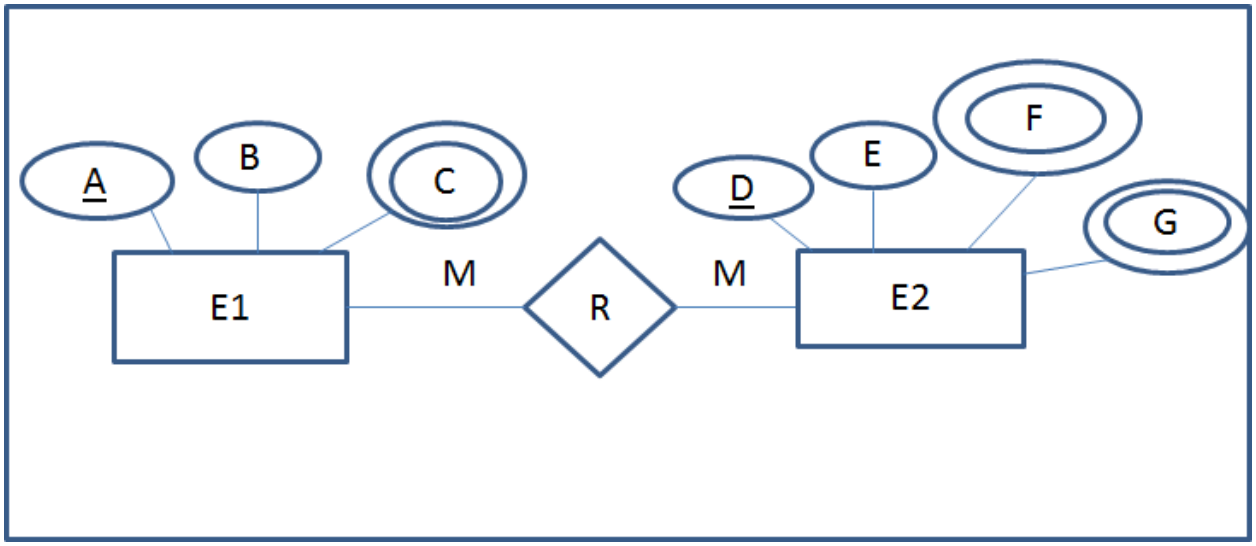
Above relation in 4NF after decomposition:

Name	Phone
Len	9847123641
Len	9621748362

Name	Game Liked
Len	Cricket
Len	Football

This design is better rt?
yes :)

IS 4NF in GATE??yes 4NF had been asked before. so, 1 nf,2,3,bcnf, and 4nf.. thats it?? Most probably.. But I suggest to just read 5NF (at least definition).



Q 1: Please Explain 4NF . and steps to identify whether a given relation in 4NF or Not

See couple of pages before..

Q 2: What is dependency preservation (DP) ?
What are the steps to find whether group of FD follows DP

When we decompose relations we must ensure losslessness (STRICT). (Checking losslessness is easy- just see if the intersection of decomposed relations is a key of one of the decomposed relation). But it is also desirable to ensure FDs are preserved. Till 3NF, there are procedures to decompose relations which always satisfy losslessness and dependency preserving.. But for BCNF we might not always get a dependency preserving decomposition.. This means we have to do join of some relations to check if a dependency is satisfied. So, in order to see if a decomposition is dependency preserving we can try 2 things

- **If all FDs are straight away in the decomposed relations it is definitely dependency preserving**
- **If all the FDs can be derived from the FDs in decomposed relations (like transitive), then also it is dependency preserving (Actually there is a more formal approach in text)**
- **If we can determine FD only with a join then the decomposition is not dependency preserving**

Sir, Could you pls explain me how to check DP with suitable example ?

If you can give a question we can see. Also, I don't follow standard procedure here, so I guess its not good for me to explain :)

http://gateoverflow.in/11745/find-whether-given-relation-is-dependency-preserving
Is the following decomposition dependency preserving?

R={ABCDEG}

AB->C (how to check?) this can be preserved..

SINCE, A->C => AB->CB (augmented B) => AB->C and AB->B (okay, if we have A->C in the set of FDs, otherwise?)

But we have derived A->C from given set of FDs.
No rt? Can we? no. We have (AB -> C, AC -> B, BC -> A, but we cannot derive anything more here) .Then? its not dependency preserving :) ok sir.. but after adding A->C is there any dependency not being preserved?

is BC->A is preserved - ? - no.. this dependency will be still not preserved.. But AB -> C and AC ->B will be preserved.

AC->B (how to check?) DO SAME AS ABOVE

AD ->E satisfied in (ABDE)

B->D satisfied in (ABDE)

BC->A (how to check?)

E->G (satisfied in EG)

for the following decomposed relations {AB , BC ,ABDE, EG}

So, 3 more FDs need to be satisfied. Is there any way to check them without a join? So, not dependency preserving? (we just need 1 FD not being preserved to say that a decomposition is not dependency preserving)

Also consider one more FD

A -> C (not in question but added for explanation) from relation AB, A->B and from relation BC, B->C. Therefore, A->C. yes. Adding A->C also helps in preserving other FDs? Yes, because A->C can help in deriving some FDs like AB->C and AC->B in above example.
It is better to follow standard algorithm (to avoid risk) but 90% of GATE questions can be answered if you know the concept correctly without using the full algorithm..

Is this new dependency being preserved in above decomposition?

Q3 : IS R-W make conflict ? I studied that R-W-R makes a conflict which is called unrepeatable Read . But (http://gateoverflow.in/1796/gate2014-1_29) Someone explained here there is R-W as conflict . Please answer it over there .

Q4 : How to check View Serializability ??

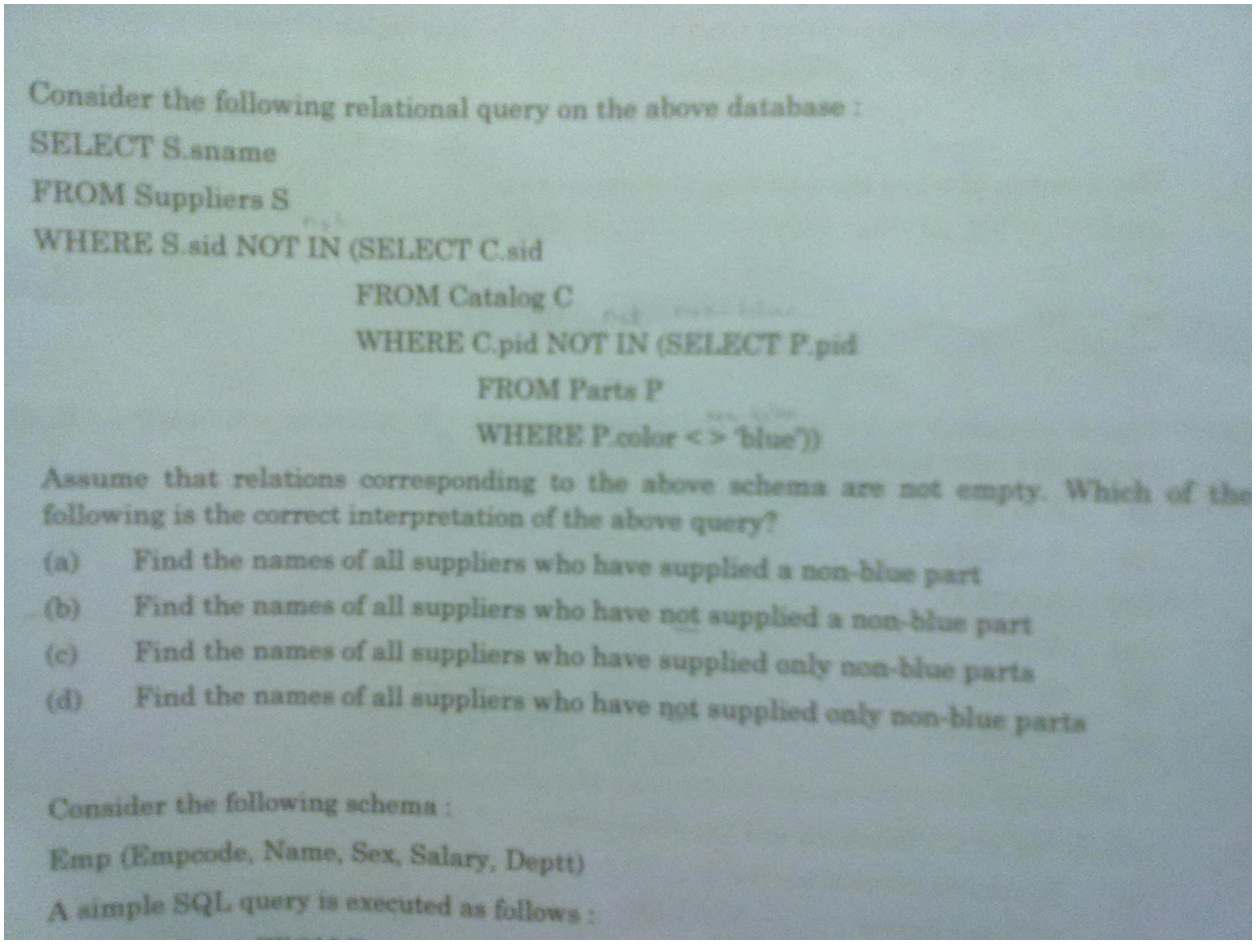
Q5: Two transactions T1 and T2 are given as follows:
T1: R1(A) W1(A) R1(B) W1(B)

T2: R2(B) W2(B) R2(C) W2(C)

The total number of conflicts serializable schedules that can be formed by T1 and T2 are____

Q6 : What is Phantom Phenomenon ? Please explain the concepts behind it .

Q7: SQL : Could you pls explain me how the following querry works? (with suitable table and diagram for more understanding) ISRO-2015



Consider the following relational query on the above database :

```
SELECT S.sname
FROM Suppliers S
WHERE S.sid NOT IN (SELECT C.sid
                    FROM Catalog C
                    WHERE C.pid NOT IN (SELECT P.pid
                                        FROM Parts P
                                        WHERE P.color <> 'blue'))
```

Assume that relations corresponding to the above schema are not empty. Which of the following is the correct interpretation of the above query?

(a) Find the names of all suppliers who have supplied a non-blue part

(b) Find the names of all suppliers who have not supplied a non-blue part

(c) Find the names of all suppliers who have supplied only non-blue parts

(d) Find the names of all suppliers who have not supplied only non-blue parts

Consider the following schema :

Emp (Empcode, Name, Sex, Salary, Deptt)

A simple SQL query is executed as follows :

This is a previous GATE question. You can see here..
http://gateoverflow.in/1339/gate2009_55-56

Q8 : Identify the Best Normal Form R satisfies ?

which R? The question is incomplete. We just have a set of FDs here and no relation(s).

a) A ---> B , BC ---> D , A ----> C
b) AB ----> C ,AB ----> D , C -----> A ,D----->B

Here first we will find Closure A+ = { A, B,C wait, but we dont have a relation rt? We check for normal form when we have a relation(s).

Given R(ABCD) and

AB → C ;

ABD → C ;

ABC → D ;

AC → D

which the highest possible normal form for the above relation ?

First we find the keys.

AB must be there in any key- because they are not in RHS of any FD. Now, AB -> CD, so AB is the key.

AB is the key.
so non prime attributes are - C, D
yes..

yes.. Now, AC -> D. Here, AC is not a super key. So, not BCNF.
D-AC = D, is not a prime attribute. So, neither 3NF.
No partial FD, so it must be 2NF..

Sir Also tell about 5NF ?? I just remember it is related to Join dependency, have to revise.. But just read the definition.. ok sir .

Q8: Concurrency

In a schedule S, three transactions T_1, T_2 and T_3 are executing concurrently. The schedule is given below. The given schedule is conflict equivalent to which of the given serial schedules.

T_1	T_2	T_3
Read Y		
Y = Y + 100	Read Z	
WriteY	Z = Z + 100	Read X
	Write Z	X = X – 100
		Write X
Read X		
X = X + 100	Read Y	
Write X	Y = Y – 100	Read Z
	Write Y	Z = Z – 100
		Write Z
Read Z		
Z = Z + 200	Read X	
Write Z	X = Z + 100	Read Y
	Write X	

(a) $T_1 \rightarrow T_2 \rightarrow T_3$ (b) $T_3 \rightarrow T_2 \rightarrow T_1$

(c) $T_1 \rightarrow T_3 \rightarrow T_2$ (d) conflict serializable