Assignment 1: Anomalies in Relational Model
Consider the following table

<u>Eid</u>	**Ename**	**<u>Dept</u>**	**Salary**	**Address**
100	**John**	**Sales**	**10000**	**Ulm**
101	**Smith**	**Production**	**20000**	**Bonn**
100	**John**	**Production**	**15000**	**Ulm**
101	**Smith**	**Automation**	**20000**	**Bonn**

(a) If you want to insert the following rows into the above table, do these insertions cause any anomalies ? Justify.
(i) (102, James, Null, 15000, Berlin)
Ans. This will cause an insertion anomaly as we can't insert a person without a Dept since it is a prime attribute.
(ii) (100, Nick, Manufacturing, 10000, Turin)
Ans .This will cause an insertion anomaly since we already have an employee with Eid 100 , as two employees can't have the same employee id ,this is an insertion anomaly.

(b) If you change the 3rd row in the above table to (100, John, Production, 20000, Berlin), does it cause any anomalies ? If so, why ?
Ans . This will cause the update anomaly since the information for john is present in two rows but address is updated to Berlin in only this row .

(c) Suppose you want to delete the 4th row from the above table. Does it cause any anomalies ? Justify your
Ans.
This will cause a deletion anomaly ,since there is no information for Dept automation after deletion of this row .

Assignment 2: Functional Dependencies
(a) Find out all the functional dependencies in the previous table.
ED -> S
E -> NA

(b) Consider Relation 'R' containing columns A, B, C, D, E and functional dependencies FD = {A→B,B→D, C→DE, CD→AB}.
Find out all the candidate keys.

C is the candidate key.
Proof:
1)C -> DE (given)
2)C -> D (decomposing DE in 1)
3)C -> E (decomposing DE in 1)
4)C -> C (trivial)
5)C -> CD (union on 4,2)
6)CD -> AB (given)
7)C -> AB (transitivity on 5,6)
8)C -> A (decomposing AB 6)
9)C -> B (decomposing AB 6)

From 2,3,4,8,9 we get

C^+ ={A,B,C,D,E}
Hence proved .

(c) Find the minimal cover of {A→C, AC→D, E→AD, E→H}.

Let's go on each dependency from left to right and check if its derivable or not .
First we decompose all righ site in dependencies .
New F = **{A→C, AC→D, E→A, E→D, E→H}**

REMOVING EXTRANEOUS ATTRIBUTE FROM RIGHT SIDE

1)
A -> C

Not derivable ,A^+={A}

2)
AC -> D

Not derivable ,

A^+={A,C}
AC^+ = {A,C}

3)
E -> A

Not derivable

E^+ ={D,H,E}

4)
E -> D

Derivable,hence redundant

i)E -> A {given}
ii)A-> C (given)
iii)AC -> D (given)
iv)A->AC (union of A-> C with A->A)
v)A->D (transitivity on iv,iii)
vi)E->D (transitivity on v,i)

5)
E -> H

Not derivable.

REMOVING EXTRANEOUS ATTRIBUTE FROM LEFT SIDE

A^+ ={A,C}
SO A-> AC and AC -> D , since C can be derived from A itself so we can write A -> D

So minimal cover is {**A→C, A→D, E→A, E→H}**

(d) Check whether the two functional dependencies 'F' and 'G' are equivalent or not. F={A→B, B→C,C→D}

Since question is incomplete i left it .

Assignment 3: Normal Forms
(a) Relation R (ABCDEF) and FD : {EC → AD, A → B, C → F}. Does it satisfy 2NF and why ? If not, decompose the table in such a way that they satisfy at least 2NF. Find out the highest normal satisfied by the decomposed table.

The candidate key is EC , and prime attributes are E,C

This is not 2NF because of **C → F** as C is prime attribute and f is not a prime attribute this causes partial dependency, which leads to violation of 2NF.

We can break this into relation R1(A,D,E,C,B) ,R2(C,F).

But This does not satisfies 3NF , because of A->B , as B is non prime attribute . (violation of 3NF in R1)

(b) Relation R(A B C D) and FD : {A→ BC, A → D, B → D}. What is the highest normal form it is satisfying ? Does it satisfy 3NF ? If not, split the table and rewrite the functional dependencies of each table so that it satisfies 3NF.
The candidate key is A , it does not satisfy 3NF because of B->D (D is not a prime attribute.)
But it is 2NF because no partial dependencies are there.
So we split as

R1(A,B,C) , R2(B,D) now it follows 3NF.

(c) Relation R (P, Q, R, S ,T) and FD : { P → QR , RS → T , Q → S , T → P }. Is the decomposition of R into R1 (P, Q, R) and R2 (P, S, T) lossless and dependency preserving ? Justify your answer.

Since P is a candidate key , and it is common in both the relations , we get decomposition is lossless.
But decomposition is not dependency preserving.
Because the dependency RS -> T is not preserved.
Proof:

For R1 we have ->
P->QR
QR -> P

For R2 we have ->
P->T
P->S
T->P
T->S

So from this calculate RS^+
We get RS^+ ={RS}

So we have loss of dependency here .

(d) Justify why a relation with only two attributes is always in BCNF ?
R(A,B)

There can only be two cases one candidate key,both are candidate key.
Case 1 A is the candidate key (A->B)   // this is simillar to the case where only B is the candidate key.

So this is satisfying BCNF since A is candidate key/

Case 2 A,B both are candidate key (A->B,B->A)

Since both are candidate key's it still satisfies BCNF.

Hence in all cases only two attributes is always in BCNF.