Question 1

a) R(A, B, C, D, E) F={A → BC, CD → E, B → D, E → A}

Break down right side:
A → BC gives A → B and A → C via 5th axiom (decomposition)

A → B and B → D entails A → D via 3rd axion (transitivity)

A → C and A → D entails A → CD via 4th axiom (union)

A → CD and CD → D entails A → E via 3rd axiom (transitivity)

A → BC and A → D and A → E entails A → BCDE via 4th axiom (union)

E → A entails E → BCDE via 3rd axiom (transitivity)

E → BCDE entails E → BCD via redundancy

E → A and E → BCD entails E → ABCD via 4th axiom (union)

CD → E entails CD → ABCD via 3rd axiom (transitivity)

CD → ABCD entails CD → AB via redundancy

CD → AB and CD → E entails CD → ABE via 4th axiom (union)

Gives,
$A^+ = \{A, B, C, D, E\}$
$B^+ = \{ B, D, \}$
$E^+ = \{A, B, C, D, E\}$
$CD^+ = \{A, B, C, D, E\}$

Therefore, candidate keys are A, E, & CD

b) Is AB → C covered by F?
We know A → C via 5th axiom
We know A → B via 5th axiom

AB → CB via 2nd axiom (augmentation)

AB → C and AB → B via 5th axiom (decomposition)

Therefore AB → C is covered by F

Question 2

$T(A, B, C, D)$
$F = \{ABC \to D, CD \to A, CA \to B, AD \to C, CD \to B\}$

Step 1: split rhs into singletons. Already singletons.

Step 2:
$\underline{ABC \to D}$
$J = H - \{ABC \to D\}$

$CD \to A : D \notin ABC^+$
$CA \to B : CA \in ABC^+$ but B already in ABC^+
$AD \to C : D \notin ABC^+$
$CD \to B : D \notin ABC^+$

Since, $D \notin ABC^+ = \{A, B, C\}$, therefore $ABC \to D$ is necessary.

$\underline{CD \to A}$
$J = H - \{CD \to A\}$

$ABC \to D : AB \notin CD^+$
$CA \to B : A \notin CD^+$
$AD \to C : A \notin CD^+$
$CD \to B : CD \in CD^+$ so, add B : $CD^+ = \{C, D, B\}$

Since, $A \notin CD^+ = \{C, D, B\}$, therefore $CD \to A$ is necessary.

$\underline{CA \to B}$
$J = H - \{CA \to B\}$

$ABC \to D : B \notin CA^+$
$CD \to A : D \notin CA^+$
$AD \to C : D \notin CA^+$
$CD \to B : D \notin CA^+$

Since, $B \notin CA^+ = \{C, A\}$, therefore $CA \to B$ is necessary.

$\underline{AD \to C}$
$J = H - \{AD \to C\}$

$ABC \to D : BC \notin AD^+$
$CD \to A : C \notin AD^+$
$CA \to B : C \notin AD^+$
$CD \to B : C \notin AD^+$

Since, $B \notin CA^+ = \{C, A\}$, therefore $CA \to B$ is necessary.

$\underline{CD \to B}$
$J = H - \{CD \to B\}$

$ABC \to D : AB \notin CD^+$
$CD \to A : CD \in CD^+$ so, add A : $CD^+ = \{A, C, D\}$
$CA \to B : CA \in CD^+$ so, add B : $CD^+ = \{A, B, C, D\}$
$AD \to C : AD \in CD^+$ but, CD^+ already contains C

Since, $B \in CD^+ = \{A, B, C, D\}$, therefore $CD \to B$ is redundant.

Step 3: try to remove attr from LHS $H = \{ABC \to D, CD \to A, CA \to B, AD \to C, \cancel{CD \to B}\}$

$\underline{ABC \to D}$
$D \notin AB^+$
$D \notin BC^+$
$D \in AC^+$

$CA \to B$ via augmentation $CA \to ABC$
Therefore, B is unnecessary : $AC \to D$

$\underline{CD \to A}$
$A \notin C^+$
$A \notin D^+$
Therefore, $CD \to A$ is not extraneous.

$\underline{CA \to B}$
$B \notin C^+$
$B \notin A^+$
Therefore, $CA \to B$ is not extraneous.

$\underline{AD \to C}$
$C \notin A^+$
$C \notin D^+$
Therefore, $AD \to C$ is not extraneous.

Therefore, the minimal basis is $M = \{AC \to D, CD \to A, CA \to B, AD \to C\}$

Question 3

a)

$CD \to E, CD \to F$ via decomposition
$AB \to C$ via augmentation $ABD \to CD$
Since $A \to D$ therefore $ABD = AB$ b/c redundant
Therefore $AB \to F$

b)

$BEF \to C$ via transitivity $BEF \to D$
$BE \to A$ via reflexivity $BEF \to A$
Therefore $BEF^+ = \{A, B, C, D, E, F\}$

Question 4

1

Two differnt companies cannot have the same company ID
$companyID \to companyName, cityName, country, assets$ (companyID is a key for Company)

Two different departments cannot have the same deptID
$deptID \to deptName, companyID, cityName, country, depMgrID$ (deptID is a key for Department)

Two different cities cannot have the same cityID
$cityID \rightarrow cityName, country$ (cityID is a key for City)

Two different cities in the same country cannot have the same name
$(country, cityName) \rightarrow cityID$ (where (country,cityName) is a key for City)

The company name and the city its located in determine the company ID
$(companyID, deptName) \rightarrow deptID$ (where (companyID, deptName) is a key for Department)

One manager cannot run 2 different departments $depMgrID \rightarrow deptID$

2

The schema defined is a good one since 3NF is satisfied. For all FDs, either the left side is a superkey, or the right side is prime.

$companyID \rightarrow companyName, cityName, country, assets$ (LHS is superkey for Company)

$deptID \rightarrow deptName, companyiD, cityName, country, depMgrID$ (LHS is superkey for Department)

$cityID \rightarrow cityName, country$ (LHS is superkey for City)

$(country, cityname) \rightarrow cityID$ (LHS is superkey for City)

$(companyName, cityID) \rightarrow companyID$ (LHS is superkey for Company)

$(companyID, deptName) \rightarrow deptID$ (LHS is superkey for Department)

$depMgrID \rightarrow deptID$ (LHS is NOT superkey for Department BUT RHS is prime)

Question 5

a)

<u>ACA</u>
T_2 depends on T_3 (because T_3 writes to Y). T_2 reads Y after commit therefore ACA.

<u>Recoverable</u>
T_1 depends on none, commits first. T_2 depends on T_3, commits after T_3. T_3 depends on none, commits second.

<u>Strict</u>
Yes, $W(Y)$ and $W(X)$ are committed before Y is read and X is written to.

b)

<u>ACA</u>
No T_2 reads from T_3 before T_3 commits.

<u>Recoverable</u>
No, T_2 depends on T_3 but commits before T_3.

Strict
No, $R_2(Y)$ happens before C_2

c)

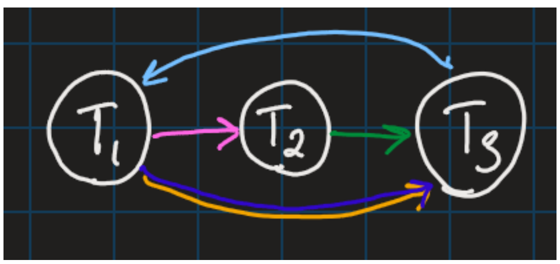
ACA
Yes, there are no reads of any resource after X, Y and Z are written to.

Recoverable
T_1 depends on none, T_2 depends on none, T_3 depends on none. Therefore, all commits can occur in any order, and the schedule is recoverable.

Strict
No, $W_2(Y)$ happens before C_3

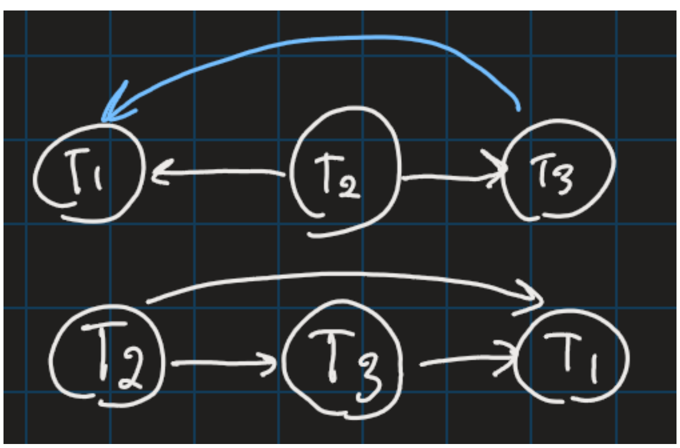
Question 7

a)



There is a cycle (T_1 depends on T_2 depends on T_1) therefore the schedule is not serializable.

b)



There is no cycle in the schedule, therefore it is serializable. The serialized equivalent is T_2, T_3, T_1.

Question 8

No lock on reads means we can read uncommitted writes, which means possible dirty reads and possible unrepeatable reads.

Serializability: no, can read before commit.
Conflict-serializability: no, dirty reads means possible cyclic dependency, therefore conflict serializability is not guaranteed.
Recoverability: same as above, reads before commit allows xactas to commit before the xacts it depends on commit.
ACA: dirty read from transaction that later aborts is possible.
Deadlock: no, lock on writes means its possible for transactions to wait on eachother cyclically.