Question1:

(a) F = {RDT -> P; RDT -> C; C -> A; PDT -> R; PDT -> C; CDT -> P; CDT -> R}
P and C are functionally dependent on R, D and T. Which means if a tuple of R, D, T show up more than once in the table, they will have the same P and C value.
Each value of C is associated with only one value of A.
R and C are functionally dependent on P, D and T, if a certain combination of R, D, T show up more than once in the table, they will have the same R and C value.
P and R are functionally dependent on C, D and T, means that C, D , T are determinant sets and P and R are dependent sets.

(b) RDT -> P;
RDT -> C, C -> A
RDT -> A (Transitivity);
RDT -> AP (Union).

(c) F = {RDT -> P; RDT -> C; C -> A; PDT -> R; PDT -> C; CDT -> P; CDT -> R}
RDT -> C and C -> A; RDT -> A (Transitivity);
RDT -> C, RDT -> A and RDT -> P; RDT -> PCA (Union); **RDT -> RDTPCA** (Augmentation
PDT -> C and C -> A; PDT -> A (Transitivity);
PDT -> R, PDT -> A and PDT -> C; PDT -> RC (Union); **PDT -> PDTRCA** (Augmentation)
CDT -> C (trivial) and C -> A; CDT -> A (Transitivity);
CDT -> P, CDT -> A and CDT -> R**;** CDT -> PR (Union); **CDT -> PDTRCA** (Augmentation)
RDT, PDT, CDT are the keys, because they cover all the attributes of the table Schedule.

(d) Schedule (C, D, T, R, P, A)
F = {RDT -> P; RDT -> C; C -> A; PDT -> R; PDT -> C; CDT -> P; CDT -> R}
Key = RDT, C, PDT, CDT
Closure: $\{\mathrm{RDT}\}^{+}$= {R, D, T, C, A, P}; Closure: $\{\mathrm{C}\}^{+}$= {C, A}; Closure: $\{\mathrm{PDT}\}^{+}$= {R, D, T, C, A, P};
Closure: $\{\mathrm{CDT}\}^{+}$= {R, D, T, C, A, P}
Schedule is not in BCNF because C is not a super key.
Schedule1(C, D, T, P, R)
F1 = {RDT -> P; RDT -> C; PDT -> R; PDT -> C; CDT -> P; CDT -> R}
Schedule2(C, A)
F2 = {C -> A}

(e) ProfsSchedule(D, T, P)
D -> D, T -> T, P -> P are trivial.
DT -> T, DT -> D, DP -> P, DP -> D, TP -> T, TP -> P are also trivial.
DTP -> DTP is also trivial.
DTP -> CAR is not possible as DTP -> C, DTP -> A, DTP -> R are not part of the table ProfsSchedule(D, T, P).
So, there are no new functional dependencies that hold over ProfsSchedule.

(f) F = {RDT -> P; RDT -> C; C -> A; PDT -> R; PDT -> C; CDT -> P; CDT -> R}
1. RDT -> P can be removed, RDT^{+} = RDTCAP
2. RDT -> C cannot be removed, RDT^{+} = RDT

3. C -> A cannot be removed, C^+ = CA
4. PDT -> R can be removed, PDT^+ = PDTRCA
5. PDT -> C cannot be removed, PDT^+ = PDT
6. CDT -> P cannot be removed, CDT^+ = CDTAR
7. CDT -> R cannot be removed, CDT^+ = CDTA

Outcome => H = {RDT -> C, C -> A, PDT -> C, CDT -> P, CDT -> R}

Minimal Cover: M = {RDT -> C, C -> A, PDT -> C, CDT -> PR}

Question2:

(a) F = {A -> BC, A -> DE}; F1 = {A -> BC}; F2 = {A -> DE}
This is a lossless join decomposition as F = {A -> BC, A -> DE}, which covers all the attributes of R (A, B, C, D, E) and F1 ∩ F2 = {A}, which is a key for R, R1 and R2.
Dependency preserving because $(\mathrm{F1\ U\ F2})^+ = F^+$ = ABCDE.

(b) F = {A -> BC, A -> DE, B -> D}; F1 = {A -> BC}; F2 = {A -> DE}
This is a lossless join decomposition as F = {A -> BC, A -> DE, B -> D}, which covers all the attributes of R (A, B, C, D, E) and F1 ∩ F2 = {A}, which is a key for R, R1 and R2.
Not Dependency preserving because cannot check B -> D without computing R1 $\bowtie$ R2.

(c) F = {A -> B, A -> D}; F1 = {A -> B}; F2 = {A -> D}
This is not lossless join decomposition as F = {A -> B, A -> D}, which does not cover all the attributes of R (A, B, C, D, E).
Dependency preserving because $(\mathrm{F1\ U\ F2})^+ = F^+$ = ABD.

(d) Every Binary relation is in BCNF.
A relation is in BCNF if whenever A -> B is not a trivial functional dependency in the relation, A is a super key (covers all the attributes of the relation). Binary relation means a relation with 2 attributes.
For example, R (A, B) is a binary relation. FDs: {A -> B} or FDs: {B -> A}. In both cases the functional dependencies are non-trivial and the left-hand side (A/B) is a super key, as they cover all attributes of R. Which means R is in BCNF.

Question3:

(a) X -> Y and W -> Z then XW -> Y Z.
- X -> Y (given) and XW -> YW (Augmentation)
- W -> Z (given) and XW -> ZX (Augmentation)
- XW -> WXYZ (Union)
- XW -> YZ (Trivial) [Proved].

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

FDs: {C -> D, BE -> A, BEF -> C}

- BEF -> C and C -> D, BEF -> D (Transitivity)
- BE -> A, BEF -> AF (Augmentation)
- BEF -> CDAF (Union)
- BEF -> ABCDEF (Trivial)

As BEF covers all the attributes of the table R, BEF is a key. [Proved]

Question4:

(a) example of a conflict-serializable schedule that is not a serial schedule:

T1: R1(A), W1(A), R1(B), W1(B)

T2: R2(A), W2(A), R2(B), W2(B)

R1(A), W2(A); R2(A), W1(A); R2(B), W1(B); R1(B), W2(B) conflict.

Serial schedule:

T1: R(A), W(A), R(B), W(B)

T2: R(A), W(A), R(B), W(B)

Serializable schedule:

T1: R(A), W(A), R(B), W(B)

T2: R(A), W(A), R(B), W(B)

(b) Consider the following locking protocol: Before a transaction T writes a data object A, T has to obtain an exclusive lock on A. For a transaction T, we hold these exclusive locks until the end of the transaction. If a transaction T reads a data object A, no lock on A is obtained. State which of the following properties are ensured by this locking protocol: serializability, conflict-serializability, recoverability, avoids cascading aborts, avoids deadlock. Justify your answer for each property.

- Needs exclusive lock to write and holds on to it until end
- No lock needed for reading

T: X(A), W(A) hold the lock until end, no lock needed for R(A).

Question5:

USERNAME	NAME	BALANCE
luke	Skywalker	8000.00
kylo	Ren	12500.00
rey	Rey	5000.00
finn	Finn	1500.00
leia	Skywalker	2000.00

1.

```
USERNAME   NAME                 BALANCE
---------- -------------------- ------------
luke       Skywalker                 8000.00
kylo       Ren                      12500.00
rey        Rey                       5000.00
finn       Finn                      1500.00
leia       Skywalker                 2000.00
snoke      First Order              20000.00

  6 record(s) selected.
```

2.

```
USERNAME   NAME                 BALANCE
---------- -------------------- ------------
luke       Skywalker                 8000.00
kylo       Ren                      12500.00
rey        Rey                       5000.00
finn       Finn                      1500.00
leia       Skywalker                 2000.00

  5 record(s) selected.
```

Session B had the original 5 inputs and session A had 6 including the last update.

This is because the two sessions are not in sync with each other.

We can commit after finishing all the actions in session A to solve this problem.

4.

```
db2 => select * from accounts

USERNAME   NAME                 BALANCE
---------- -------------------- ------------
luke       Skywalker                 8000.00
kylo       Ren                      12500.00
rey        Rey                       5000.00
finn       Finn                      1500.00
leia       Skywalker                 2000.00

  5 record(s) selected.

db2 => select * from accounts

USERNAME   NAME                 BALANCE
---------- -------------------- ------------
luke       Skywalker                 8000.00
kylo       Ren                      12500.00
rey        Rey                       5000.00
finn       Finn                      1500.00
leia       Skywalker                 2000.00
snoke      First Order              20000.00
```

6.

```
USERNAME   NAME                 BALANCE
---------- -------------------- ------------
luke       Skywalker                 8000.00
kylo       Ren                       7500.00
rey        Rey                      10000.00
finn       Finn                      1500.00
leia       Skywalker                 2000.00
snoke      First Order              20000.00

  6 record(s) selected.
```

7.

```
USERNAME   NAME                 BALANCE
---------- -------------------- ------------
luke       Skywalker                 8000.00
kylo       Ren                       7500.00
rey        Rey                      10000.00
finn       Finn                      1500.00
leia       Skywalker                 2000.00
snoke      First Order              20000.00

  6 record(s) selected.
```

The command did not go through and the session stopped running. This is because we did not commit the actions in session A. while we can still see the change in session B without committing in session A, we cannot run any actions in session B.

8.

```
USERNAME   NAME                 BALANCE
---------- -------------------- ------------
luke       Skywalker                 8000.00
kylo       Ren                       7500.00
rey        Rey                      10000.00
finn       Finn                     11500.00
leia       Skywalker                 2000.00
snoke      First Order              10000.00

  6 record(s) selected.
```

10.

```
USERNAME   NAME                 BALANCE
---------- -------------------- ------------
luke       Skywalker                 8000.00
kylo       Ren                       3750.00
rey        Rey                      10000.00
finn       Finn                     11500.00
leia       Skywalker                 5750.00
snoke      First Order              10000.00

  6 record(s) selected.
```

11.

```
USERNAME   NAME                 BALANCE
---------- -------------------- ------------
luke       Skywalker                18000.00
kylo       Ren                       3750.00
rey        Rey                      10000.00
finn       Finn                     11500.00
leia       Skywalker                 5750.00
snoke      First Order                  0.00

  6 record(s) selected.
```

In session B Luke's balance is 18000.00. This is the reflection of the last query in session A, where all Snoke's balance was transferred to Luke. In step 7 we used CS (cursor stability) on session and could see the changes made by session A in session B, but could not commit actions without committing session A. In this step we used UR (uncommitted read) where both sessions can read and change each other's uncommitted work. For RS does not let other sessions see uncommitted work.

13.

```
USERNAME   NAME                 BALANCE
---------- -------------------- ------------
luke       Skywalker                 8000.00
kylo       Ren                       3750.00
rey        Rey                      10000.00
finn       Finn                     11500.00
leia       Skywalker                 5750.00
snoke      First Order              10000.00

  6 record(s) selected.
```