Database 2019 Exam Solution

Disclaimer: This is only a possible solution. Take those with a grain of salt, and please correct anything if it is incorrect, thanks!!!

1a

```sql
--Section A i)
SELECT Title
FROM MOVIE
WHERE MovieID = SOME (
    SELECT MovieID
    FROM DVD_RENTAL
)
ORDER BY Title

--ii)
SELECT DISTINCT Title, (SUM(nof_DVDs)
                        OVER (PARTITION BY Title))
                        AS number
FROM MOVIE JOIN DVDs ON MOVIE.MovieID = DVDs.MovieID
ORDER BY Title

--iii)
SELECT Address
FROM RENTER
UNION
SELECT Address
FROM DVD_STORE JOIN DVD_RENTAL
               ON DVD_RENTAL.StoreID = DVD_STORE.StoreID

--iv)
SELECT DISTINCT ProducerName AS Name
FROM MOVIE
WHERE EXISTS (
    SELECT *
    FROM DVD_RENTAL JOIN RENTER AS RENTER1
                    ON RENTER.MemberNo = DVD_RENTAL.MemberNo
    WHERE RENTER.Name LIKE 'K%'
    AND RENTER1.MovieID = MOVIE.MovieID
    AND EXISTS (
        SELECT *
        FROM DVD_RENTAL JOIN RENTER AS RENTER2
                    ON RENTER.MemberNo = DVD_RENTAL.MemberNo
        WHERE RENTER1.Name = RENTER2.Name
    AND RENTER1.MovieID <> RENTER2.MovieID
  )
)
ORDER BY Name

--v)
SELECT MemberNo
FROM DVD_RENTAL JOIN RENTER AS Movie1
        ON DVD_RENTAL.MemberNo = RENTER.MemberNo
WHERE Movie1.DateDue < CURDATE()
AND EXISTS (
  SELECT *
  FROM DVD_RENTAL JOIN RENTER AS Movie2
        ON DVD_RENTAL.MemberNo = RENTER.MemberNo
  WHERE Movie1.MemberNo = Movie2.MemberNo
  AND Movie2.DateDue < CURDATE()
)
ORDER BY MemberNo
```

1b

$\text{Contractor}(\underline{\text{Ssn}}, \text{Division}, \text{SupervisorSsn})$

$\quad \text{SupervisorSsn } \textbf{references } \text{Contractor}.\text{Ssn } \textbf{on update cascade}$

$\quad \text{Division } \textbf{references } \text{Division}.\text{Name } \textbf{on update cascade}$

$\text{Division}(\underline{\text{Name}}, \text{JobID})$

$\quad \text{JobID } \textbf{references } \text{Job}.\text{JobID } \textbf{on update cascade}$

$\text{Job}(\underline{\text{JobID}})$

$\text{Task}(\underline{\text{Name}}, \underline{\text{JobID}})$

$\quad \text{JobID } \textbf{references } \text{Job}.\text{JobID } \textbf{on delete cascade}$

$\text{Leads}(\underline{\text{ContractorSsn}}, \text{TaskName}, \text{Date})$

$\quad \text{TaskName } \textbf{references } \text{Task}.\text{Name } \textbf{on delete cascade}$

$\text{LeadsBoth}(\underline{\text{ContractorSsn}}, \text{JobID}, \text{TaskName})$

$\quad \text{ContractorSsn } \textbf{references } \text{Contractor}.\text{Ssn } \textbf{on delete cascade}$

$\quad \text{JobID } \textbf{references } \text{Job}.\text{JobID } \textbf{on delete cascade}$

$\quad \text{TaskName } \textbf{references } \text{Task}.\text{Name } \textbf{on delete cascade}$

$\text{WorksOn}(\text{ContractorSsn}, \underline{\text{JobID}}, \text{Hours})$

$\quad \text{ContractorSsn } \textbf{references } \text{Contractor}.\text{Ssn } \textbf{on delete cascade}$

$\quad \text{JobID } \textbf{references } \text{Job}.\text{JobID } \textbf{on delete cascade}$

2a

The simplified version of the functional dependency is:
$F = \{ABC \to D, ABC \to E, BC \to A, DE \to B, CE \to A, CE \to B\}$.

Since $BC \to A$, and $ABC \to D$, $ABC \to E \models BC \to DE$, we have:
$F' = \{BC \to D, BC \to E, BC \to A, DE \to B, CE \to A, CE \to B\}$.

Since $BC^+ = ABCE$ without $BC \to D$, we cannot remove this FD.

Since $BC^+ = ABCD$ without $BC \to E$, we cannot remove this FD.

Since $BC^+ = ABCDE$ without $BC \to A$, we can safely remove this FD, leaving us $F'' = \{BC \to D, BC \to E, DE \to B, CE \to A, CE \to B\}$.

Since $DE^+ = DE$ without $DE \to B$, we cannot remove this FD.

Since $CE^+ = BCDE$ without $CE \to A$, we cannot remove this FD.

Since $CE^+ = ACE$ without $CE \to B$, we cannot remove this FD.

Hence, $F_c = \{BC \to DE, DE \to B, CE \to AB\}$

2b

Since $C \to A$, $ACD \to B$, we can simplify $ACD \to B$ into $CD \to B$.

Since $(C)D \to (C)EG$, $CG \to BD$, we can remove $CD \to B$, leaving us $F' = \{AB \to C, C \to A, BC \to D, D \to E, D \to G, BE \to C, CG \to B, CG \to D, CE \to A, CE \to G\}$.

Since $CG \to B(C)$, $BC \to D$, we can remove $CG \to D$.

Since $C \to A$, we can remove $CE \to A$ as well.

Removing any other FD would not be possible, so $F_c = \{AB \to C, C \to A, BC \to D, D \to EG, BE \to C, CG \to B, CE \to G\}$

2c

First, we need to find the minimal candidate keys.

Since none of the FDs contain the attribute F, any minimal candidate key must contain F.

Since ABF^+, BCF^+, BEF^+, CGF^+, CEF^+, CDF^+ are all equal to $ABCDEFG$, they are the minimal candidate keys.

Hence, all attributes are prime and the table is already in 3NF.

2d

From 2c we know that the superkeys are ABF, BCF, BEF, CGF, CEF, CDF. Since every superkey has F, none of the FDs listed satisfy the principle of BCNF.

Since the minimal cover implies the original FDs, we will operate on the minimal cover just to save time.

We pick any of the FDs to start: take $AB \to C$ first. We decompose the relation into $R_1(A, B, D, E, F, G)$ and $R_2(A, B, C)$. We then project the FDs onto R_1 and R_2.

Take $C \to A$ for relation R_2, since C is not a superkey, we decompose the relation R_2 into $R_3(B, C)$ and $R_4(C, A)$.

Take $D \to EG$ for relation R_1, we decompose the relation R_1 into $R_5(A, B, D, F)$ and $R_6(D, E, G)$.

Hence, we have the relations R_3, R_4, R_5, R_6 in BCNF. Notice that this decomposition does not preserve the functional dependencies and it is not alway possible to preserve FDs in BCNF decomposition. In this case, preserving FDs is not possible.

2e

First, we need to find the minimal candidate keys.

The only possible minimal candidate key is AB since on the right hand side of all the FDs, there is no A and B, so A and B must be in any possible superkey.

We pick any of the FDs that violates the principle of BCNF to start: take $A \to DE$, we decompose the relation intop $R_1(A, B, C, F, G, H, I, J)$ and $R_2(A, D, E)$.

Take $B \to F$, we decompose the relation R_1 into $R_3(A, B, C, G, H, I, J)$ and $R_4(B, F)$.

Take $AB \to C$ on R_3, we decompose R_3 into $R_5(A, B, G, H, I, J)$ and $R_6(A, B, C)$.

Hence, we have the relation R_2, R_4, R_5, R_6 in BCNF. However, this decomposition does not preserve all the FDs. The FDs that are left out are $F \to GH$ and $D \to IJ$. So we need $R_7(F, G, H)$ and $R_8(D, I, J)$ as well. To eliminate redundancy, we discard R_5 and hence the final result of BCNF with FD preservation is R_2, R_4, R_6, R_7, R_8.