**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**

1--Section A

i) 2SELECT Title 3FROM MOVIE 4WHERE MovieID = SOME ( 5SELECT MovieID 6FROM DVD\_RENTAL 7)8ORDER BY Title 910

--ii) 11SELECT DISTINCT Title, (SUM(nof\_DVDs) 12OVER (PARTITION BY Title)) 13AS number 14FROM MOVIE JOIN DVDs ON MOVIE.MovieID = DVDs.MovieID 15ORDER BY Title 1617

--iii) 18SELECT Address 19FROM RENTER 20UNION 21SELECT Address 22FROM DVD\_STORE JOIN DVD\_RENTAL 23ON DVD\_RENTAL.StoreID = DVD\_STORE.StoreID 2425

--iv) 26SELECT DISTINCT ProducerName AS Name 27FROM MOVIE 28WHERE EXISTS ( 29SELECT \* 30FROM DVD\_RENTAL JOIN RENTER AS RENTER1 31ON RENTER.MemberNo = DVD\_RENTAL.MemberNo 32WHERE RENTER.Name LIKE 'K%' 33AND RENTER1.MovieID = MOVIE.MovieID 34AND EXISTS ( 35SELECT \* 36FROM DVD\_RENTAL JOIN RENTER AS RENTER2 37ON RENTER.MemberNo = DVD\_RENTAL.MemberNo 38

WHERE RENTER1.Name = RENTER2.Name

AND 39RENTER1.MovieID <> RENTER2.MovieID 40) 41)42ORDER BY Name 4344

SELECT Name

FROM PRODUCER

WHERE EXISTS (SELECT MemberNo

FROM RENTER

NATURAL JOIN DVD\_RENTAL

NATURAL JOIN MOVIE  
 WHERE Name LIKE ‘K%’

AND PRODUCER.Name = ProducerName

GROUP BY MemberNo

HAVING COUNT(DISTINCT MovieID) >= 2)

ORDER BY Name

--v) 45SELECT MemberNo 46FROM DVD\_RENTAL JOIN RENTER AS Movie1 47ON DVD\_RENTAL.MemberNo = RENTER.MemberNo 48WHERE Movie1.DateDue < CURDATE() 49AND EXISTS ( 50SELECT \* 51FROM DVD\_RENTAL JOIN RENTER AS Movie2 52ON DVD\_RENTAL.MemberNo = RENTER.MemberNo 53WHERE Movie1.MemberNo = Movie2.MemberNo 54AND Movie2.DateDue < CURDATE() 55)56

ORDER BY MemberNo

v) Why not simply

SELECT MemberNo

FROM DVD\_RENTAL

WHERE CURDATE() > DateDue

GROUP BY MemberNo

HAVING COUNT(\*) > 1  
ORDER BY MemberNo

?

**1b 2a**

The simplified version of the functional dependency is:

.

Since , and , , we have:

.

Since without , we cannot remove this FD.

Since without , we cannot remove this FD.

Since without , we can safely remove this FD, leaving us

.

Since without , we cannot remove this FD.

Since without , we cannot remove this FD.

Since without , we cannot remove this FD.

Hence,

**2b**

Since , , we can simplify into .

Since , , we can remove , leaving us

.Since , , we can remove .

Since , we can remove as well.

Removing any other FD would not be possible, so

**2c**

First, we need to find the minimal candidate keys.

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

Since , , , , , are all equal to , 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 , , , , , . Since every superkey has , 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 first. We decompose the relation into

and . We then project the FDs onto and .

Take for relation , since is not a superkey, we decompose the relation into

and .

Take for relation , we decompose the relation into and

.

Hence, we have the relations 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 since on the right hand side of all the FDs, there is no and , so and must be in any possible superkey.

We pick any of the FDs that violates the principle of BCNF to start: take , we decompose the relation intop and .

Take , we decompose the relation into and .

Take on , we decompose into and .

Hence, we have the relation in BCNF. However, this decomposition does not preserve all the FDs. The FDs that are left out are and . So we need

and as well. To eliminate redundancy, we discard and hence the final result of BCNF with FD preservation is .