2.a) the candidate key can be A or C or D or H.

b) The canonical cover is: A->BCE, C->D, H->IA, D->H

c)It is in BCNF form.

d)It is dependency preserving.

3.a)This is not a good design for following reasons:

1) Current schema does not reflect a lot of functional dependencies. For instance, as assumptions in the question, {mid → mtitle} holds, but since the {mid} is not a candidate key, we will end up repeating this information in several tuples.

2) Inserting values for the separate entities in one table like Mid or Midtitle will lead to storing NULL and duplicate values at several places. This will increase the storage and our data will be inconsistent.

3) We have to access this one table for querying even minimal data, this will be time consuming as everything is under this table.

4) Lastly, this schema does not adhere to the normalization rules. As everything is under one table, it will make maintenance and querying from the database difficult and it will be hard to see relation between each attribute. it's better to divide this big table to many smaller ones.

b) aid->aname

mid->mtitle

mid,rolename->payph

aid,mid,rolename->hours

c)aid, mid, rolename

d) aid->aname

mid->mtitle

mid, rolename->payph

aid,mid, rolename->hours

e)No, it's not in BCNF. Because for schema to be BCNF for each non-trivial dependency A->B, A should be a superkey, which is not the case here.

Convert into BCNF form:

Actors(aid,aname)

Movies(mid,mtitle)

Pay(mid, rolename, payph)

Acthour(aid, mid, rolename, hours)

f) Yes, the schema in section e) is dependency preserving, because all the functional dependencies can be checked in BCNF form.

g) The candidate key is {aid,mid,rolename}.

Functional dependencies:

Aid->aname

mid->mtitle

aname, mid-> payph

mid, rolename->payph

aid, mid, rolename->hours

Canonical cover:

Aid->aname

mid->mtitle

aname, mid-> payph

mid, rolename->payph

aid, mid, rolename->hours

No, it is not in BCNF form, convert it into BCNF:

Actors(aid,aname)

Movies(mid,mtitle)

Payment(aid,mid,payph)

Acthours(aid, mid, rolename, hours)

No, it is not dependency preserving because we cannot check aname, mid-> payph in the result of Acthours.

3NF form: Actors(aid,aname)

Movies(mid,mtitle)

Payment(aid,mid,payph)

Acthours(aid, mid, rolename, hours)

ActorPay(aname,mid, payph)

4.a)select data_type, count(*)

from columns

where table_schema="dbhw3"

group by data_type;

```
mysql> select data_type, count(*)
    -> from columns
    -> where table_schema="dbhw3"
    -> group by data_type;
+-----------+----------+
| data_type | count(*) |
+-----------+----------+
| datetime  |        2 |
| int       |       13 |
| varchar   |       10 |
+-----------+----------+
3 rows in set (0.01 sec)
```

b) select c1.table_name, c2.table_name

from columns c1,columns c2

where c1.column_name=c2.column_name and c1.table_name>c2.table_name and c1.table_schema="dbhw3" and c1.table_schema=c2.table_schema ;

```
mysql> select c1.table_name, c2.table_name
    -> from columns c1,columns c2
    -> where c1.column_name=c2.column_name and c1.table_name>c2.table_name an
1.table_schema="dbhw3" and c1.table_schema=c2.table_schema ;
+------------+------------+
| table_name | table_name |
+------------+------------+
| contain    | cake       |
| orders     | cake       |
| ingredient | cake       |
| orders     | contain    |
| ingredient | contain    |
| orders     | customer   |
+------------+------------+
6 rows in set (0.22 sec)
```

c) select count(*)

from views

where table_schema="dbhw3";

```
mysql> select count(*)
    -> from views
    -> where table_schema="dbhw3";
+----------+
| count(*) |
+----------+
|        0 |
+----------+
1 row in set (0.00 sec)
```

d) select c.*

from dbhw3.customer c ,columns

where table_schema="dbhw3" and c.lastname = columns.column_name;

```
mysql> select c.*
    -> from dbhw3.customer c ,columns
    -> where table_schema="dbhw3" and c.lastname = columns.column_name;
Empty set (0.01 sec)
```

e) select table_name, column_name

from key_column_usage

where table_schema="dbhw3" and constraint_name="PRIMARY";

```
mysql> select table_name, column_name
    -> from key_column_usage
    -> where table_schema="dbhw3" and constraint_name="PRIMARY";
+------------+-------------+
| table_name | column_name |
+------------+-------------+
| cake       | cakeid      |
| contain    | cakeid      |
| contain    | ingredid    |
| customer   | custid      |
| ingredient | ingredid    |
| orders     | custid      |
| orders     | cakeid      |
| orders     | ordertime   |
+------------+-------------+
8 rows in set (0.00 sec)
```

f)select column_name

from columns

where table_schema="dbhw3" and column_name like '%name%' ;

```
mysql> select column_name
    -> from columns
    -> where table_schema="dbhw3" and column_name like '%name%' ;
+-------------+
| column_name |
+-------------+
| cakename    |
| firstname   |
| lastname    |
| iname       |
+-------------+
4 rows in set (0.00 sec)
```