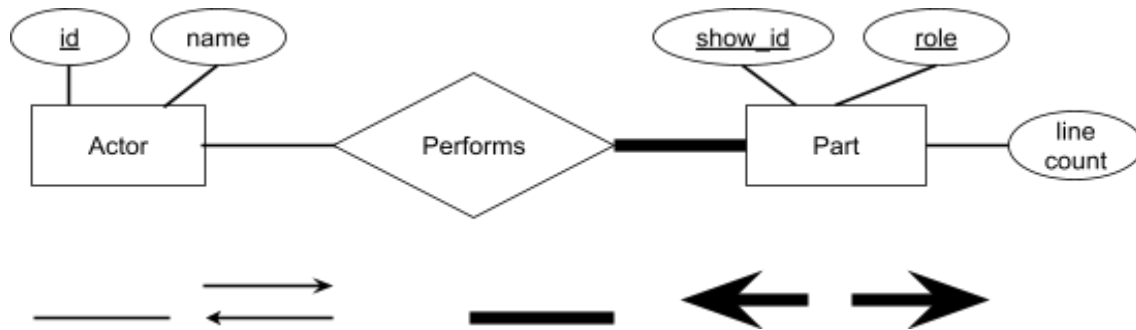
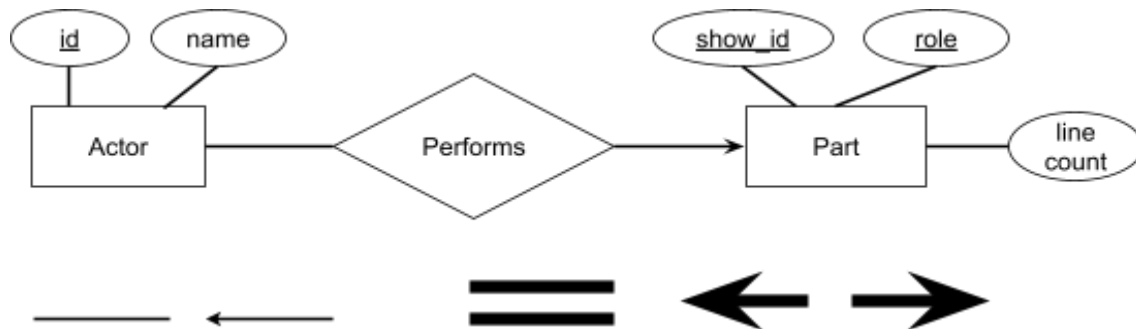


Problem Set 1: Part I
Problem 1: ER diagram basics

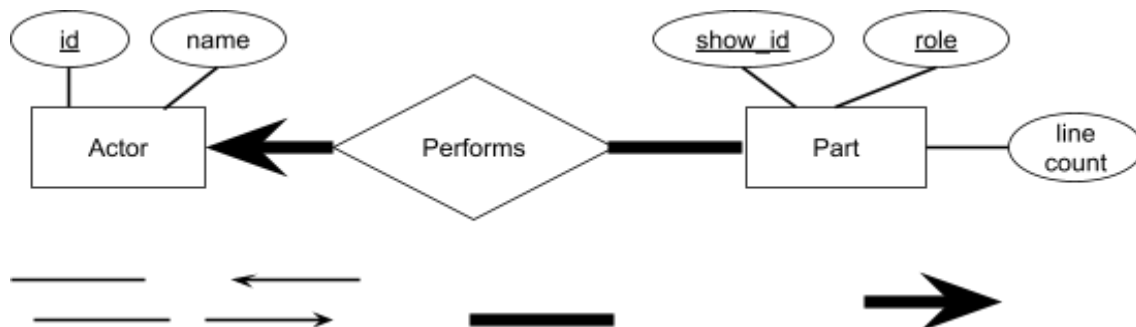
1.



2.



3.



4.

Yes, it is possible to convert the diagram into a relational schema that has only two tables/relations. A relation for the relationship set is not needed. For efficiency, the attributes from the relationship set(Performs) are captured in the entity set on the many side (Actor) of the relationship.

Actor(id, name, show id)
 Part(show id, role, line count)

Problem 2: Database design

1. Many-to-one:

Borrows is many-to-one from Customer to Loan

HasType is many-to-one from Account to AccountType

2. Constraints:

Each customer borrows at most one loan

Each account has one account type

Each loan is borrowed by at least one customer

Each account has at least one account type

Each account is owned by at least one customer

3. Relational schema (you may not need all of the rows)

Relation	Foreign keys (if any)
Customer(<u>id</u> , name, address, loan)	Loan, which refers to Loan.id Id, which refers to Owns.customer id
Loan(<u>id</u> , principal, interest, interest rate)	
Owns(<u>customer id</u> , account id)	Customer id, which refers to Customer.id account id, which refers to Account.id
Account(<u>id</u> , balance, AccountType)	Id, which refers to Owns.account id AccountType, which refers to AccountType.id
AccountType(<u>id</u> , name, interest rate)	

Problem 3: Combining relations

Use the Insert->Table menu option to insert an appropriate table for each answer.

1. Cartesian product

R.a	R.b	R.c	S.d	S.b	S.a
1	2	3	3	2	1

1	2	3	5	4	3
1	2	3	9	8	5
3	4	6	3	2	1
3	4	6	5	4	3
3	4	6	9	8	5
5	6	1	3	2	1
5	6	1	5	4	3
5	6	1	9	8	5

2. Natural join

a	b	c	d
1	2	3	3
3	4	6	5

3. Left outer join

a	b	c	d
1	2	3	3
3	4	6	5
1	2	3	null
5	6	1	null

4. Right outer join

a	b	c	d
1	2	3	3
3	4	6	5
5	8	null	9

5. Full outer join

a	b	c	d
1	2	3	3
3	4	6	5
1	2	3	null
5	6	1	null
5	8	null	9