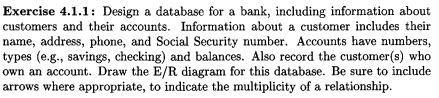
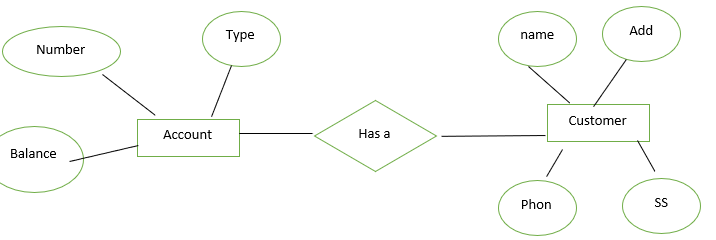
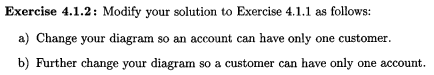
# Exercise 4.1.1 page X(pdf)





# Exercise 4.1.2 page 188(pdf)

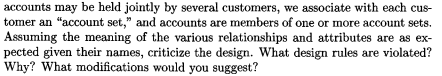


a)  
When we want to have an account so ONLY one customer used it we have to user to the customer.

b) for account is also same as customer.

# Exercise 4.2.1 page 188(pdf)

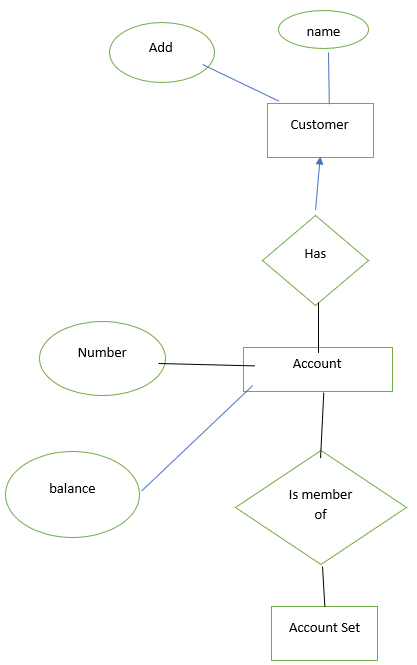




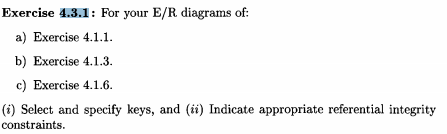
Redundancy:

For address there is no need to have separate entity we just can have it as attribute.

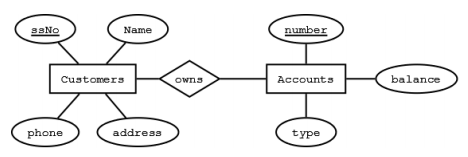
For Account set , Owner address is also duplicated , just remove it. Since there’s a 1-to-1 between customers and AccSets



# Exercise 4.3.1 page 188(pdf)



a)



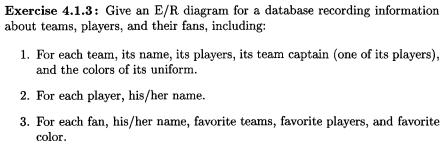
Give the keys:

Key for customers are here the social security number.  
Key for accounts is the account number.

Indicate appropriate referential integrity constraints:

No referential integrity constraints are appropriate here.

b)



Keys:

Teams will have it’s name as the key value.  
Players will have it’s name as the key value.  
Fans will have their name as the key value.

Assuming all their names are unique.

Restrictions:

Each player in a team must be present in the players entity

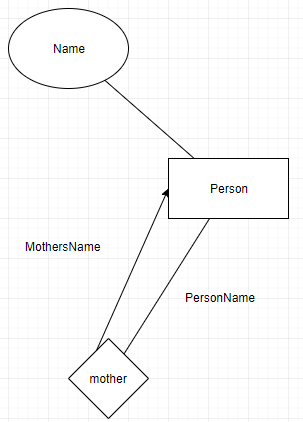
Each captain on a team must be present in the players entity

Each fans favorite player must be present in the players entity

Each fans favorite team must be present in the teams entity.

c)

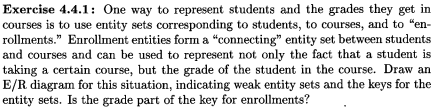


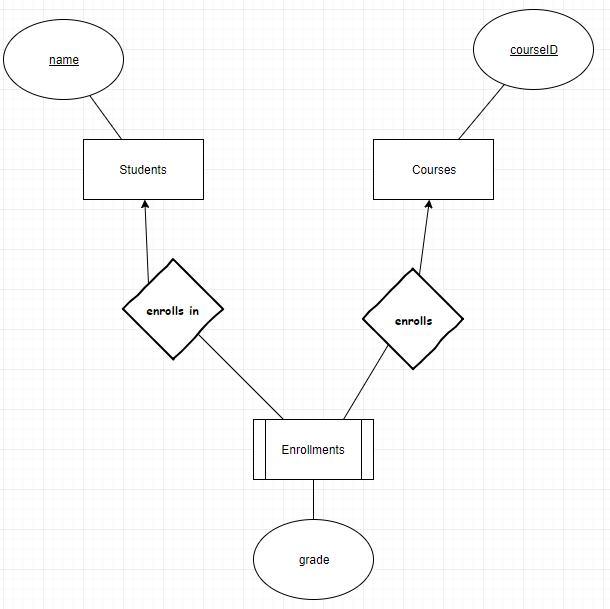


The key for the entity will be the persons name assuming the names are unique.

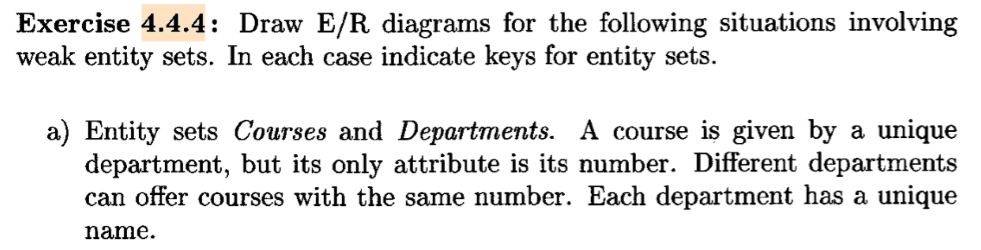
We will need referential integrity constraints for each of the attribute that denotes who the person are related to, meaning that the mother the father and the children will have to be in the people set as a name as well.

# Exercise 4.4.1 page 193





Enrollments is a weak entity-set getting the key values from Courses and Students. As this match is already unique the grade attribute in Enrollments is not part of the key.



Courses (number)

Departments (name)

Departments

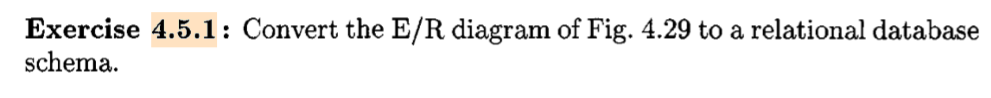
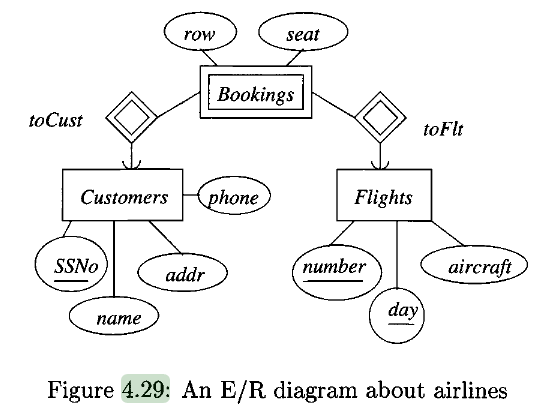
Given by

Courses

Name

Number

Always the same, binary one to many, key atr from strong also key atr for weak

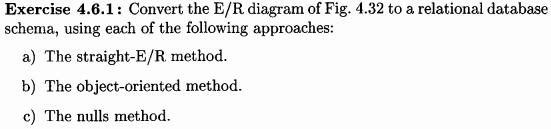


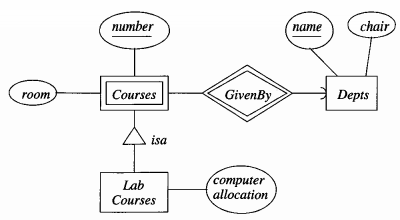
Customers(SSNo, name, address, phone)

Flights(number, day, aircraft)

Bookings(CSSNo, Fnumber, Fday, row, seat)

Not necessary to convert the supporting relations for  
a weak entity set.

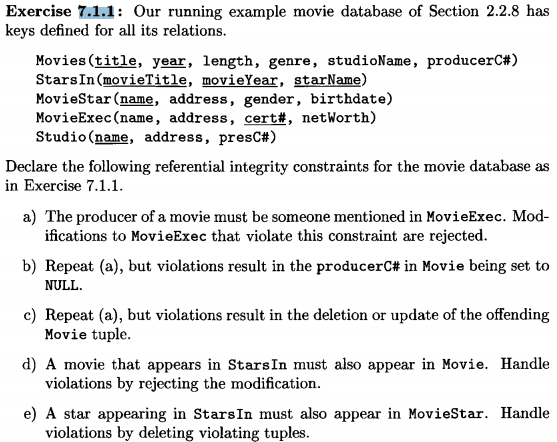




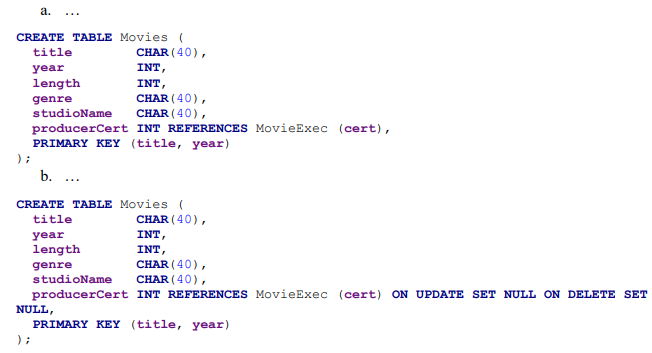
Straight: Kun keys gives videre gennem isa  
Object oriented: Alt gives videre, og hvis flere laves case for hvis en er begge(ikke I det her eksempel)  
Nulls: Lave en for alle muligheder. Hvis et kursus ikke er et lab kursus har den bare NULL I computer allocation.

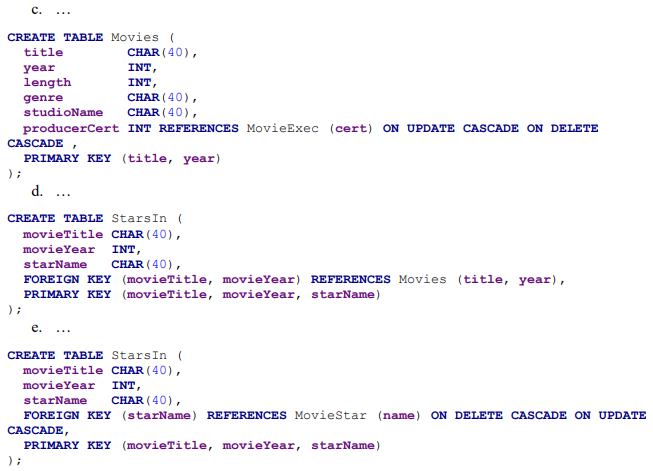
|  |
| --- |
| a) |
| Kun |  |
|  | Depts(name, chair) |
|  | Courses(number, room, DeptName) |
|  | LabCourses(number, DeptName, computerAllocation)  Ikke laves - GivenBy(number, DeptName, name) |
|  |  |
|  | b) |
|  |  |
|  | Depts(name, chair) |
|  | Courses(number, room, DeptName) |
|  | CoursesAndLabCourses(number, room, DeptName, computerAllocation) |
|  |  |
|  | c) |
|  |  |
|  | Depts(name, chair) |
|  | Courses(number, room, DeptName, computerAllocation) |

# Exercise 7.1.1

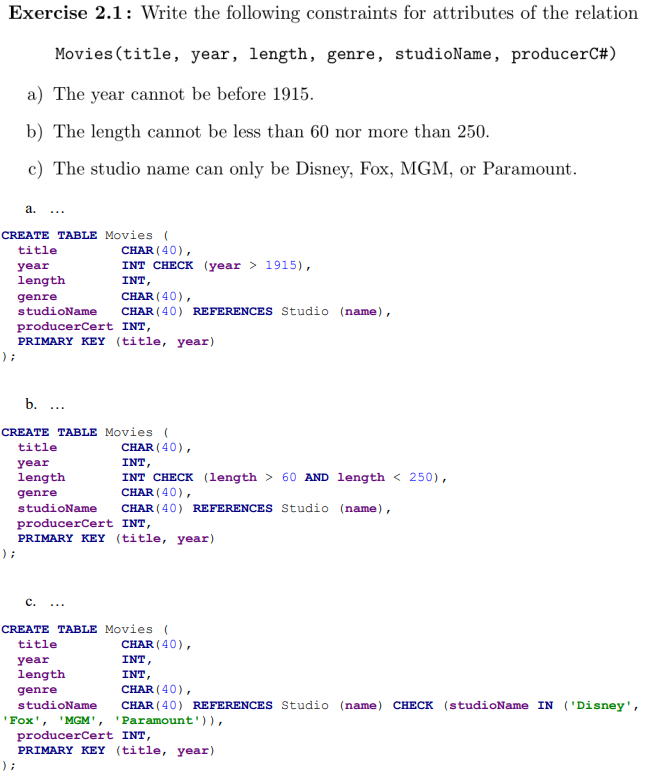


Policies  
Standard: Afviser alle ændringer.  
Cascade: Hvis ændring ikke overholder ændres der også hvor referencen kommer fra.  
SetNull: Sæt referencerne til NULL. Hvis vi sletter fra MovieExec sættes de til NULL I Movies.

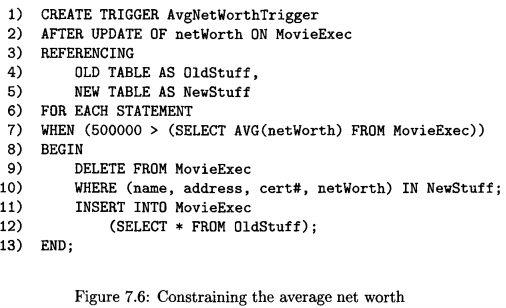




|  |
| --- |
| a) |
|  | ALTER TABLE Movies |
|  | ADD CONSTRAINT FK\_Producer |
|  | FOREIGN KEY (producer#C) REFERENCES MovieExec(cert#); |
|  |  |
|  | b) |
|  | ALTER TABLE Movies |
|  | ADD CONSTRAINT FK\_Producer |
|  | FOREIGN KEY (producer#C) REFERENCES MovieExec(cert#) |
|  | ON DELETE SET NULL |
|  | ON UPDATE SET NULL; |
|  |  |
|  | c) |
|  | ALTER TABLE Movies |
|  | ADD CONSTRAINT FK\_Producer |
|  | FOREIGN KEY (producer#C) REFERENCES MovieExec(cert#) |
|  | ON DELETE CASCADE |
|  | ON UPDATE CASCADE; |
|  |  |
|  | d) |
|  | ALTER TABLE StarsIn |
|  | ADD CONSTRAINT FK\_movieTitle |
|  | FOREIGN KEY (movieTitle) REFERENCES Movies(title); |
|  |  |
|  | e) |
|  | ALTER TABLE StarsIn |
|  | ADD CONSTRAINT FK\_starName |
|  | FOREIGN KEY (starName) REFERENCES MovieStar(name) |
|  | ON DELETE CASCADE |
|  | ON UPDATE CASCADE; |



# Exercise 7.5.1





|  |
| --- |
| CREATE TRIGGER AvgNetWorthTrigger |
|  | AFTER INSERT ON MovieExec |
|  | REFERENCING |
|  | NEW ROW AS newRow |
|  | FOR EACH ROW |
|  | WHEN (500.000 > (SELECT AVG(newWorth) FROM MovieExec)) |
|  | BEGIN |
|  | DELETE FROM MovieExec |
|  | WHERE (name, address, cert#, netWorth) IN newRow; |
|  | END; |
|  |  |
|  | CREATE TRIGGER AvgNetWorthTrigger |
|  | AFTER DELETE ON MovieExec |
|  | REFERENCING |
|  | OLD ROW AS oldRow |
|  | NEW ROW AS newRow |
|  | FOR EACH ROW |
|  | WHEN (500.000 > (SELECT AVG(newWorth) FROM MovieExec)) |
|  | BEGIN |
|  | INSERT INTO MovieExec |
|  | WHERE (name, address, cert#, netWorth) IN OldStuff AND |
|  | (name, address, cert#, netWorth) NOT IN NewStuff; |
|  | END; |

# Exercise 4.2.2 page 183

When there is exactly one Studios and exactly one President, we can combine the two entity sets into a single entity.

# Exercise 4.2.3 page 183

1. All relationships in which E is involved must have arrows entering E. That is, E must be the “one” in many-one relationships, or its generalization for the case of multiway relationships.

2. If E has more than one attribute, then no attribute depends on the other attributes, the way address depends on name for Studios. That is, the only key for E is all its attributes.

3. No relationship involves E more than once

a) If there is a many-one relationship R from some entity set F to E, then remove R and make the attributes of E be attributes of F, suitably renamed if they conflict with attribute names for F. In effect, each F-entity takes, as attributes, the name of the unique, related identity.2 For instance, Movies entities could take their studio name as an attribute, should we dispense with studio addresses.

b) If there is a multiway relationship R with an arrow to E, make the attributes of E be attributes of R and delete the arc from R to E. An example of this transformation is replacing Fig. 4.8, where there is an entity set Salaries with a number as its lone attribute, by its original diagram in Fig. 4.7.

The three constraints for being allowed to replace entity Studios is in this case uphold. As in case b) it’s seen that Studios is part of a multiway relationship and therefore the remaining attribute Name is moved to the Contracts relation and Studios is removed.

# Exercise 4.2.4 page 183

In order to give the entity sets attributes allowing us to still replace them by attributes is for every attribute to be part of the key for the entity for 2) to be uphold. For Stars and Movies we can see that the relations are many-to-one pointing at Stars and movies so 1) is uphold. They also have no relations involving them more than once so 3) is uphold.

a)

For stars we can give the two attributes firstName and lastName where they are both needed as key. We can then use scenario a) to remove Stars and add the two attributes to the Contracts entity.

b)

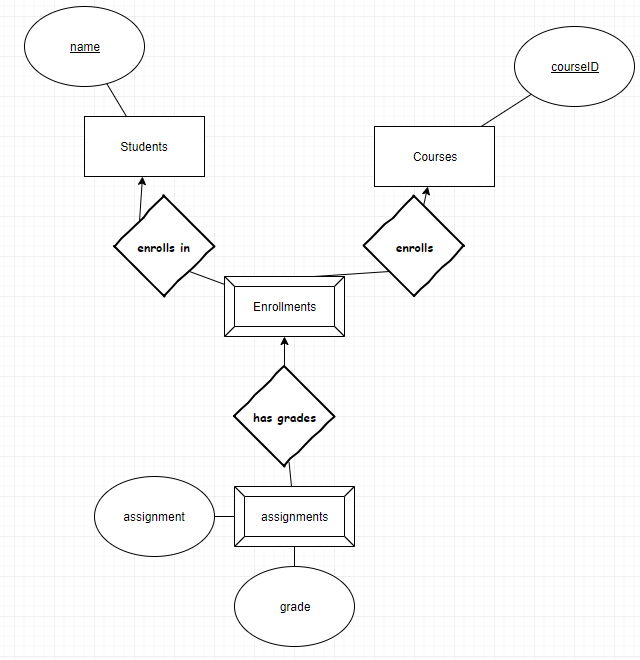
For Movies we can give it Title and year to form a key together and to the same as for Stars

c)

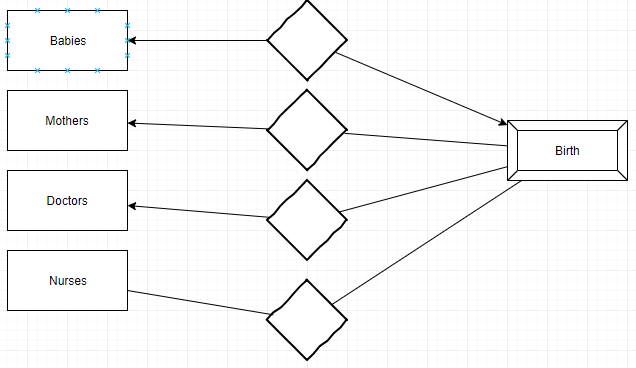
It’s not possible for Studios to be replaced by an attribute as there’s a many-to-many relation between Contracts and

# Exercise 4.4.2 page 183

Assignments is in a many-to-many relation with Enrollments as each enrollment can have multiple assignments, but assuming assignments can also be group assignments they can each also have multiple enrollments. Assignments is also a weak entity as it requires keys from an enrollment to be unique.



# Exercise 4.4.3 page 183



Key attributes from entity sets that are reached by certain many-one relationships from E to other entity sets. These many-one relationships are called supporting relationships for E, and the entity sets reached from E are supporting entity sets.

Birth is a weak entity set with supporting relationships from Babies, Mothers and Doctors.

Assuming the 4 entities Babies, Mothers, Doctors and nurses all have a unique name those will be the keys otherwise they could have their social security number.