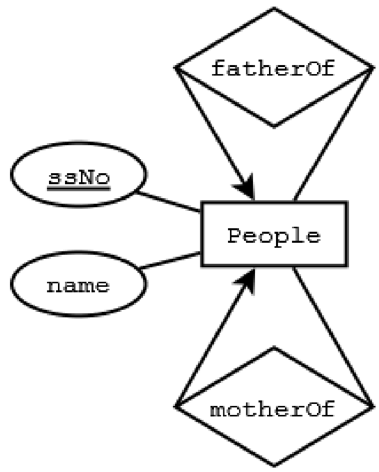
Solution to Assignment 2

NOTE: The only acceptable submission is through EAS.
We don't have the time or disk space to accept email submission.

1.



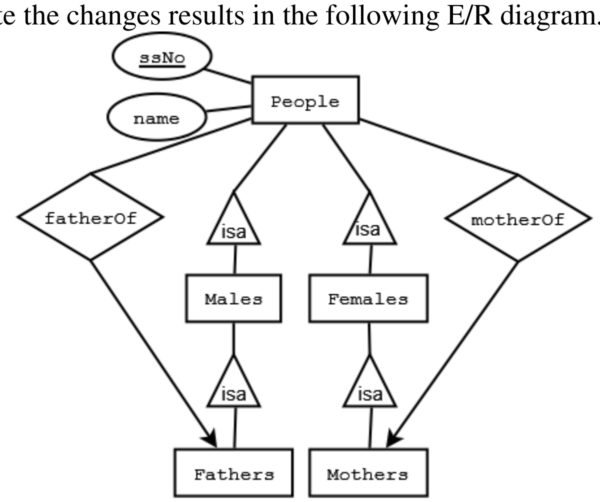
(Figure: by the textbook authors)

Converting the above diagram into relations, we get the following 3 relations.

People (ssNo, name)
MotherOf (Child-ssNo, mother-ssNo)
FatherOf (Child-ssNo, father-ssNo)

As MotherOf and FatherOf relationships are many-to-one, the two may be merged and replaced by the single relation Parents(Child-ssnNo, Mother-ssNo, Father-ssNo).

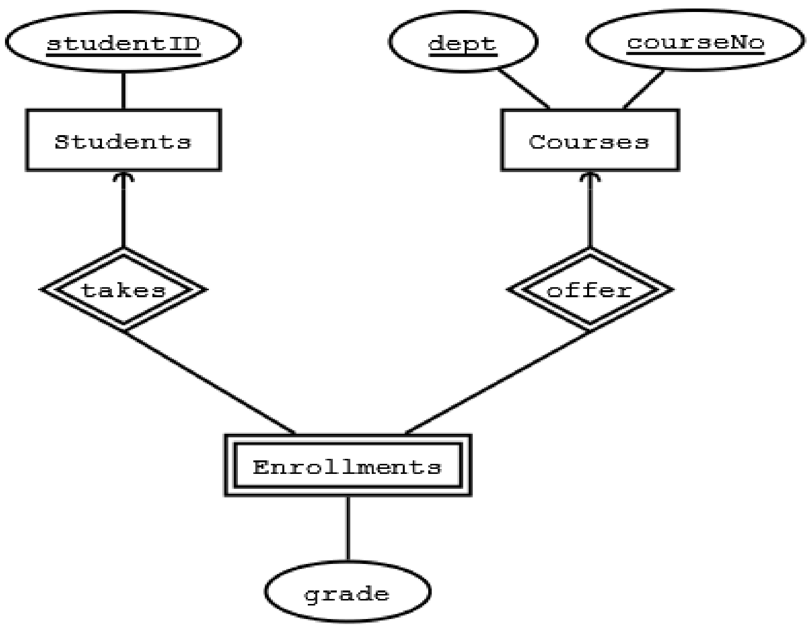
2. One way to accommodate the changes results in the following E/R diagram.



(Figure: by the textbook authors)

4.4.1. The following diagram represents the information. The weak entity set Enrollment is connected by two binary, many to one weak relationships to entity sets Students and Courses. The key

studentID (in Students) together with *dept* and *courseNo* (in Courses) form part of the key attributes of Enrollment.
Concerning the grade, it should not be a key attribute of Enrollment, in general. However, if the students, in the world being modeled, don't fail courses, then *grade* won't be needed as part of the key. However, if this is not the reality, *grade* has to be part of the key. To be more realistic, we need to consider new attributes such as semester and year in Enrollment to for a key together with the attributes *studentID*, *dept*, and *courseNo*.



(Figure: by the authors)

4.5.4(a).

Stars (<u>str-name</u>, addr)
Studios (<u>std-name</u>, addr)
Movies (<u>title</u>, <u>year</u>, genre, length)
Contracts (<u>title</u>, <u>year</u>, <u>str-name</u>, <u>std-name</u>, salary)

Depending on other information not shown in the E/R diagram, std-name may not be needed as an attribute of Contract or being part of its key.

4.6.2(a).

Person (<u>name</u>, <u>address</u>)
Child (<u>name</u>, <u>address</u>)
Father (<u>name</u>, <u>address</u>)
Mother (<u>name</u>, <u>address</u>)
Married (<u>Hname</u>, <u>Haddress</u>, Wname, Waddress)
/* For Married, we could also consider wife's info: {Wname, Waddress} as the key */
FatherOf (<u>Fname</u>, <u>Faddress</u>, Cname, Caddress)
MotherOf (<u>Mname</u>, <u>Maddress</u>, Cname, Caddress)
ChildOf (<u>Cname</u>, <u>Caddress</u>, Pname, Paddress)

On the basis of the given information, relationships and their multiplicities, we can consider the following database schema with the same information content as the above but with fewer relations, justified as follows. We can merge MotherOf and FatherOf relationships into one relation, since they are many-to-one from Child. The one-to-one relationship Married is modeled by including spouse's information in FatherOf (or MotherOf) relation.

Person (name, address)
ChildOf (Cname, Caddress, Pname, Paddress)
Child (Cname, Caddress, Fname, Faddress, Mname, Maddress)
FatherOf (Fname, Faddress, Childname, Childaddress)
Mother (name, address)

(b). In an OO approach, each object belongs to exactly one class. Also note the relationships from a class A to B could be modeled as a property of type *relationship* when defining A.

Person (name, address)
PersonChild (name, address)
PersonChildFather (name, address)
PersonChildMother (name, address)
PersonFather (name, address)
PersonMother (name, address)
ChildOf (Pname, Paddress,Cname, Caddress)
FatherOf (Cname, Caddress, Fname, Faddress)
MotherOf (Cname, Caddress, Mname, Maddress)
Married (Hname, Haddress, Wname, Waddress)

(c). Following the null approach, we get relations Person and ChildOf defined below.
Note that null will be used for instance when a person is not married or when she/he has no child.

Person (Pname,Paddress,Fname,Faddress,Mname,Maddress,Hname,
Haddress,Wname,Waddress)
/*Note: W stands for Wife and H for Husband */
ChildOf (Cname, Caddress, Pname, Paddress)

4.6.4(a). When converting an *isa* hierarchy, taking an E/R style approach, we get one relation for each entity set E in the hierarchy. In this question, this means we get exactly *e* relations, so the minimum and maximum numbers of relations generated are *e*. The relation at the root has *a* attributes, *k* of which form the key. Any other relation in the hierarchy has these *k* attributes from the root plus *a* number of they each have. Thus in this approach we get exactly e relations and the minimum attributes they have are *a* (for the root relation) and the maximum is $a+k$ (for other relations in the hierarchy).

(b). Recall that we get many more relations in an OO approach by considering all possible subtrees of the root. The minimum number of such subtrees is *e* (when the structure of the *isa* hierarchy is a strict hierarchy) and the maximum number of such trees is 2^{e-1} (when the isa hierarch is a two-level tree structure in which there is a relation at the root and every other relation is a child of the root. For instance, for entity sets A, B, and C in the hierarchy with A as the root (when $e=3$), the classes we get for the strict structure would be A, AB, ABC, and for the two-level structure, we get the 4 classes, A, AB, AC, ABC.

The minimum number of attributes we may have is *a* (for the class that created for the root) and the number of attributes we may have in other classes could be at most *a**e*.

(c). For the Null approach, we create one relation for the entire *isa* hierarchy. The number of attributes in this relation is exactly *a**e*.

3.2.1(a). The question asks to find non-trivial new FD's of the form X→ A that hold on R, in which X is a subset of R and A is a single attribute in R. This is done by computing the closure of every subset of R with respect to the FD's provided.
Starting from the singletons: $\{A\}^+=\{A\}$, $\{B\}^+=\{B\}$, $\{C\}^+=\{A,C,D\}$, $\{D\}^+=\{A,D\}$. Thus, the only new, non-trivial FD we get here is (1) C→A.
Next, we compute the closure of every subset of two attributes of R, as follows.

$\{A,B\}^+ = \{A,B,C,D\}$	(2) AB→D
$\{A,C\}^+ = \{A,C,D\}$	(3) AC→D
$\{A,D\}^+ = \{A,D\}$	/*this yields no non-trivial FD's*/
$\{B,C\}^+ = \{A,B,C,D\}$	(4) BC→A, (5) BC→D
$\{B,D\}^+ = \{A,B,C,D\}$	(6) BD→A, (7) BD→C
$\{C,D\}^+ = \{A,C,D\}$	(8) CD→A

We next compute the closure of every subset of R of size 3, as follows.

$\{A,B,C\}^+ = \{A,B,C,D\}$	(9) ABC→D
$\{A,B,D\}^+ = \{A,BC,D\}$	(10) ABD→C
$\{A,C,D\}^+ = \{A,C,D\}$	/*this yields no nontrivial FD's*/
$\{B,C,D\}^+ = \{A,B,C,D\}$	(11) BCD→A

This completes the solution, noting that we get no non-trivial FD for $\{A,B,C,D\}^+$.

(b) Note that keys are minimal, by definition. Based on the above, we look for minimal subset X of R for which $X^+=R$. Following this, we get 3 keys for R: AB, BC, and BD.

(c) To find the superkeys of R that are not minimal, we begin with each key K of R to which we include non-empty subsets of R-K. Doing this for the key K=AB, we get the 3 non minimal superkeys: ABC, ABD, and ABCD. From BC, we get just BCD, and from BD, we get ABD. The list of non minimal superkeys of R includes: ABC, ABD, ABCD, BCD, and ABD.

3.3.4(a). A counter-example would be an instance of relation R(A,B) that has two tuples: (1,0) and (2,0). As can be seen, this relation instance trivially satisfies the FD: A→B but not B→A.
(b). A counter-example for this would be an instance of relation R(A,B,C) that has the two tuples: (1,1,0), and (2,1,1). Again note that here, this instance trivially satisfies the FD's: AB→C and A→C, but not B→C.
(c). A counter-example for this would be an instance of relation R(A,B,C) that includes three tuples: (0,0,0), (1,0,1), and (1,1,0). Here, the FD: AB→C is trivially satisfied, however, presence of the first 2 tuples indicates violation of B→C and presence of the last two indicates violation of A→C.

3.2.10(a). We compute the closures of every subsets of attributes in S={A,B,C} with respect to the given set of FD's (except the empty set and S). We can also remove D and E from the closures as these attributes are not "relevant" to S.

(a). This yields 3 relevant and non-trivial FD's, as follows:

$\{A\}^+=A$
$\{B\}^+=B$
$\{C\}^+=ACE$ (1) $C\rightarrow A$
$\{AB\}^+=ABCDE$ (2) $AB\rightarrow C$
$\{AC\}^+=ACE$
$\{BC\}^+=ABCDE$ (3) $BC\rightarrow A$

(b). Following the solution idea above, the only non-trivial FD that is relevant to S(A,B,C) from the closure of {A,C} is $AC\rightarrow B$, as follows; all other FD's we get by computing the closure of other subsets of S are either trivial or not relevant to S.

$\{A\}^+=AD$
$\{B\}^+=B$
$\{C\}^+=C$
$\{AB\}^+=ABDE$
$\{AC\}^+=ABCDE$ (1) $AC\rightarrow B$
$\{BC\}^+=BC$