COS 597A, Fall 2011
Solutions to Problem Set 1: problems 2 and 4
The Movie Database

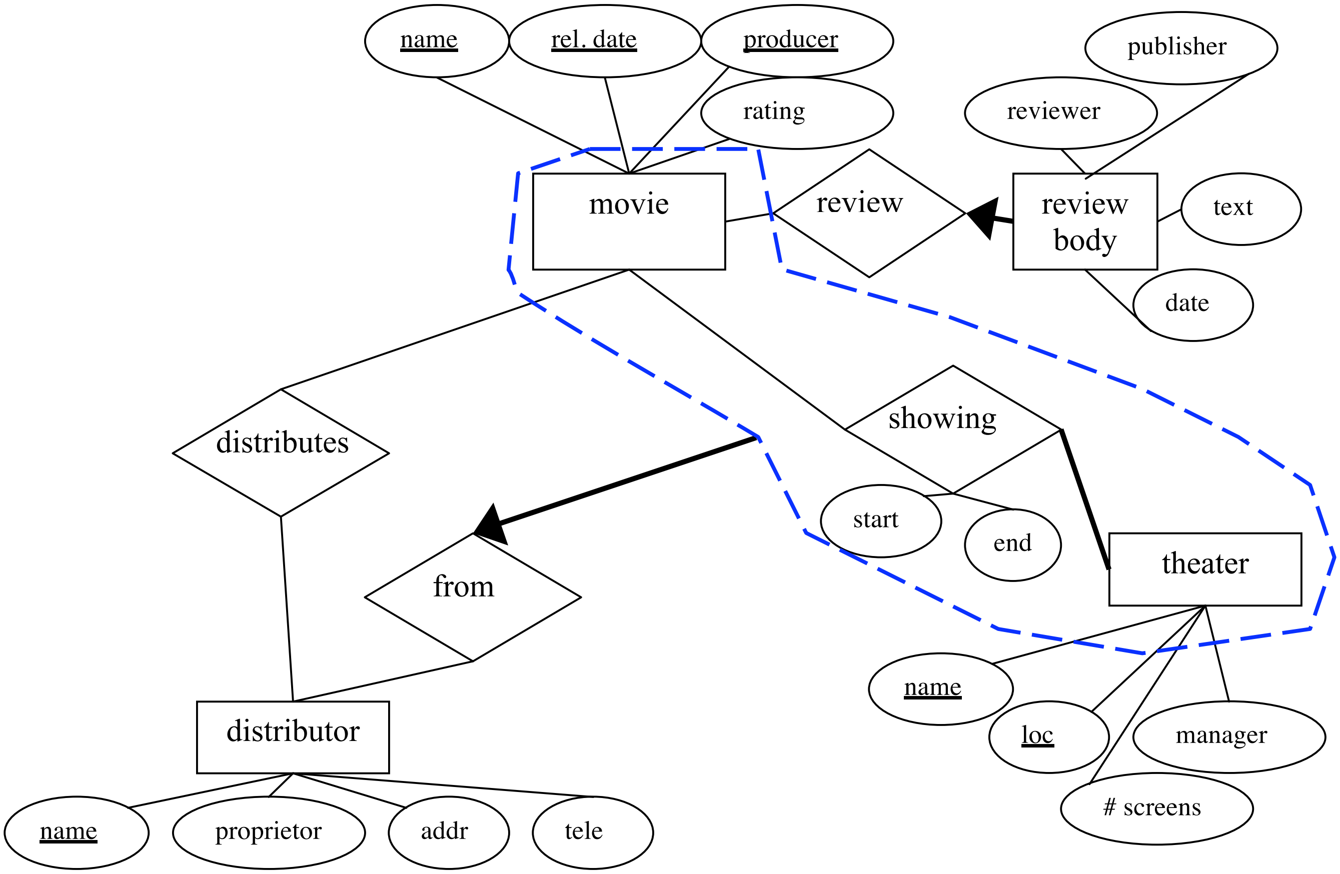
Problem 2:

Entity key constraints:

For movie: name, producer, release date
For theater: name, location
For distributor: business name

Other constraints from the prose description:

Numbers of screens ≥ 1.
There is only one distributor for any one movie in one theater.
The number of movies showing in a theater is equal to the number of screens.
A review is of a specific movie.



This is one of several correct ER diagrams. Most variations are different ways of trying to capture the constraints given in the prose description that are not entity key constraints. Of those four constraints, only the constraint "A review is of a specific movie." is straightforwardly captured as a key constraint on "review body" in relationship "review". The other three constraints are not easily captured. The solution here captures "There is only one distributor for any one movie in one theater." by using aggregation to relate a "movie showing in a theater" to a unique distributor with a key and participation constraint from the movie-theater pair. However, the aggregation results in a consistency constraint that cannot be captured: the distributor related to the "movie showing in theater" pair through the *from* relationship must also be related to that movie in the *distributes* relationship. Constraints "Numbers of screens ≥ 1" and "The number of movies showing in a theater is equal to the number of screens." cannot be captured (they are constraints relating values of entities), but these constraints do imply the total participation constraint of *theater* in *showing.* The total participation of "review body" in "review" is not a constraint stated in the prose description but rather comes from the ER model design: the "review body" entity has been created specifically to hold information about published reviews.

Problem 4:
movie: **(** name , producer, rel_date, rating**)**

theater: **(**name, loc, #_screens, manager**)**

distributor: **(**name, proprietor, addr, tele**)**

review: **(**reviewer, publisher, text, date, name, produce, rel_date**)**
name **not null**
producer **not null**
rel_date **not null**
foreign key (name, producer, rel_date**) references** *movie*

distributes: **(**name, producer, rel_date, distrib_name**)**
foreign key (name, producer, rel_date**) references** *movie*
foreign key (distrib_name**) references** *distributor*

showing: **(**name, producer, rel_date, t_name, t_loc, distrib_name, start, end**)**
distrib_name **not null**
foreign key (name, producer, rel_date, distrib_name**) references** *distributes*
foreign key (t_name, t_loc**) references** *theater*

The entity key constraints are all directly captured as primary keys of the corresponding relations.
The key constraint on "review body" in relationship "review" allows the entity and relationship to be folded into one relation, *review*. The total participation of "review

body" in "review" is then represented by the "not null" constraints on the attributes referencing relation *movie*.

Because there is a key constraint on the aggregation of "showing" in the relationship "from", the "from" relationship can be folded into the relationship "showing", yeilding one relation *showing*. The total participation of the "showing" aggregation in relationshiop "from" is represented by the "not null" constraint on attribute "distrib_name" in relation *showing.* The total participation of "theater" in "showing" is not represented. Note that we achieve consistency between *distribu*tes and *showing* by making (name, producer, rel_date**,** distrib_name) a foreign key referencing *distributes*. This is something we could not do in the ER model. The two constraints relating values of entites, which we could not represent in the ER model, cannot be represented in the relational model either. However, we will see extensions in SQL to the model that do allow these constraints to be expressed.