

Tutorial 8 – ERD Conversion

WITH SUGGESTED SOLUTIONS

Answers to Review Questions

1. Explain the following terms:

- a. *Mapping (ER models into relations)*: transformation of entities, attributes, and relationships into corresponding tables, columns, and foreign keys.
- b. *Identifying relationship*: relationship between parent and dependent entity that becomes a part of the identifier for the dependent entity.
- c. *Binary relationship*: relationship that involves two entities.
- d. *Unary relationship*: relationship that involves a single entity.

2. Discuss the following:

- a. *Differences between identifiers and keys*: identifiers are used in the context of ERDs as identifiers for entities; keys are used in the context of the relational model i.e. primary and foreign keys.
- b. *Mapping 1:N relationships*: transforms into two relations, one of the relations (at the *many* end of the relationship) takes the foreign key.
- c. *Mapping M:N relationships*: transforms into three relations, one relation for each entity plus an intersection relation that contains a composite primary key formed by the primary keys of the associated relations.
- d. *Mapping of weak entities*: transforms into two relations; one for the parent entity and one for the weak entity.
- e. *How is the primary key of the relation that corresponds to a weak entity formed?* The primary key is a composite key consisting of the parent's primary key and the identifier of the weak entity.
- f. *Mapping of associative entities*: transforms into three relations – identical the transformation as for M:N relationship.
- g. *Mapping of subtypes*: three valid options exist: 1) form a single relation for the supertype and all subtypes, 2) form separate relations for each subtype, 3) form separate relations for each subtype and the supertype.

Solutions to Problems and Exercises

1. Convert the ERD for the New Oriental Hospital case study using the suggested solution given for tutorial 5 into a corresponding set of relations, indicating primary and foreign keys. Use an underline to indicate primary key attributes and asterisk (*) to indicate foreign keys, as shown below:

employee(empno, ename, sal, deptno*)
FK (deptno) references dept

Relations:

DRUG(drugNo, drugDesc, drugDosage, drugMethod, drugName, drugUnitPrice)

STAFF (staffID, staffName)

PATIENT (patID, patFname, patLname, patDOB, patGender, patAddress, patPhone, patRegDate, CompID*)
FK (CompID) references COMPANY

PATIENTCHART (patCID, patID*, patAdmDate, patDischDate, patSymp, patDiag, staffIDDoc*, staffIDNurse*, wardNo)
FK (staffIDDoc) references STAFF
FK (staffIDNurse) references STAFF
FK (patID) references PATIENT

PRESCRIBEDDRUG (drugNo*, (patCID, patID)*, drugAmtPDay, Sdate, Edate)
FK (DrugNo) references DRUG
FK (patCID, patID) references PATIENTCHART

COMPANY (compID, compName, compAddress, compPh, compRep)

2. Now convert the ERD for the extended NOH case study from tutorial 6. Comment on any design decisions that you made.

Relations:

DRUG(drugNo, drugDesc, drugDosage, drugMethod, drugName, drugUnitPrice)

STAFF (staffID, staFName, staLName, staAddress, staGender, staPhone, staDOB, staSScale, staCSalary, staJType, staSDate, DocPager, DocSpecialty, NursePosition, StaffType)

WARD(WardNo, WardName, WardLoc, WardExt, WardCap)

NURSEWARD(staIDNurse*, WardNo*, Startdate, Shift, EndDate)

FK (staIDNurse) references STAFF

FK (WardNo) references WARD

COMPANY (compID, compName, compAddress, compPh, compRep)

PATIENT (patID, patFname, patLname, patDOB, patAge, patAddress, patPhone, patRegDate, CompID*)

FK (CompID) references COMPANY

PATIENTCHART (patCID, patID*, patAdmDate, patDischDate, patSymp, patDiag, staffIDDoc*, staffIDNurse*, wardNo*)

FK (staffIDDoc) references STAFF

FK (staffIDNurse) references STAFF

FK (patID) references PATIENT

FK (wardNo) references WARD

PRESCRIBEDDRUG (drugNo*, (patCID, patID)*, drugAmtPDay, Sdate, Edate)

FK (DrugNo) references DRUG

FK (patCID, patID) references PATMCHART

MCHARGES (miscNo, miscDesc, MiscPrice)

MISCITEMUSED(miscNo*, (patCID, patID)*, miscAmt)

FK (patCID, patID) references PATMCHART

FK (miscNo) references MISCELLANEOUS

BILL(BillNo, BillDate, (patCID, patID)*)

FK(patCID, patID) references PATMCHART

Note: STAFF, DOCTOR, and NURSE transformed into a single relation STAFF that includes the subtype attributes: DocPager, DocSpecialty, NursePosition. This solution has the benefit of being simple (i.e. only one relation), but the subtype attributes will contain null values (e.g. in records that represent doctors the POSITION attribute will be null). Please note that we need StaffType attribute in this relation to differentiate between DOCTOR and NURSE (though this attribute is not shown in an entity STAFF). Alternative solutions exist:

- a) Two separate relations:

DOCTOR(staffID, staFName, staLName, staAddress, staGender, staPhone, staDOB, staSScale, staCSalary, staJType, staSDate, DocPager, DocSpecialty)

NURSE(staffID, staFName, staLName, staAddress, staGender, staPhone, staDOB, staSScale, staCSalary, staJType, staSDate, NursePosition)

Advantage: avoids null values, but two relations.

Should only be used when the subtypes are Disjoint (D). If we use this approach with overlapping subtypes, we will be storing redundant data for one instance of, for example in this case, staff member – their personal details will be stored in more than one relation/table.

b) Three separate relations:

STAFF(staffID, staFName, staLName, staAddress, staGender, staPhone, staDOB, staSScale, staCSalary, staJType, staSDate)

DOCTOR(staffID*, DocPager, DocSpecialty)

FK (staffID) references STAFF

NURSE(staffID*, NursePosition)

FK (staffID) references STAFF

This solution is used in situations where the subtypes are overlapping to avoid redundancy (does not apply to this example, as DOCTOR and NURSE are disjoint subsets)