Tutorial 10 – Bottom Up Normalisation

WITH SUGGESTED SOLUTIONS

Review Questions

Discuss the following:

a. *Why redundancy should be avoided when designing a database : so that update anomalies are avoided*

b. *Are there situations where data redundancy is acceptable? Yes, if the data is never updated, update anomalies cannot arise, e.g. historical data, audit trails, etc.*

c. *Define and explain 1NF: all attributes must be atomic*

d. *Define and explain 2NF: 1NF + no partial dependencies on relational key(s)*

*(**Note**: relation key(s) is any key that uniquely identify the row e.g. primary key, candidate key)*

e. *Define and explain 3NF: 2NF + no inter-dependencies between non-key attributes*

f. *Define and explain BCNF: all determinants must be relation keys*

g. *How can BCNF be violated in relations that are in 3NF: BCNF can be violated if the relation has overlapping keys*

h. *How can keys be derived from functional dependencies: form a set of all determinants to get a superkey and reduce this set by substitution (i.e. using FDs) until irreducible subsets are produced – these are relation keys*

Solutions to Problems and Exercises

Bottom-up Design

1. Use the list of all the functional dependencies (FDs) from tutorial9 solution, perform the following tasks:

i) create a relation that includes all the attributes in the form
ii) decompose this relation to ensure BCNF
iii) justify your answers

Once you have done this for all of the forms, check whether you have any relations that have exactly the same set of attributes, decide what relation name is most suitable for that set, and use only one relation for it.

2. Put together your final set of fully normalised relations (BCNF).

Solution

Functional dependencies (for first part of case study)

- *PatID → patFname, patLname, patAge, patGender, patDOB, patAddress, patPhone, patRegDate, CompID, compname, compAddress, compPhone, compRep*
- *CompID → compname, compAddress, compPhone, compRep*
- *PatCID, PatID → wardNo, patAdmDate, patDisDate, patSymp, patDiag, staIdDoc, staIdNurse*
- *DrugNo → DrugName, DrugDesc, DrugDosg, DrugMethod, DrugPrice*
- *PatCID, PatID, DrugNo → AmtPerDay, Sdate, Edate*
- *staffID → staffName*

Functional Dependencies (for the second part of case study)

- *StaffID → First Name, Last Name, Address, Gender, Phone, DOB, Salary Scale, Job Type, Current Salary, Start Date, Pager, Position, Speciality*
- *Ward No → Ward Capacity, Ward Name, Ward Extension, Ward Location*
- *Ward No, StaffID, Start_Date → Shift, End_Date*
- *BillNo → Date, PatientChartID, PatientID*
- *PatientID → InsuranceCompanyID*
- *PatientChartID, PatientID → BillNO*
- *PatientChartID, PatientID, DrugNo → DrugAmount*
- *BillNo, DrugNo → DrugAmount*
- *MiscNo → Description, Price*
- *DrugNo → DrugName, Price*
- *PatientChartID, PatientID, MiscNo → MiscAmount*
- *BillNo, MiscNo → MiscAmount*

PATIENT REGISTRATION FORM (Figure 1)

PATIENT_REGISTRATION (patID, patFname, patLname, patDOB, patAge, patGender, patAddress, patPhone, patRegDate, compID, compName, compAddress, compPh, compRep)

Key: patID
transitive dependence e.g. patID -> compID->compName, etc. - 3NF violation
Decompose into:

PATIENT (<u>patID</u>, patFname, patLname, patDOB, patAge, patGender, patAddress, patPhone, patRegDate, CompID)

Now: patAge is a derived attribute (from patDOB) and should be removed

Relation PATIENT is in BCNF – patID is a determinant and a relation key
Alternatively:
- no partial dependencies of relational key
- no inter-dependencies between non-prime attributes

COMPANY (<u>compID</u>, compName, compAddress, compPh, compRep)

Relation COMPANY is in BCNF – compID is a determinant and a relation key

Alternatively:
- no partial dependencies of relational key
- no inter-dependencies between non-prime attributes

PATIENT MEDICAL CHART FORM (Figure 2)

PATIENT_CHART (patCID, patID, patAdmDate, patDischDate, patSymp, patDiag, staffIDDoc, staffIDNurse, wardNo, drugNo, drugAmtPDay, Sdate, Edate, drugDesc, drugDosage, drugMethod, drugName, drugUnitPrice)

Key: patID, patCID, drugNo

Partial dependence on the key, e.g. patCID, patID -> wardNo, etc. - 2NF violation

Decompose into:

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

Key: patID, patCID

Relation PATIENTCHART is in BCNF – patCID, patID is a determinant and a relation key
Alternatively:
- no partial dependencies of relational key
- no inter-dependencies between non-prime attributes

DRUG_PRESCRIPTION (patCID, patID, drugNo, drugAmtPDay, Sdate, Edate, drugDesc, drugDosage, drugMethod, drugName, drugUnitPrice)

Key: patID, patCID, drugNo

Partial dependence on the key, e.g. drugNo -> drugName, etc. - 2NF violation
Decompose into:

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

Key: drugNo

Relation DRUG is in BCNF – drugNo is a determinant and a relation key
Alternatively:
- no partial dependencies of relational key
- no inter-dependencies between non-prime attributes

PRESCRIPTION (drugNo, patCID, patID, drugAmtPDay, Sdate, Edate)

Relation PRESCRIPTION is in BCNF – drugNo, patCID, patID is determinant and a relation key
Alternatively:
- no partial dependencies of relational key
- no inter-dependencies between non-prime attributes

STAFF FORM (Tutorial 6 – Figure 3)

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

Key: staffID

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

Relation STAFF is in BCNF – staffID is a determinant and a relation key
Alternatively:
- no partial dependencies of relational key
- no inter-dependencies between non-prime attributes

WARD ALLOCATION FORM (Tutorial 6 – Figure 4)

WARD_ALLOCATION (WardNo, WardName, WardLoc, WardExt, WardCap, staIDNurse, Startdate, Shift)

Key: WardNo, staIDNurse, Startdate

Partial dependence on the key, e.g. WardNo -> WardName, etc. - 2NF violation

Decompose into:

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

Key: WardNo

Relation WARD is in BCNF – WardNo is a determinant and a relation key
Alternatively:
- no partial dependencies of relational key
- no inter-dependencies between non-prime attributes

NURSEWARD (staIDNurse,WardNo, Startdate, Shift)

Key: WardNo, staIDNurse, Startdate

Relation NURSEWARD is in BCNF – staIDNurse, WardNo, Startdate is a determinant and a relation key
Alternatively:
- no partial dependencies of relational key
- no inter-dependencies between non-prime attributes

PATIENT MEDICAL BILL FORM (Tutorial 6 – Figure 5)

BILL (BillNo, BillDate, patCID, patID, CompID, drugNo, DrugName, DrugAmt, DrugPrice, miscNo, miscDesc, MiscPrice, miscAmt)

Key: BillNo, drugNo, miscNo

Partial dependence on the key, e.g. BillNo -> BillDate, etc. - 2NF violation
Decompose into:

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

Key: BillNo

Transitive dependency: BillNo -> patID -> compID; 3NF violation

Decompose into:
BILL (BillNo, BillDate, patCID, patID)
PATIENT2 (patID, compID)

Relation BILL is in BCNF – BillNo is a determinant and a relation key
Alternatively:
- no partial dependencies of relational key
- no inter-dependencies between non-prime attributes

Relation PATIENT2 is in BCNF – patID is a determinant and a relation key
Alternatively:
- no partial dependencies of relational key
- no inter-dependencies between non-prime attributes

Partial dependence on the key, e.g. drugNo -> DrugName, etc. - 2NF violation
Decompose into:

DRUG2 (drugNo, DrugName, DrugAmt, DrugPrice)

Relation DRUG2 is in BCNF – drugNo is a determinant and a relation key
Alternatively:
- no partial dependencies of relational key
- no inter-dependencies between non-prime attributes

Partial dependence on the key, e.g. miscNo -> miscDesc, etc. - 2NF violation
Decompose into:

MISCELLANEOUS (miscNo, miscDesc, MiscPrice)

Relation MISCELLANEOUS is in BCNF – miscNo is a determinant and a relation key
Alternatively:
- no partial dependencies of relational key
- no inter-dependencies between non-prime attributes

CHARGES(miscNo, patCID, patID, miscAmt)

Relation CHARGES is in BCNF – miscNo, patCID, patID is a determinant and a relation key
Alternatively:
- no partial dependencies of relational key
- no inter-dependencies between non-prime attributes

Resulting set of relations:

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

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

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

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

PRESCRIPTION (drugNo, patCID, patID, drugAmtPDay, Sdate, Edate)

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)

BILL (BillNo, BillDate, patCID, patID)

PATIENT2 (patID, compID)

DRUG2 (drugNo, DrugName, DrugAmt, DrugPrice)

MISCELLANEOUS (miscNo, miscDesc, MiscPrice)

CHARGES(miscNo, patCID, patID, miscAmt)

Finally, combine relations with identical primary keys and remove derived attributes:

i) combine PATIENT and PATIENT2
ii) combine DRUG and DRUG2, assume that DrugAmt, DrugPrice are derived from drugAmtPDay, drugUnitPrice, and the number of days, i.e, Edate – Sdate

Final set of BCNF relations: (note, this set should be identical to relations derived from top-down design in tutorial 9)

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

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

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

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

PRESCRIPTION (drugNo, patCID, patID, drugAmtPDay, Sdate, Edate)

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)

BILL (BillNo, BillDate, patCID, patID)

MISCELLANEOUS (miscNo, miscDesc, MiscPrice)

CHARGES (miscNo, patCID, patID, miscAmt)

3. Analyse and compare results for question 1 from this tutorial and those given for the normalisation done in tutorial 9. Are they the same or different sets of relations? Why do you think they are different, or the same? What changes would you make to your database design (final set of relations) as a result of this comparison?

Most relations from tutorial 9 (Top down normalisation) are the same with the relations here (bottom up normalisation). The main difference is from Top Down normalisation, there are separate relations for doctor and nurse. The separation of doctor and nurse relations is better database design because we do not need to have null value for the attributes which belong to only doctor or nurse. For example, only staff members who are doctors will have values stored for attributes Pager and Specialty; nurses will thus not have null values for these attributes because they are not attributes in the Nurse relation, only in the doctor relation.
Further, the bottom-up approach results in an extra attribute StaffType in the Staff relation which is not necessary when separate relations are created for Doctor and Nurse.

Extra notes regarding bottom-up normalization:

The approach that you should follow to reach BCNF in the given scenario can be described as following steps:

1) Explore the functional dependencies: all functional dependencies should be recognized and depicted as follow:
   Determinant 1 -> Attribute11, Attribute 12, …, Attribute 1n
   .
   .
   Determinant m-> Attribute m1, Attribute m2,…,Attribute mn

   where each determinant can be simple (one attribute) or can be composite (composed of couple attributes), but must be unique.

2) Create an initial relation: This relation should include all existing attributes and have a candidate key. The candidate key can be simple (one attribute) or composite (composed of number determinants). However, if the candidate key is composite, it must be minimal. The candidate key is a minimal super key. It means that we cannot remove any part of the composite candidate key without losing its uniqueness. Let assume we have composite key composed of attributes A: Student ID, B: Subject ID, C: Date of enrolment with following records in enrolment table:
   Key: Student Id, Subject ID, Date of Enrolment

Student ID	Subject ID	Date of Enrolment	Fee of enrolment
123	D001	02/08/2012	3000
123	D002	03/08/2012	3500
126	D001	02/08/2012	3300
125	D002	03/08/2012	3400

   In this case the composite key is super key and is not candidate key, because it is not minimal and we can remove the Date of Enrolment without losing the uniqueness. So the candidate key should be: Key: Student Id, Subject ID
   Now if we add one more record to table like follow (Same student with id 123 enroll to same subject with id D001 in different time) :
   Key: Student Id, Subject ID, Date of Enrolment

Student ID	Subject ID	Date of Enrolment	Fee of enrolment
123	D001	02/08/2012	3000
123	D002	03/08/2012	3500
126	D001	02/08/2012	3300
125	D002	03/08/2012	3400
123	D002	02/03/2013	3300

   In this case the composite key is candidate key, because it is minimal and we cannot remove Date of Enrollment without losing its uniqueness. If we remove the Date of Enrolment the remaining (Student Id, Subject ID) is not unique any more.

3) Check the 1 NF: there is not non-atomic and derived attribute.

4) Check the 2 NF: There is not partial FD. It happens when the relation has the composite candidate key: R (D1, D2, D3, A1, A2, A3, A4). Let assume A2 and A3 are attributes that are only dependent on D2. D2 -> A2, A3
   We should split the relation into two relations like:
   R1 (D1, D2, D3, A1, A4): Copy the composite key and the attributes A1 and A4. (Relation R – A1 and A2)
   R2 (D2, A2, A3) : Copy the key D2 and the attributes A2 and A3.

   You have to check the 2NF for these relations again. If there is any partial FD then repeat the step 4.

5) Check the 3NF: There is no transitive FD. It happens when one non-key attribute is dependent on another non-key attribute. Given R (D1, A1, A2, A3), and the attribute A3 is dependent on attribute A1: D1 -> A1 -> A3.
   We should split the relation into two relations like:
   R1 (D1, A1, A2): Copy the key and the attributes A1 and A2 (Relation R – A3)
   R2 (A1, A3) : Copy the attribute A1 and make it key and add attribute A3.

   You have to check the 3NF for these relations again. If there is any transitive FD then repeat the step 5.