Logical Database Design and the Relational Model - III

Lecture 13 - 14

CSE 303: Database Management System

Further Normalization

- Merging Relations
  - On large projects, the work of several subteams comes together during logical design, so there is often a need to merge relations.
  - Integrating existing databases with new information requirements often leads to the need to integrate different views.
  - New data requirements may arise during the life cycle, so there is a need to merge any new relations with what has already been developed.

Example

EMPLOYEE1(EmployeeID, Name, Address, Phone)

EMPLOYEE2(EmployeeID, Name, Address, Jobcode, NoYears)

EMPLOYEE(EmployeeID, Name, Address, Phone, Jobcode, NoYears)

Problems

- Synonym: Two (or more) attributes that have different names but the same meaning.
- Homonym: An attribute that may have more than one meaning.
- Transitive dependencies
- Super-type sub-type relations

Other Normal Forms

- Boyce-Codd NF
  - All determinants are candidate keys...there is no determinant that is not a unique identifier
- 4^{th} NF
  - No multivalued dependencies
- 5^{th} NF
  - No "lossless joins"
- Domain-key NF
  - The "ultimate" NF...perfect elimination of all possible anomalies

Review Normalization

- Normalization of data can be viewed as a process during which a 'badly structured' relation schema can be decomposed by breaking up the attributes into smaller schemas.
- E.F. Codd had originally proposed the first three normal forms. Since then, ideas of other normal forms were developed.
- Normalization is primarily aimed to reduce data redundancy. It also invokes taking complex record and converting it into a series of simpler records with the suitable relationships and with no loss of information.

Why Normalization?

- Repetition of information.
- Inability to represent certain information.
- Loss of information.

Boyce-Codd Normal Form (BCNF)

- When a relation has more than one candidate key, anomalies may result even though the relation is in 3NF.
- 3NF does not deal satisfactorily with the case of a relation with overlapping candidate keys
  - i.e. composite candidate keys with at least one attribute in common.
- BCNF is based on the concept of a *determinant*.
  - A determinant is any attribute (simple or composite) on which some other attribute is fully functionally dependent.
- A relation is in BCNF if, and only if, every determinant is a candidate key.

Example 1

Patient No	Patient Name	Appointment Id	Time	Doctor
1	John	0	09:00	Zorro
2	Kerr	0	09:00	Killer
3	Adam	1	10:00	Zorro
4	Robert	0	13:00	Killer
5	Zane	1	14:00	Zorro

Two possible keys

- DB (Patno, PatName, appNo, time, doctor)
- Determinants:
  - Patno → PatName
  - Patno, appNo → Time,doctor
  - Time → appNo

- Two options for 1NF primary key selection:
  - DB(<u>Patno,</u>PatName,<u>appNo</u>,time,doctor) (example 1a)
  - DB(<u>Patno,</u>PatName,appNo,<u>time</u>,doctor) (example 1b)

Example 1a

- DB(Patno,PatName,appNo,time,doctor)
- No repeating groups, so in 1NF
- 2NF – eliminate partial key dependencies:
  - DB(Patno,appNo,time,doctor)
  - R1(Patno,PatName)
- 3NF – no transitive dependencies so in 3NF
- Now try BCNF.

BCNF Every determinant is a candidate key

DB(Patno,appNo,time,doctor)

R1(Patno,PatName)

- Is determinant a candidate key?
  - Patno → PatName
    Patno is present in DB, but not PatName, so irrelevant.

Continued...

DB(Patno,appNo,time,doctor)

R1(Patno,PatName)

- Patno,appNo → Time,doctor
  All LHS and RHS present so relevant. Is this a candidate key? Patno,appNo IS the key, so this is a candidate key.
- Time → appNo
  Time is present, and so is appNo, so relevant. Is this a candidate key? If it was then we could rewrite DB as:
  DB(Patno,appNo,time,doctor)
  This will not work, so not BCNF.

Rewrite to BCNF

- DB(Patno,appNo,time,doctor)

  R1(Patno,PatName)

- BCNF: rewrite to

  DB(Patno,time,doctor)

  R1(Patno,PatName)

  R2(time,appNo)

- time is enough to work out the appointment number of a patient. Now BCNF is satisfied, and the final relations shown are in BCNF.

Example 1b

- DB(Patno,PatName,appNo,time,doctor)
- No repeating groups, so in 1NF
- 2NF – eliminate partial key dependencies:
  - DB(Patno,time,doctor)
  - R1(Patno,PatName)
  - R2(time,appNo)
- 3NF – no transient dependencies so in 3NF
- Now try BCNF.

BCNF Every determinant is a candidate key

DB(Patno,time,doctor)
R1(Patno,PatName)
R2(time,appNo)

- Is determinant a candidate key?
  - Patno → PatName
    Patno is present in DB, but not PatName, irrelevant.
  - Patno,appNo → Time,doctor
    Not all LHS present so not relevant
  - Time → appNo
    Time is present, but not appNo, so not relevant.
  - Relations are in BCNF.

Thank You