Normal Forms

Source/References:
Database Systems: The Complete Book, and Elmasir/Navathe

pm jat @ daiict

Normal Forms

- Normal forms are used as measure of “goodness” of a relation.

- Higher the normal form, less redundancies it has, and less the anomalies it has; and better the relation is.

The process of having “good relations” in a relational database schema, i.e. they are free from anomalies and redundancies is called as **Normalization**

Normal forms

- Initially Edgar(Ted) F. Codd proposed three normal forms, which he called First, Second, and Third normal forms.
- A stronger definition of 3NF, called Boyce-Codd Norm Form (BCNF) – was proposed later by Boyce and Codd.
- All these normal forms are based on the functional dependencies among the attributes of a relation.
- Later a Fourth normal form (4NF) and Fifth normal form (5NF) were proposed based on multi-value dependencies and join dependencies respectively

Normal Forms

- First Normal Form (1NF)
- Second Normal Form (2NF)
- Third Normal Form (3NF)
- Boyce - Codd Normal Form (BCNF)
- Fourth Normal Form (4NF)
- Let us not go higher

First Normal Form (1NF)

- A relation is in First normal form if it qualifies to following
  - A relation is set of tuples, i.e. every tuple is distinct
  - Each value in a tuple is “atomic”

Example-First Normal Form (1NF)

- With given definition, following relation are not in 1NF-

  Room_HOR(RoomNo, Wing, Floor, Resident_Set),
  and suppose it has following two tuples-
  <C115, C, 1, {201001023, 201001111}>
  <H211, H, 2, {201111011}>

Not in 1NF, because attribute is not atomic

- Alternative following scheme is also not in 1NF-
  Room_HOR(RoomNo, Wing, Floor, Resident1, Resident2),
  and suppose it has following two tuples-
  <C115, C, 1, 201001023, 201001111>
  <H211, H, 2, 201111011, null>

Because- not correct way of dealing with multiple values of an attribute

Example-First Normal Form (1NF)

- Following are more relation schemes that are not in 1NF-

  Department(DNO, DName, MGRSSN, DLOCATIONS)
  <4,'Marketing',101, {Delhi,Mumbai,Pune}>

- Composite attributes are also not atomic therefore following relation is also not in 1NF-
  Student(ID, Name(Fname, Minit, Lname), Batch, CPI)
  <200701001,<Charu,K,Chawla>, 2007, 6.8>

First Normal Form – issue?

- Is not DOB is a composite in following relation scheme?
- Student(ID, FName, MInit, LName, DOB, Batch, CPI), <200701001, Charu, K, Chawla, <1988,11,21>, 2007, 6.8>
- is still the relation is 1NF?

First Normal Form (1NF)

- CJ Date given modern understanding of atomicity as single value of "appropriate type".

- And type could be anything – scalar, complex, arrays, user defined types, and even relations, and that is what modern DBMS supports – ORDBMS!

- Does not it messes up with all learning about atomicity?

Normal Forms

- First Normal Form (1NF)
- Second Normal Form (2NF)
- Third Normal Form (3NF)
- Boyce - Codd Normal Form (BCNF)
- Fourth Normal Form (4NF)
- Let us not go higher

Two ways of understanding 2nd, 3rd, BC Normal Forms

- Begin with BCNF (more stricter form), and come down to 2NF, or
- Reverse, i.e. start with 2NF, and go upto BCNF
- These days First approach is preferred with following reasons:
  - Most relations that you design using your common sense, or by mapping ER to Relations; you are likely to have BCNF
  - Second, 2NF and 3NF are discussed more for historical reasons

2NF and 3NF

- Term: Prime Attribute. Is the one that is part of any of (minimal) key.

- **2NF**: A relation is 2NF is all non-prime attributes are functionally (irreducibly) dependent on key.

- **3NF**: A relation is 3NF is all non-prime attributes are functionally (irreducibly) directly, and not transitively dependent on key.

2NF and 3NF - examples

- Following relations are not in 2NF
  - R1(SSN, PNO, PNAME, HOURS)
  - R2(AcadYear, Semester, CourseNo, Credit, FacultyID)
  - R3(AcadYr, Sem, CourseNo, Credit, StudentID, Grade)

- Following relations are in 2NF (but not in 3NF)
  - EMP(SSN, FName, DOB, DNO, Salary, Dname, MGRSSN)
  - Team(TeamID, PWD, MentorID, InstituteID)

BCNF

- A relation R is in Boyce-Codd Normal Form, when determinant of every FD that holds on R, is key of R.
- In other words, For every FD **A** → **B** that holds on relation R, A is its key.
- This is to be checked for all FDs on R, if any FDs fail to meet this criteria, the relation is not in BCNF

Example BCNF

- Check if these relations are in BCNF

employee(ssn, fname, salary, DoB, superssn, dno)

department(dno, dname, mgrssn, mgrstartdate)

dept_locations(dno, dlocation)

project(pno, pname, plocation, dno)

works_on(essn, pno, hours)

dependent(essn, dep_name, dep_bdate, relationship)

ssn → fname
ssn → minit
ssn → lname
ssn → salary
ssn → superssn
ssn → bdate
ssn → gender
ssn → dno
dno → dname
dno → mgrssn
dno → mgrstartdate
pno → pname
pno → proj_dno
pno → plocation
{ssn, pno} → hours
{ssn, dep_name} → dep_gender
{ssn, dep_name} → dep_bdate
{ssn, dep_name} → relationship

Are relations in BCNF?

Member(MembID, MembName, MembEmail, TeamID)

Team(TeamID, TeamPWD, MentorID)

Mentor(MentorID, MentorName, Email, InstID)

Institute(InstID, InstName, City, PIN, State)

FDs:

MembID -> MembName, MembEmail, TeamID, TeamPWD, MentorID, MentorName, Email, InstID, InstName, City, PIN, State

TeamID -> TeamPWD, MentorID, MentorName, Email, InstID, InstName, City, PIN, State

MentorID -> MentorName, Email, InstID

InstID -> InstName, City, PIN, State

PIN -> City, State

Another Definition of 3NF

- 3NF is less restrictive than BCNF, it relaxes BCNF condition for *prime attributes* (attribute that are part of some candidate key)

- A relation is in 3NF, if, for every FD **A → B** that holds on relation R,
  - A is its key, or
  - B is a prime attribute.

BCNF and 3NF - exercise

SSN	FNAME	PNO	PNAME	HOURS
101	Sumit	1	P-1	38
101	Sumit	2	P-2	20
102	Vipul	1	P-1	64
103	Ajay	2	P-2	58

- FDs (Minimal Set):
  - **{SSN, PNO} → HOURS**
  - **SSN → FNAME**
  - **PNO → PNAME**
- What are keys? What NF the relation is?

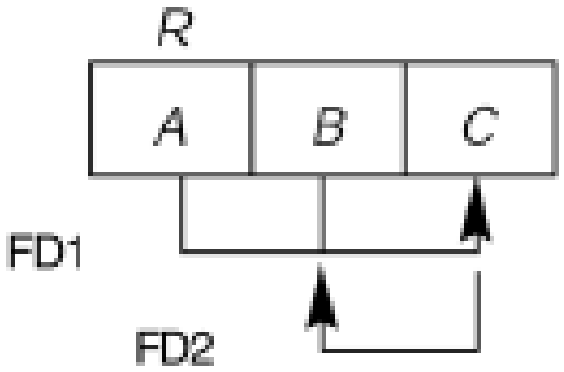
BCNF and 3NF

- Suppose we have WORKS_ON as following:
  WORKS_ON(ESSN, PNo, PName*, Hours)
- FDs (suppose):
  PNO → Pname
  PName → PNo //Assume that Pname is Unique
  {ESSN, PNo} → Hours
  {ESSN, PName} → Hours
  Keys: {ESSN, PNO}, and {ESSN, PName}

- This is not in BCNF, last two FDs are violating BCNF requirement!

BCNF and 3NF

- Below is typical situation where a relation is in 3NF but not in BCNF.
- A relation R(A,B,C} having following FDs: {A,B} → C, C→A; and {A,B} as key.



BCNF and 3NF - Exercises

- Is the relation in BCNF <Yes/No>?
- Is the relation in 3NF <Yes/No>? Consider both definitions of 3NF!

  EMP_DEP(ssn, fname, salary, superssn, dno, dname, mgrssn, mgrstartdate)

FDs in **EMP_DEP**:

ssn -> fname, salary, superssn, dno, dname, mgrssn, mgrstartdate

dno -> dname, mgrssn, mgrstartdate

BCNF and 3NF - Exercises

- Is the relation in 2NF <Yes/No>?

2NF Definition (A relation is 2NF is all non-prime attributes are functionally (irreducibly) dependent on key).

EMP_DEP(ssn, fname, salary, superssn, dno, dname, mgrssn, mgrstartdate)

FDs in **EMP_DEP**:

ssn -> fname, salary, superssn, dno, dname, mgrssn, mgrstartdate

dno -> dname, mgrssn, mgrstartdate

Other definition in 2NF

- In 2NF, we permit FD X→B; when we also have a FD A→X, and A is key:
  - That means we have FD A → B transitively inferred from FDs A → X and X → B)
  - B is still determined by A (key)
- Example: dno --> mgrssn is acceptable FD in 2NF, because you also have FD ssn → dno and ssn is Key

Normal Forms - Summarized

- BCNF: For every FD X → Y that holds on a relation, X is key.
- 3NF: Relaxes BCNF. For every FD X → Y that holds on a relation, one of following is true –
  - X is key, OR
  - Y is prime attribute
- 2NF: for every FD X → Y, one of following is true –
  - X is key, OR
  - Y is prime attribute, OR
  - There is another FD A → X, where A is the key; i.e. Y is transitively determined by key.
- 1NF: attributes have “atomic” values

Normal Forms- alternate definitions

- **2NF**: all non-prime attributes are irreducibly, or fully dependent on key. (should not partially dependent on key)
- **3NF**: all non-prime attributes are irreducibly and directly dependent on key, transitively dependency is not allowed.
- Note that both definitions of 2NF and 3NF(i.e. earlier and this) mean same only.

Sources/References

- Database Systems: The Complete Book, by
  Hector G. Molina, Jeff Ullman, and Jennifer Widom
  Pearson Education, India
  Home Page:
  http://infolab.stanford.edu/~ullman/dscb.html

- Elmasir/Navathe