CS422: Database Management Systems

Lesson 5 - Normalization

Normalization Exercises

1) A relation R of journal holdings has a schema *R(Title, Keywords, Holdings)*

   A few tuples might look like:

Title	Keywords	Holdings
SIGSAM Algorithms	Computing, algorithms	Dec 1990 to Feb 1995
SIGSAM Algorithms	Computing, algorithmics	Jan 1998 to present
The Network Expert	Computer Science, networks	Sept 1999 to present

   a) **What normal forms are violated?**
   ANS:
   1NF: Keywords is a repeating field, Holdings contains two separable data items
   2NF: The only candidate key is (Title, Holdings), but there is a functional dependency *(Title → Keywords)*. So we have a partial dependency of Keywords on the primary key, (Title, Holdings).

   b) **What problems does this cause?**
   ANS:
   1NF: There is no control over keywords, so the same intention may be represented by numerous variations. For example, algorithms or algorithmics, computing or computer science.
   2NF: If we want to add a keyword for a journal title, we may have to add it on more than one row.

   c) **Propose a normalized version of the relation.**
   ANS:
   1NF: Repeat tuples so that each tuples has one keyword. Split the attribute Holdings into two attributes, DateFrom and DateTo.

   2NF: Split R into
   R1(<u>Title</u>, <u>DateFrom</u>, DateTo)
   R2(<u>Title</u>, <u>Keyword</u>)

   Repeating the title of a journal in many tuples is potentially a waste of space and might be trying to update. There is a question of policy here: if a journal changes its title, does one keep the backissues with their original title? The answer is generally yes. So we do not need to worry about updates, though we should note that the current design fails to retrieve all holdings of a journal which has changed its name.

We could avoid the repeated title and save space by using:
R1(TitleID, DateFrom, DateTo)
R2(TitleID, Keyword)
R3(TitleID, Title)
It is possible, however, that the cost in time spent in doing the join for almost every query might outweigh the savings in space.

We could also eliminate the confusion of keywords by introducing a Keyword relation:
R1(TitleID, DateFrom, DateTo)
R2(TitleID, KeywordID)
R3(TitleID, Title)
R4(KeywordID, Keyword)

---

2) **Given the relational schema:**
Manager(BranchID, BranchEID, Name, Phone, Secretary, SecretaryPhone, SalaryHistory, Branch, BranchDirector)

and the sample relation

BID	BEID	Name	Phone	Sec	SecPh	SalHist	Branch	BrDir
4	7	Jim	413	Ann	241	1999, $20000 2000, $30000	Bangor	Eric
5	7	John	502	Alice	679	2000, $35000	Biloxi	Earl
4	2	Jesse	658	Ann	241	2000, $40000	Bangor	Eric
4	2	Jesse	659	Ann	241	2000, $40000	Bangor	Eric

a) **Make a minimal change to put the relation into 1NF.**
ANS:
Manager(BID, BEID, Name, Phone, Sec, SecPh, Branch, BrDr)
SalaryHistory(BID, BEID, Year, Salary)

b) **Make a minimal change to put your answer to (a) into 2NF.**
ANS:
The only candidate key is *(BID, BEID, Phone)*. There are partial dependencies on *BID* and on *(BID, BEID)*:

Branch(BID, Name, Director)
Manager(BID, BEID, Name, Sec, SecPh)
Phone(BID, BEID, Phone)
SalaryHistory(BID, BEID, Year, Salary)

c) Make a minimal change to put your answer to (b) into 3NF.
ANS:
There is one transitive dependency, *(Sec→SecPh)*

Branch(BID, Name, Director)
Manager(BID, BEID, Name, SecID)
Secretary(SecID, Name, SecPh)
Phone(BID, BEID, PhNo)
SalaryHistory(BID, BEID, Year, Salary)

---

3) In a relation R(A,B,C,D), A is the primary key, BC is an alternate key, and there is a functional dependency D→C.

a) Show that this is possible by constructing an instance of R with these properties.
ANS:

A	B	C	D
1	a	7	x
2	a	8	y
3	b	7	x
4	c	7	z

b) Are there any other alternate keys?
ANS:
As the example shows, neither B, nor C, nor D is an alternate key.

AB, AC, and AD are not minimal, so they are not alternate keys. Since BC is an alternate key and (D→C), we see that BD is also an alternate key.
From the example, we find that CD is not an alternate key.

Each set of three attributes, BCD, ACD, ABD, and ABC, properly contain an alternate key and so are not themselves alternate keys. Similarly for ABCD.

So, BD is the only other alternate key.

c) Is there an insertion anomaly?
ANS:
Yes. If we insert a row (5, c, 9, x), then A and BC are still candidate keys, but (D→C) no longer holds.

d) Is there an update anomaly?
ANS:
Yes. If we update row (1, a, 7, x) to (1, a, 9, x), again DC will be violated.

e) **Is there a deletion anomaly?**
ANS:
Yes. If we delete row (3, a, 8, y), then the association (8, y) of C and D will be lost.

f) **Propose a normalization of R.**
ANS:
We must split R(A, B, C, D) into R1=(A,B,D) and R2=(C,D):

After normalization: R1

A	B	D
1	a	x
2	a	y
3	b	x
4	c	z

R2

C	D
8	y
7	x
7	z

g) **Is there an insertion anomaly?**
ANS:
Yes.The delicate point is the functional dependency FD(BC, D). Perhaps we can have two tuples with the same BC values but different D values? Looking at the first row of R, (1, a, 7, x), we aim at a tuple (5, a, 7, q). This tuple comes from inserting (5, a, q) into R1 and (7, q) into R2.

h) **Is there an update anomaly?**
ANS:
Yes. It is possible to update (8, y) in R2 to (7, y). Then when we create R from the join of R1 and R2, BC will no longer be a candidate key.

i) **Is there a deletion anomaly?**
ANS:
No. Deleting a row from R1 or R2 will not lead to unexpected data loss. There is the issue of whether to cascade the deletions from R2 to R1, but this does not count as an anomaly.

4) Suppose we have a relational schema R(A,B,C,D,E) for which the following functional dependencies hold:
A→C,E
B→D,
BC→E,
DE→B

a) Find all the candidate keys for R.
ANS:
Since A is not determined by any of the given FD's, any candidate key must contain A. Taking each pair as a determinant:
A → ACE
AB → ABCE→ ABCDE
AC → ACE
AD → ACDE → ABCDE
AE → ACE
So AB and AD are the only candidate keys.

There are no others. A candidate key cannot properly contain AB or AD. The only attribute set that fits these conditions is ACE, which does not determine B or E.

b) Show that R is not in 2NF.
ANS:
If we choose either AB or AD as primary key, then (A → C,E) violates 2NF.

c) Propose a lossless join decomposition of R that puts it into 3NF.
ANS:
One possibility is to first factor out B→D, second factor out A→C,E:
R(<u>AB</u>CDE) = R1(<u>B</u>D) join_B R2(<u>AB</u>CE)
= R1(<u>B</u>D) join_B (R3(<u>AB</u>) join_A R4(<u>A</u>CE))

d) Does your splitting preserve dependencies? (Do not worry if it does not!)
ANS:
No. (A→C,E) and (B→D) are preserved, but (BC→E) and (DE→B) are broken.