Database Management System – 36 Database design – Properties of Relational Decompositions

Ajay James
Asst. Prof in CSE
Government Engineering College Thrissur

Outline

- Introduction
- Dependency Preservation
- Lossless join
- Algorithm to test lossless join property
- Binary decomposition – lossless join

Introduction

- Universal Relation Schema:
  - A relation schema R = {A_1, A_2, …, A_n} that includes all the attributes of the database
- Decomposition:
  - Process of decomposing the universal relation schema R into a set of relation schemas D = {R_1,R_2, …, R_m} that will become the relational database schema by using the functional dependencies

Properties of Relational Decompositions

- Attribute preservation condition:
  - Each attribute in R will appear in at least one relation schema R_i in the decomposition so that no attributes are “lost”
- Each individual relation R_i in the decomposition D should be in BCNF or 3NF
- Prevent from generating spurious tuples
- Dependency Preservation

Example

EMP_LOCS

Ename	Plocation
Smith, John B.	Bellaire
Smith, John B.	Sugarland
Narayan, Ramesh K.	Houston
English, Joyce A.	Bellaire
English, Joyce A.	Sugarland
Wong, Franklin T.	Sugarland
Wong, Franklin T.	Houston
Wong, Franklin T.	Stafford
Zelaya, Alicia J.	Stafford
Jabbar, Ahmad V.	Stafford
Wallace, Jennifer S.	Stafford
Wallace, Jennifer S.	Houston
Borg, James E.	Houston

EMP_PROJ1

Ssn	Pnumber	Hours	Pname	Plocation
123456789	1	32.5	ProductX	Bellaire
123456789	2	7.5	ProductY	Sugarland
666884444	3	40.0	ProductZ	Houston
453453453	1	20.0	ProductX	Bellaire
453453453	2	20.0	ProductY	Sugarland
333445555	2	10.0	ProductY	Sugarland
333445555	3	10.0	ProductZ	Houston
333445555	10	10.0	Computerization	Stafford
333445555	20	10.0	Reorganization	Houston
999887777	30	30.0	Newbenefits	Stafford
999887777	10	10.0	Computerization	Stafford
987987987	10	35.0	Computerization	Stafford
987987987	30	5.0	Newbenefits	Stafford
987654321	30	20.0	Newbenefits	Stafford
987654321	20	15.0	Reorganization	Houston
888665555	20	NULL	Reorganization	Houston

Example contd...

	Ssn	Pnumber	Hours	Pname	Plocation	Ename
	123456789	1	32.5	ProductX	Bellaire	Smith, John B.
*	123456789	1	32.5	ProductX	Bellaire	English, Joyce A.
	123456789	2	7.5	ProductY	Sugarland	Smith, John B.
*	123456789	2	7.5	ProductY	Sugarland	English, Joyce A.
*	123456789	2	7.5	ProductY	Sugarland	Wong, Franklin T.
	666884444	3	40.0	ProductZ	Houston	Narayan, Ramesh K.
*	666884444	3	40.0	ProductZ	Houston	Wong, Franklin T.
*	453453453	1	20.0	ProductX	Bellaire	Smith, John B.
	453453453	1	20.0	ProductX	Bellaire	English, Joyce A.
*	453453453	2	20.0	ProductY	Sugarland	Smith, John B.
	453453453	2	20.0	ProductY	Sugarland	English, Joyce A.
*	453453453	2	20.0	ProductY	Sugarland	Wong, Franklin T.
*	333445555	2	10.0	ProductY	Sugarland	Smith, John B.
*	333445555	2	10.0	ProductY	Sugarland	English, Joyce A.
	333445555	2	10.0	ProductY	Sugarland	Wong, Franklin T.
*	333445555	3	10.0	ProductZ	Houston	Narayan, Ramesh K.
	333445555	3	10.0	ProductZ	Houston	Wong, Franklin T.
	333445555	10	10.0	Computerization	Stafford	Wong, Franklin T.
*	333445555	20	10.0	Reorganization	Houston	Narayan, Ramesh K.
	333445555	20	10.0	Reorganization	Houston	Wong, Franklin T.

- NATURAL JOIN to the tuples in EMP_PROJ1 and EMP_LOCS just for employee with Ssn = “123456789”.

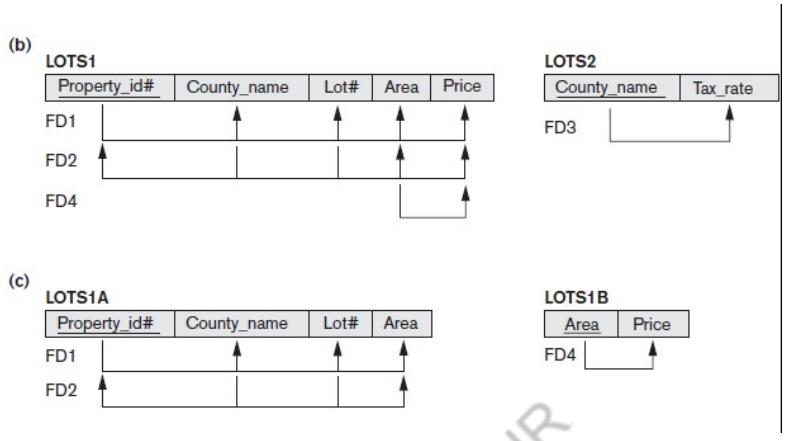
Properties of Relational Decompositions

- **Projection**
- Definition:
  - Given a set of dependencies F on R,
  - the projection of F on R_i, denoted by $\pi_{Ri}(F)$
  - where R_i is a subset of R,
  - is the **set of dependencies X → Y in F^+**
  - such that the attributes in X U Y are all contained in R_i

Dependency Preservation Property of a Decomposition

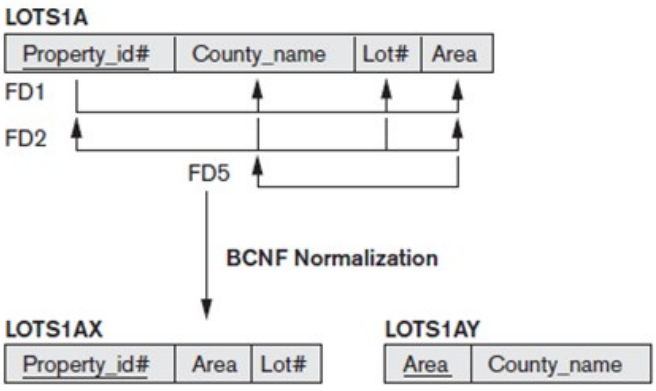
- A decomposition D = {R_1, R_2, ..., R_m} of R is dependency-preserving with respect to F
  - if the union of the projections of F on each R_i in D is equivalent to F; that is
  - $((\pi_{R1}(F)) \cup \ldots \cup (\pi_{Rm}(F)))^+ = F^+$

Dependency preservation example



Non dependency preserving

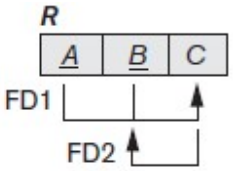
- FD2 is lost



Non dependency preserving

TEACH

Student	Course	Instructor
Narayan	Database	Mark
Smith	Database	Navathe
Smith	Operating Systems	Ammar
Smith	Theory	Schulman
Wallace	Database	Mark
Wallace	Operating Systems	Ahamad
Wong	Database	Omiecinski
Zelaya	Database	Navathe
Narayan	Operating Systems	Ammar



- FD1: {Student, Course} → Instructor
- FD2:Instructor → Course

1. R1 (Student, Instructor) and R2(Student, Course)
2. R1 (Course, Instructor) and R2(Course, Student)
3. R1 (Instructor, Course) and R2(Instructor, Student)

Dependency Preservation Property of a Decomposition

- Claim 1:
  - It is always possible to find a dependency-preserving decomposition D with respect to F such that each relation R_i in D is in 3nf.

Non-additive (Lossless) Join Property of a Decomposition

- Definition:
- A decomposition D = {R_1, R_2, ..., R_m} of R has the lossless (non-additive) join property wrt to the set of dependencies F on R if,
- for every relation state r of R that satisfies F, the following holds,
- where * is the natural join of all the relations in
- D: * (π_{R1}(r), ..., π_{Rm}(r)) = r

Algorithm: Testing for Lossless Join Property

- Input: A universal relation R, a decomposition D = {R1, R2, … , Rm} of R, and a set F of functional dependencies

1. Create an initial matrix S with one row i for each relation R_i in D, and one column j for each attribute A_j in R.
2. Set $S(i, j)$: = b_{ij} for all matrix entries. (*Each b_{ij} is a distinct symbol associated with indices (i, j)*)
3. For each row i representing relation schema R_i
   {for each column j representing attribute A_j
   {if (relation R_i includes attribute A_j) then set $S(i, j)$: = a_j;};}; (*Each a_j is a distinct symbol associated with index (j)*)

Testing for Lossless Join Property

4. Repeat the following loop until a *complete loop execution* results in no changes to S
{for each functional dependency $X \rightarrow Y$ in F
{for all rows in S *that have the same symbols* in the columns corresponding to attributes in X
{make the symbols in each column that correspond to an attribute in Y be the same in all these rows as follows: If any of the rows has an a symbol for the column, set the other rows to that *same* a symbol in the column. If no a symbol exists for the attribute in any of the rows, choose one of the b symbols that appears in one of the rows for the attribute and set the other rows to that same b symbol in the column ;} ; } ;};
5. If a row is made up entirely of a symbols, then the decomposition has the nonadditive join property; otherwise, it does not.

Example

R = {Ssn, Ename, Pnumber, Pname, Plocation, Hours} $D = \{R_1, R_2\}$
R_1 = EMP_LOCS = {Ename, Plocation}
R_2 = EMP_PROJ1 = {Ssn, Pnumber, Hours, Pname, Plocation}

F = {Ssn → Ename; Pnumber → {Pname, Plocation}; {Ssn, Pnumber} → Hours}

	Ssn	Ename	Pnumber	Pname	Plocation	Hours
R1	b11	b12	b13	b14	b15	b16
R2	b21	b22	b23	b24	b25	b26

	Ssn	Ename	Pnumber	Pname	Plocation	Hours
R1	b11	a2	b13	b14	a5	b16
R2	a1	b22	a3	a4	a5	a6

Example

EMP

Ssn	Ename

PROJECT

Pnumber	Pname	Plocation

WORKS_ON

Ssn	Pnumber	Hours

R = {Ssn, Ename, Pnumber, Pname, Plocation, Hours}
R_1 = EMP = {Ssn, Ename}
R_2 = PROJ = {Pnumber, Pname, Plocation}
R_3 = WORKS_ON = {Ssn, Pnumber, Hours}

$D = \{R_1, R_2, R_3\}$

F = {Ssn → Ename; Pnumber → {Pname, Plocation}; {Ssn, Pnumber} → Hours}

	Ssn	Ename	Pnumber	Pname	Plocation	Hours
R1	**a1**	**a2**	b13	b14	b15	b16
R2	b21	b22	**a3**	**a4**	**a5**	b26
R3	**a1**	b32	**a3**	b34	b35	**a6**

	Ssn	Ename	Pnumber	Pname	Plocation	Hours
R1	**a1**	**a2**	b13	b14	b15	b16
R2	b21	b22	**a3**	**a4**	**a5**	b26
R3	**a1**	~~b32~~ **a2**	**a3**	~~b34~~ **a4**	~~b35~~ **a5**	**a6**

Testing Binary Decompositions for the Nonadditive Join Property

- A decomposition D = {R1, R2} of R has the lossless (nonadditive) join property with respect to a set of functional dependencies F on R if and only if either
- The FD ((R1 ∩ R2) → (R1 − R2)) is in F^+, or
- The FD ((R1 ∩ R2) → (R2 − R1)) is in F^+

Example

TEACH

Student	Course	Instructor
Narayan	Database	Mark
Smith	Database	Navathe
Smith	Operating Systems	Ammar
Smith	Theory	Schulman
Wallace	Database	Mark
Wallace	Operating Systems	Ahamad
Wong	Database	Omiecinski
Zelaya	Database	Navathe
Narayan	Operating Systems	Ammar

- FD1: {Student, Course} → Instructor
- FD2:Instructor → Course

1. R1 (Student, Instructor) and R2(Student, Course)
2. R1 (Course, Instructor) and R2(Course, Student)
3. R1 (Instructor, Course) and R2(Instructor, Student)

Reference

- Elmasri R. and S. Navathe, Database Systems: Models, Languages, Design and Application Programming, Pearson Education 6th edition and 7th edition

Thank you