M.S. Ramaiah Institute of Technology
(Autonomous Institute, Affiliated to VTU)
Department of Computer Science and Engineering

Course Name: Database Systems
Course Code: CS52
Credits: 3:1:0
UNIT 4

Term: Oct 2021 – Feb 2022

Faculty:
Dr. Sini Anna Alex

RAMAIAH Institute of Technology

Examples:

R=(A,B,C,D,E,F)

F1={A->BC, B->CDE, AE->F}

F2={A->BCF, B->DE, E->AB}

Check whether F1 and F2 are equivalent or not.

Solution

To check F1 covers F2 –

A^+={A,B,C,D,E,F} contains B,C,F

B^+={B,C,D,E} contains D,E

E^+={E} contains A,B

So F1 does not cover F2.

Hence F1 and F2 are not equivalent.

Consider another example where two functional dependencies are equivalent.

R=(A,C,D,E,H)

F1={A->C, AC->D, E->AD, E->H},

F2={A->CD, E->AH}

Check whether F1 and F2 are equivalent or not?

Solution

To check F1 covers F2 –

A^+={A,C,D} contains C,D

E^+={A,D,E,H} contains A,H

So F1 covers F2

To check F2 covers F1:

A^+={A,C,D} contains C

{ A,C}+={A,C,D} contains D

E^+={A,C,D,E,H} contains A,D,H

So F2 covers F1.

Hence F1 and F2 are equivalent.

Minimal Sets of FDs

A set of FDs is **minimal** if it satisfies the following conditions:

1. Every dependency in F has a single attribute for its RHS.
2. We cannot replace any dependency X -> A in F with a dependency Y -> A, where Y proper-subset-of X (Y subset-of X) and still have a set of dependencies that is equivalent to F.
3. We cannot remove any dependency from F and have a set of dependencies that is equivalent to F.
   1. Simple FD
   2. Left Reduced FD
   3. Non-redundant FD

RAMAIAH Institute of Technology

Minimal Sets of Functional Dependencies

Algorithm 10.2: Finding a Minimal Cover F for a Set of Functional Dependencies E

1. Set $F := E$.
2. Replace each functional dependency $X \rightarrow \{A_1, A_2, \ldots, A_n\}$ in F by the n functional dependencies $X \rightarrow A_1, X \rightarrow A_2, \ldots, X \rightarrow A_n$.
3. For each functional dependency $X \rightarrow A$ in F

   for each attribute B that is an element of X

   if $\{ \{ F - \{X \rightarrow A\} \} \cup \{(X - \{B\}) \rightarrow A\} \}$ is equivalent to F,

   then replace $X \rightarrow A$ with $(X - \{B\}) \rightarrow A$ in F.
4. For each remaining functional dependency $X \rightarrow A$ in F

   if $\{ F - \{X \rightarrow A\} \}$ is equivalent to F,

   then remove $X \rightarrow A$ from F.

In Chapter 11 we will see how relations can be synthesized from a given set of dependencies E by first finding the minimal cover F for E.

Minimal Sets of FDs

- Every set of FDs has an equivalent minimal set.
- There can be several equivalent minimal sets.
- There is no simple algorithm for computing a minimal set of FDs that is equivalent to a set F of FDs.
- To synthesize a set of relations, we assume that we start with a set of dependencies that is a minimal set.
- Minimal Cover: set of FD’s equivalent to given FD’s
- Canonical Cover: Left side must be unique

Find Minimal Cover/ Canonical Cover of E

Let the given set of FDs be E : {B → A, D → A, AB → D}.We have to find the minimum cover of E.

- All above dependencies are in canonical form; so we have completed step 1 of Algorithm 10.2 and can proceed to step 2.

In step 2 we need to determine if AB → D has any redundant attribute on the left-hand side; that is, can it be replaced by B → D or A → D?

- Since B → A, by augmenting with B on both sides (IR2), we have BB → AB, or B → AB

(i). However, AB → D as given (ii).

- Hence by the transitive rule (IR3), we get from (i) and (ii), B → D. Hence AB → D may be replaced by B → D.
- We now have a set equivalent to original E , say E′ : {B → A, D → A, B → D}. No further reduction is possible in step 2 since all FDs have a single attribute on the left-hand side.
- In step 3 we look for a redundant FD in E′. By using the transitive rule on B → D and D → A, we derive B → A. Hence B → A is redundant in E′ and can be eliminated.
- **Hence the minimum cover of E is {B → D, D → A}**

Compute the Minimal Cover

Consider the following set *F* of functional dependencies:

F= {

$A \longrightarrow BC$

$B \longrightarrow C$

$A \longrightarrow B$

$AB \longrightarrow C$

}

Definitions of Keys and Attributes Participating in Keys

A **superkey** of a relation schema R = {A1, A2, ..., An} is a set of attributes S *subset-of* R with the property that no two tuples t1 and t2 in any legal relation state r of R will have t1[S] = t2[S]

A **key** K is a **superkey** with the *additional property* that removal of any attribute from K will cause K not to be a superkey any more.

Definitions of Keys and Attributes Participating in Keys

If a relation schema has more than one key, each is called a **candidate** key.

- One of the candidate keys is *arbitrarily* designated to be the **primary key**, and the others are called **secondary keys**.

A **Prime attribute** must be a member of *some* candidate key

A **Nonprime attribute** is not a prime attribute—that is, it is not a member of any candidate key.

Finding Candidate Keys and Super Keys of a Relation using FD set

The **set of attributes** whose attribute closure is set of all attributes of relation is called super key of relation.

Consider the following FD set. **{E-ID->E-NAME, E-ID->E-CITY, E-ID->E-STATE, E-CITY->E-STATE}**

Let us calculate attribute closure of different set of attributes: **EMPLOYEE(E-ID, E-NAME, E-CITY, E-STATE)**

(E-ID)+ = {E-ID, E-NAME,E-CITY,E-STATE}

(E-ID,E-NAME)+ = {E-ID, E-NAME,E-CITY,E-STATE}

(E-ID,E-CITY)+ = {E-ID, E-NAME,E-CITY,E-STATE}

(E-ID,E-STATE)+ = {E-ID, E-NAME,E-CITY,E-STATE}

(E-ID,E-CITY,E-STATE)+ = {E-ID, E-NAME,E-CITY,E-STATE}

(E-NAME)+ = {E-NAME}

(E-CITY)+ = {E-CITY,E-STATE}

First Normal Form

Disallows

- composite attributes
- multivalued attributes
- **nested relations**; attributes whose values for an *individual tuple* are non-atomic

Considered to be part of the definition of relation

Normalization into 1NF

(a)

DEPARTMENT

Dname	Dnumber	Dmgr_ssn	Dlocations

(b)

DEPARTMENT

Dname	Dnumber	Dmgr_ssn	Dlocations
Research	5	333445555	{Bellaire, Sugarland, Houston}
Administration	4	987654321	{Stafford}
Headquarters	1	888665555	{Houston}

(c)

DEPARTMENT

Dname	Dnumber	Dmgr_ssn	Dlocation
Research	5	333445555	Bellaire
Research	5	333445555	Sugarland
Research	5	333445555	Houston
Administration	4	987654321	Stafford
Headquarters	1	888665555	Houston

Figure 10.8
Normalization into 1NF. (a) A relation schema that is not in 1NF. (b) Example state of relation DEPARTMENT. (c) 1NF version of the same relation with redundancy.

Normalization nested relations into 1NF

(a)

EMP_PROJ

Ssn	Ename	Projs	
		Pnumber	Hours

(b)

EMP_PROJ

Ssn	Ename	Pnumber	Hours
123456789	Smith, John B.	1	32.5
		2	7.5
666884444	Narayan, Ramesh K.	3	40.0
453453453	English, Joyce A.	1	20.0
		2	20.0
333445555	Wong, Franklin T.	2	10.0
		3	10.0
		10	10.0
		20	10.0
999887777	Zelaya, AliciaJ.	30	30.0
		10	10.0
987987987	Jabbar, Ahmad V.	10	35.0
		30	5.0
987654321	Wallace, Jennifer S.	30	20.0
		20	15.0
888665555	Borg, James E.	20	NULL

(c)

EMP_PROJ1

Ssn	Ename

EMP_PROJ2

Ssn	Pnumber	Hours

Figure 10.9
Normalizing nested relations into 1NF. (a) Schema of the EMP_PROJ relation with a *nested relation* attribute PROJS. (b) Example extension of the EMP_PROJ relation showing nested relations within each tuple. (c) Decomposition of EMP_PROJ into relations EMP_PROJ1 and EMP_PROJ2 by propagating the primary key.

Thank you