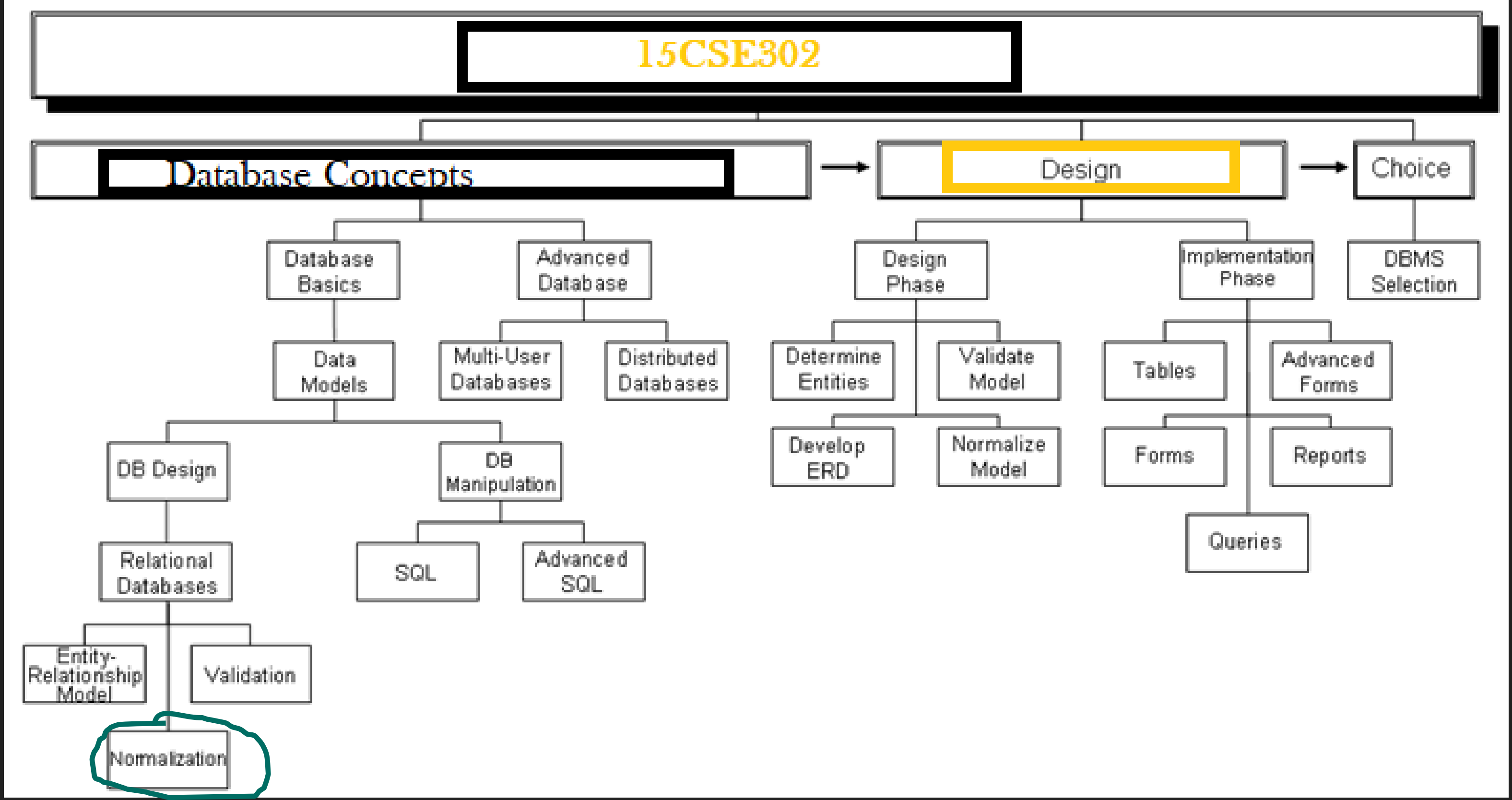
15CSE302 Database Management Systems
Lecture 24 Dependency Preserving

B.Tech /III Year CSE/V Semester **L T P C 2 0 2 3**

DBMS Team
Dr G Jeyakumar
Bindu K R
Dr Priyanka Kumar
R. Manjusha
Department of CSE
Amrita School of Engineering

Slides Courtesy : Jason Brown

Syllabus



15CSE302
Database Concepts
Design
Choice
Database Basics
Advanced Database
Design Phase
Implementation Phase
DBMS Selection
Data Models
Multi-User Databases
Distributed Databases
Determine Entities
Validate Model
Tables
Advanced Forms
DB Design
DB Manipulation
Develop ERD
Normalize Model
Forms
Reports
Queries
Relational Databases
SQL
Advanced SQL
Entity-Relationship Model
Validation
Normalization

Brief Recap of Previous Lecture

- Chase method -Lossless Join Decomposition

Today we'll discuss

Dependency Preserving

Desirable properties of Decomposition

When we decompose a relation schema R with a set of functional dependencies F into R_1, R_2, ..., R_n ,these properties should be satisfied

- **Non-additive (Lossless) join decomposition:**
  - Otherwise decomposition would result in information loss.
- **Dependency preservation:**
  - Otherwise, checking updates for violation of functional dependencies may require computing joins, which is expensive.
- **No redundancy:**

  The relations R_i preferably should be in either Boyce-Codd Normal Form or 3NF.

Dependency Preservation

Slides Courtesy: Jason Allen

Why Do We Preserve The Dependency?

- We would like to check easily that updates to the database do not result in illegal relations being created.

- It would be nice if our design allowed us to check updates without having to compute natural joins.

Dependency Preserving :Definition

- A decomposition $D = \{R_1, R_2, ..., R_n\}$ 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

  if $$(F_1 \cup F_2 \cup ... \cup F_n)^+ = F^+$$

Dependency Preserving :Definition

In Layman’s Term each Functional Dependency specified in F either appears directly in one of the relations in the decomposition.

Dependency Preserving

- It is not necessary that all dependencies from the relation R appear in some relation R_i.
- It is sufficient that the union of the dependencies on all the relations R_i be equivalent to the dependencies on R.

Property of Dependency-Preservation

- If a decomposition is not dependency-preserving, therefore, that dependency is lost in the decomposition.

Example of Dependency Preservation

- R(A B C D)

- FD_1: A → B
- FD_2: B → C
- FD_3: C → D

- Decomposition:
  R_1(A B C) R_2(C D)

Example of Dependency Preservation

- FD_1: A → B
- FD_2: B → C
- FD_3: C → D

R_1(A B C)

FD_1

FD_2

Example of Dependency Preservation

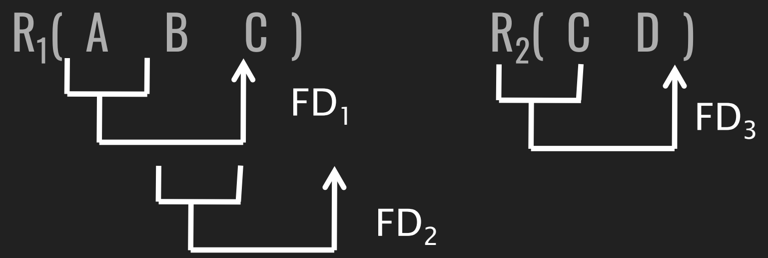
- FD_1: $A \rightarrow B$
- FD_2: $B \rightarrow C$
- FD_3: $C \rightarrow D$

$R_2($ C D $)$

FD_3

Example of Dependency Preservation

- FD_1: A → B
- FD_2: B → C
- FD_3: C → D



- Has all 3 functional dependencies!
- Therefore, it's preserving the dependencies

Example of Non-Dependency Preservation

- R(A B C D)

- FD_1: A → B
- FD_2: B → C
- FD_3: C → D

- Decomposition:
  R_1(A C D) R_2(B C)

Example of Non-Dependency Preservation

- FD_1: $A \rightarrow B$
- FD_2: $B \rightarrow C$
- FD_3: $C \rightarrow D$

$$R_1(\ A \quad C \quad D\ )$$

FD_3

Example of Non-Dependency Preservation

- FD_1: $A \rightarrow B$
- FD_2: $B \rightarrow C$
- FD_3: $C \rightarrow D$

$$R_2(\underbrace{B \qquad C}_{FD_2})$$

Example of Non-Dependency Preservation

- FD_1: $A \rightarrow B$
- FD_2: $B \rightarrow C$
- FD_3: $C \rightarrow D$

FD_3

$R_1(\ A \quad C \quad D\ )$

FD_2

$R_2(\ B \quad C\ )$

Does not support FD_1: $A \rightarrow B$
Therefore, it does not preserve the dependencies

Example 2 Dependency Preservation

- R(A B C D E)

- FD_1: A → B
- FD_2: BC → D

- Decomposition:
  R_1(A C E) R_2(B C D) R_3(A B)

Example 2 Dependency Preservation

- FD_1: A → B
- FD_2: BC → D

$$R_1(\ A \quad C \quad E\ )$$

No Dependencies

Example 2   Dependency Preservation

- FD_1: $A \rightarrow B$
- FD_2: $BC \rightarrow D$

$$R_2(\ B \quad C \quad D\ )$$

FD_2

Example 2   Dependency Preservation

- FD_1: $A \rightarrow B$
- FD_2: $BC \rightarrow D$

$$R_3(\; A \xrightarrow{FD_1} B \;)$$

Example 2 Dependency Preservation

- FD_1: A → B
- FD_2: BC → D

R_1(A C E)

R_3(A B)

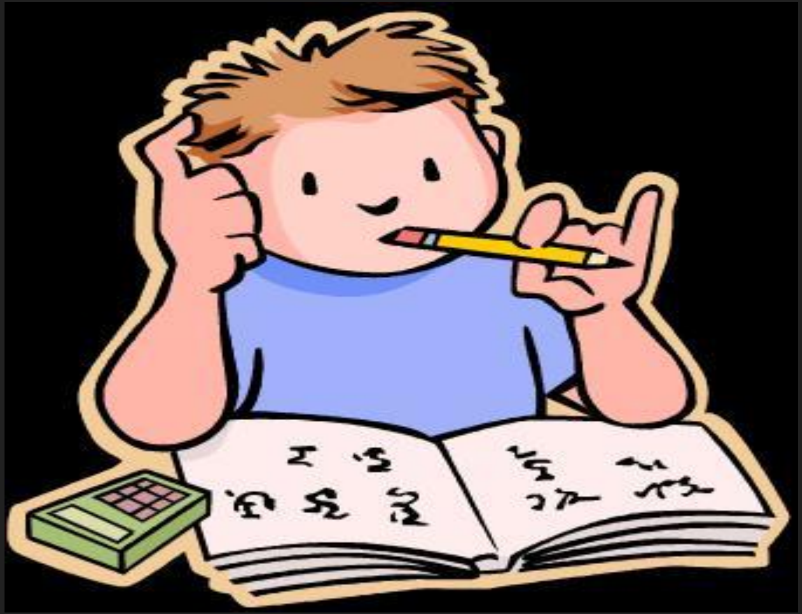
FD_1

R_2(B C D)

FD_2

Has all 2 functional dependencies!
Therefore, it's preserving the dependencies

Exercise Problem



Example 3 Dependency Preservation

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

- FD_1: D → A, B
- FD_2: C → E, F

- Decomposition:
  R_1(A, C, D)
  R_2(A, D, B)
  R_3(D, E, F)
  R_4(C, E, F)

R_1(A C D)

R_2(A D B)

FD$_1$

R_3(D E F)

R_4(C E F)

FD$_2$

FD1: D ➔ A, B
FD2: C ➔ E, F

Answer

- Yes!
  This is a dependency-preservation

Example 4 Check if it is Lossless Decomposition and Dependency Preservation

Consider the following relationship : **R (A,B,C,D)**

and following dependencies :

A -> BCD
BC -> AD
D -> B

Consider the following relationship : **R (A,B,C,D)**

and following dependencies :

A -> BCD
BC -> AD
D -> B

Above relationship is already in 3rd NF. Keys are **A** and **BC.**

Hence, in the functional dependency, **A -> BCD**, A is the super key.
in second relation, **BC -> AD**, BC is also a key.
but in, **D -> B**, D is not a key.

Hence we can break our relationship R into two relationships **R1** and **R2**.

R(A, B, C, D)

R1 (A, D, C) R2(D, B)

Breaking, table into two tables, one with A, D and C while the other with D and B.

- R1(A,B,C,D,E)   R2(B,F)
- CANDIDATE KEY IS B
- A→BCDE
- BC→ADE
- D→E   R1

- 2NF

R2(A,B,C,D)   R3(D,E)

Normalisation

Steps

- 1NF
  - – Removing repeating groups
- 2NF
  - – Remove partial dependencies
- • 3NF
  - – Remove transitive dependencies
- BCNF
  - – Remove non-candidate key dependencies

Project Code	Project Title	Project Manager	Project Budget	Employee No.	Employee Name	Department No.	Department Name	Hourly Rate
PC010	Pensions System	M Phillips	24500	S10001	A Smith	L004	IT	22.00
PC010	Pensions System	M Phillips	24500	S10030	L Jones	L023	Pensions	18.50
PC010	Pensions System	M Phillips	24500	S21010	P Lewis	L004	IT	21.00
PC045	Salaries System	H Martin	17400	S10010	B Jones	L004	IT	21.75
PC045	Salaries System	H Martin	17400	S10001	A Smith	L004	IT	18.00
PC045	Salaries System	H Martin	17400	S31002	T Gilbert	L028	Database	25.50
PC045	Salaries System	H Martin	17400	S13210	W Richards	L008	Salary	17.00
PC064	HR System	K Lewis	12250	S31002	T Gilbert	L028	Database	23.25
PC064	HR System	K Lewis	12250	S21010	P Lewis	L004	IT	17.50
PC064	HR System	K Lewis	12250	S10034	B James	L009	HR	16.50

1NF Tables: Repeating Attributes Removed

Project Code	Employee No.	Employee Name	Department No.	Department Name	Hourly Rate
PC010	S10001	A Smith	L004	IT	22.00
PC010	S10030	L Jones	L023	Pensions	18.50
PC010	S21010	P Lewis	L004	IT	21.00
PC045	S10010	B Jones	L004	IT	21.75
PC045	S10001	A Smith	L004	IT	18.00
PC045	S31002	T Gilbert	L028	Database	25.50
PC045	S13210	W Richards	L008	Salary	17.00
PC064	S31002	T Gilbert	L028	Database	23.25
PC064	S21010	P Lewis	L004	IT	17.50
PC064	S10034	B James	L009	HR	16.50

Project Code	Project Title	Project Manager	Project Budget
PC010	Pensions System	M Phillips	24500
PC045	Salaries System	H Martin	17400
PC064	HR System	K Lewis	12250

2NF Tables: Partial Key Dependencies Removed

Project Code	Project Title	Project Manager	Project Budget
PC010	Pensions System	M Phillips	24500
PC045	Salaries System	H Martin	17400
PC064	HR System	K Lewis	12250

Project Code	Employee No.	Hourly Rate
PC010	S10001	22.00
PC010	S10030	18.50
PC010	S21010	21.00
PC045	S10010	21.75
PC045	S10001	18.00
PC045	S31002	25.50
PC045	S13210	17.00
PC064	S31002	23.25
PC064	S21010	17.50
PC064	S10034	16.50

Employee No.	Employee Name
S10001	A Smith
S10030	L Jones
S21010	P Lewis
S10010	B Jones
S31002	T Gilbert
S13210	W Richards
S10034	B James

Department No.	Department Name
L004	IT
L023	Pensions
L004	IT
L004	IT
L028	Database
L008	Salary
L009	HR

References

- Hillyer Mike, MySQL AB. An Introduction to Database Normalization, http://dev.mysql.com/tech-resources/articles/intro-to-normalization.html, accessed October 17, 2006.

- Microsoft. Description of the database normalization basics, http://support.microsoft.com/kb/283878 , accessed October 17, 2006.

- Wikipedia. Database Normalization. http://en.wikipedia.org/wiki/Database_normalization.html , accessed October 17, 2006.
  https://www.db-book.com/db6/index.html
- https://www.youtube.com/watch?v=mfVCesoMaGA&list=PLroEs25KGvwzmvIxYHRhoGTz9w8LeXek0&index=22

Reference

- Chen, Y. (2005). “Decomposition”. Retrieved on March 21, 2010 from http://www.cs.sjsu.edu/faculty/lee/cs157/Decomposition_YuHungChen.ppt
- Kamel, A. “Chapter 11Relational Database Design Algorithms” Retrieved on March 22, 2010 from http://www.cord.edu/faculty/kamel/08F-330/Presentations/ch11.pdf
- Lee, S. “Huffman code and Lossless Decomposition”. Retrieved on March 21, 2010 from http://www.cs.sjsu.edu/~lee/cs157b/29SpCS157BL14HuffmanCode&LosslessDecomposition.ppt
- Zaiane, O. (1998) “Dependency Preservation”. Retrieved on March 21, 2010 from http://www.cs.sfu.ca/CC/354/zaiane/material/notes/Chapter7/node8.html

Summary

- Normalization basics
- Anomalies

Next Lecture

Functional dependency

Thank You

Happy to answer any questions ! ! !