

Database Management System 15

Database Design

Database Design

Bad Database
Design/Concept of
Anomalies

Functional
Dependency(FD)

Trivial FDs and Non-Trivial
FDs

Armstrong's Inference
Axioms

Logical Implication

Closure of a Set of
Functional Dependencies

Closure of a Set of
Attributes

Redundancy of FDs

Canonical Cover/Minimal
Cover

Chittaranjan Pradhan
School of Computer Engineering,
KIIT University

Database Design

- First characterize fully the data requirements of the prospective database users, which usually involves in **textual** descriptions
- Next, choose **ER model** to translate these requirements into a conceptual schema of the database
- In the logical design phase, map the high level conceptual schema onto the implementation data model of the database system that will be used. The implementation data model is typically the **Relational data model**
- Finally, use the resulting system specific database schema in the subsequent physical design phase, in which the physical features of the database are specified
- In designing a database schema, the major pitfalls which should be avoided are:
 - **redundancy**: it means repetition of the information
 - **incompleteness**: it means certain aspects of the enterprise may not be modeled due to difficulty or complexity

Database Design

Bad Database Design/Concept of Anomalies

Functional Dependency(FD)

Trivial FDs and Non-Trivial FDs

Armstrong's Inference Axioms

Logical Implication

Closure of a Set of Functional Dependencies

Closure of a Set of Attributes

Redundancy of FDs

Canonical Cover/Minimal Cover

Bad Database Design/Concept of Anomalies

Bad Database Design/Concept of Anomalies

Database anomalies are the problems in relations that occur due to redundancy in the relations. These anomalies affect the process of inserting, deleting and updating data in the relations

The intension of relational database theory is to eliminate anomalies from occurring in a database

Student database

Name	Course	Phone_no	Major	Prof	Grade
Mahesh	353	1234	Comp sc	Alok	A
Nitish	329	2435	Chemistry	Pratap	B
Mahesh	328	1234	Comp sc	Samuel	B
Harish	456	4665	Physics	James	A
Pranshu	293	4437	Decision sc	Sachin	C
Prateek	491	8788	Math	Saurav	B
Prateek	356	8788	Math	Sunil	In prog
Mahesh	492	1234	Comp sc	Paresh	In prog
Sumit	379	4575	English	Rakesh	C

Bad Database Design/Concept of Anomalies...

Insertion Anomaly

It is the anomaly in which the user cannot insert a fact about an entity until he/she has an additional fact about another entity. In other words, there are circumstances in which certain facts can not be recorded at all.

Ex: We cannot record a new prof details without assigning a course to him

Deletion Anomaly

It is the anomaly in which the deletion of facts about an entity automatically deleted the fact of another entity.

Ex: If we want to delete the information about course 293, automatically the information of prof Sachin will be deleted which is not our interest

Updation Anomaly

It is the anomaly in which the modification in the value of specific attribute requires modification in all records in which that value occurs. In other words, the same data can be expressed in multiple rows. Therefore, updates to the table may result in logical inconsistencies.

Ex: If the updation to the phone_no of Mahesh is done in a single row only, then the updation process will put the database in an inconsistent state so that the phone_no of Mahesh will give conflicting answers

Functional Dependency(FD)

Functional Dependency is the building block of normalization principles

Attribute(s) A in a relation schema R functionally determines another attribute(s) B in R if for a given value a_1 of A ; there is a single, specific value b_1 of B in relation r of R

The symbolic expression of this FD is:

$A \rightarrow B$

where A (LHS of FD) is known as the **determinant** and B (RHS of FD) is known as the **dependent**

In other words, if A functionally determines B in R , then it is invalid to have two or more tuples that have the same A value, but different B values in R

Database Design

Bad Database
Design/Concept of
AnomaliesFunctional
Dependency(FD)Trivial FDs and Non-Trivial
FDsArmstrong's Inference
Axioms

Logical Implication

Closure of a Set of
Functional DependenciesClosure of a Set of
Attributes

Redundancy of FDs

Canonical Cover/Minimal
Cover

Functional Dependency(FD)

From Student schema, we can infer that $\text{Name} \rightarrow \text{Phone_no}$ because all tuples of Student with a given Name value also have the same Phone_no value

Likewise, it can also be inferred that $\text{Prof} \rightarrow \text{Grade}$. At the same time, notice that Grade does not determine Prof

When the determinant or dependent in an FD is a composite attribute, the constituent atomic attributes are enclosed by braces as shown in the following example:

$\{\text{Name, Course}\} \rightarrow \text{Phone_no}$

FD is a constraint between two sets of attributes in a relation from a database

Database Design

Bad Database
Design/Concept of
AnomaliesFunctional
Dependency(FD)Trivial FDs and Non-Trivial
FDsArmstrong's Inference
Axioms

Logical Implication

Closure of a Set of
Functional DependenciesClosure of a Set of
Attributes

Redundancy of FDs

Canonical Cover/Minimal
Cover

Trivial FDs and Non-Trivial FDs

Trivial FDs

A functional dependency $X \rightarrow Y$ is a trivial functional dependency if Y is a subset of X

For example, $\{\text{Name, Course}\} \rightarrow \text{Course}$. If two records have the same values on both the Name and Course attributes, then they obviously have the same Course

Trivial dependencies hold for all relation instances

Non-Trivial FDs

A functional dependency $X \rightarrow Y$ is called as non-trivial type if $Y \cap X = \emptyset$

For example, $\text{Prof} \rightarrow \text{Grade}$

Non-trivial FDs are given implicitly in the form of constraints when designing a database

Armstrong's Inference Axioms

The inference axioms or rules allow users to infer the FDs that are satisfied by a relation

Let $R(X, Y, Z, W)$ where X, Y, Z , and W are arbitrary subsets of the set of attributes of a universal relation schema R

The three fundamental inference rules are:

- **Reflexivity Rule:** If Y is a subset of X , then $X \rightarrow Y$ (Trivial FDs). Ex: $\{Name, Course\} \rightarrow Course$
- **Augmentation Rule:** If $X \rightarrow Y$, then $\{X, Z\} \rightarrow \{Y, Z\}$. Ex: as $Prof \rightarrow Grade$, therefore $\{Prof, Major\} \rightarrow \{Grade, Major\}$
- **Transitivity Rule:** If $X \rightarrow Y$ and $Y \rightarrow Z$, then $X \rightarrow Z$. Ex: as $Course \rightarrow Name$ and $Name \rightarrow Phone_no$ functional dependencies are present, therefore $Course \rightarrow Phone_no$

[Database Design](#)[Bad Database Design/Concept of Anomalies](#)[Functional Dependency\(FD\)](#)[Trivial FDs and Non-Trivial FDs](#)[Armstrong's Inference Axioms](#)[Logical Implication](#)[Closure of a Set of Functional Dependencies](#)[Closure of a Set of Attributes](#)[Redundancy of FDs](#)[Canonical Cover/Minimal Cover](#)

Armstrong's Inference Axioms

The four secondary inference rules are:

- **Union or Additive Rule:** If $X \rightarrow Y$ and $X \rightarrow Z$, then $X \rightarrow \{Y, Z\}$. Ex: as $Prof \rightarrow Grade$ and $Prof \rightarrow Course$ FDs are present; therefore, $Prof \rightarrow \{Grade, Course\}$
- **Decomposition Rule:** If $X \rightarrow \{Y, Z\}$, then $X \rightarrow Y$ and $X \rightarrow Z$. Ex: if $Prof \rightarrow \{Grade, Course\}$, then this FD can be decomposed as $Prof \rightarrow Grade$ and $Prof \rightarrow Course$
- **Composition Rule:** If $X \rightarrow Y$ and $Z \rightarrow W$, then $\{X, Z\} \rightarrow \{Y, W\}$. Ex: if $Prof \rightarrow Grade$ and $Name \rightarrow Phone_no$, then the FDs can be composed as $\{Prof, Name\} \rightarrow \{Grade, Phone_no\}$
- **Pseudotransitivity Rule:** If $X \rightarrow Y$ and $\{Y, W\} \rightarrow Z$, then $\{X, W\} \rightarrow Z$. Ex: if $Prof \rightarrow Grade$ and $\{Grade, Major\} \rightarrow Course$, then the FD $\{Prof, Major\} \rightarrow Course$ is valid

Logical Implication

Given a relation schema R and a set of functional dependencies F . Let FD $X \rightarrow Y$ is not in F . *F can be said to logically imply $X \rightarrow Y$ if for every relation r on the relation schema R that satisfies the FD in F , the relation r also satisfies $X \rightarrow Y$*

F logically implies $X \rightarrow Y$ is written as:

$$F \models X \rightarrow Y$$

Let $R = (A, B, C, D)$ and $F = \{A \rightarrow B, A \rightarrow C, BC \rightarrow D\}$

$$F \models A \rightarrow D$$

Given $F = \{A \rightarrow B, C \rightarrow D\}$ with $C \subseteq B$, show that $F \models A \rightarrow D$

Database Design

Bad Database
Design/Concept of
AnomaliesFunctional
Dependency(FD)Trivial FDs and Non-Trivial
FDsArmstrong's Inference
Axioms

Logical Implication

Closure of a Set of
Functional DependenciesClosure of a Set of
Attributes

Redundancy of FDs

Canonical Cover/Minimal
Cover

Closure of a Set of Functional Dependencies

Closure of a Set of Functional Dependencies

Given a set F of functional dependencies for a relation schema R , we define F^+ , the closure of F , to be the set of all functional dependencies that are logically implied by F . Mathematically,

$$F^+ = \{X \rightarrow Y / F \models X \rightarrow Y\}$$

To generate all FDs that can be derived from F , the steps are:

- First, apply the inference axioms to all single attributes and use the FDs of F whenever it is applicable
- Second, apply the inference axioms to all combinations of two attributes and use the functional dependencies of F whenever it is applicable
- Next apply the inference axioms to all combinations of three attributes and use the FDs of F when necessary
- Proceed in this manner for as many different attributes as there are in F

Closure of a Set of Functional Dependencies...

Let $R=(A, B, C)$ and $F=\{A \rightarrow B, A \rightarrow C\}$

$F^+ = \{A \rightarrow A, B \rightarrow B, C \rightarrow C, A \rightarrow B, A \rightarrow C, A \rightarrow BC, A \rightarrow AB, A \rightarrow AC, AB \rightarrow A, AB \rightarrow B, AB \rightarrow AB, AC \rightarrow A, AC \rightarrow C, AC \rightarrow AC, BC \rightarrow B, BC \rightarrow C, BC \rightarrow BC, ABC \rightarrow AB, ABC \rightarrow AC, ABC \rightarrow BC, ABC \rightarrow ABC\}$

Let $R=(W, X, Y)$ and $F=\{W \rightarrow X, X \rightarrow Y, W \rightarrow XY\}$

Uses of set of Functional Dependency Closure:

- **Computing if two sets of functional dependencies F and G are equivalent:** When $F^+ = G^+$, then the functional dependencies sets F and G are equivalent

$F=\{W \rightarrow X, X \rightarrow Y, W \rightarrow XY\}$ and $G= \{W \rightarrow X, W \rightarrow Y, X \rightarrow Y\}$

Closure of a Set of Attributes

Closure of a Set of Attributes

Given a set of attributes X and a set of functional dependencies F , then the closure of the set of attributes X under F , denoted as X^+ , is the set of attributes A that can be derived from X by applying the Armstrong's Inference Axioms to the functional dependencies of F

The closure of X is always a non empty set

$R = (A, B, C, D)$ and $F = \{A \rightarrow C, B \rightarrow D\}$

$\{A\}^+ = \{A, C\}$, $\{B\}^+ = \{B, D\}$, $\{C\}^+ = \{C\}$, $\{D\}^+ = \{D\}$,
 $\{A, B\}^+ = \{A, B, C, D\}$, $\{A, C\}^+ = \{A, C\}$, $\{A, D\}^+ = \{A, C, D\}$,
 $\{B, C\}^+ = \{B, C, D\}$, $\{B, D\}^+ = \{B, D\}$, $\{C, D\}^+ = \{C, D\}$,
 $\{A, B, C\}^+ = \{A, B, C, D\}$, $\{A, B, D\}^+ = \{A, B, C, D\}$,
 $\{B, C, D\}^+ = \{B, C, D\}$, $\{A, B, C, D\}^+ = \{A, B, C, D\}$

$R = (X, Y, Z)$ and $F = \{X \rightarrow Y, Y \rightarrow Z\}$

Uses of Attribute Closure:

- **Testing for key:** To test whether X is a key or not, X^+ is computed. X is a key iff X^+ contains all the attributes of R . X is a candidate key if none of its subsets is a key
- **Testing functional dependencies:** To check whether a functional dependency $X \rightarrow Y$ holds or not, just check if $Y \subseteq X^+$

Given $R=(A, B, C, D)$ and $F=\{AB \rightarrow C, B \rightarrow D, D \rightarrow B\}$, find the candidate keys of the relation. How many candidate keys are in this relation?

Redundancy of FDs

Redundancy of FDs

Given a set of functional dependencies F , a functional dependency $A \rightarrow B$ of F is said to be redundant with respect to the FDs of F if and only if $A \rightarrow B$ can be derived from the set of FDs $F - \{A \rightarrow B\}$

Eliminating redundant functional dependencies allows us to minimize the set of FDs

Ex: $A \rightarrow C$ is redundant in $\{A \rightarrow B, B \rightarrow C, A \rightarrow C\}$

Redundant attribute on RHS

In a functional dependency, some attributes in the RHS may be redundant

Ex: $F = \{A \rightarrow B, B \rightarrow C, A \rightarrow \{C, D\}\}$ can be simplified to $\{A \rightarrow B, B \rightarrow C, A \rightarrow D\}$

Redundant attribute on LHS

In a functional dependency, some attributes in the LHS may be redundant

Ex: $F = \{A \rightarrow B, B \rightarrow C, \{A, C\} \rightarrow D\}$ can be simplified to $\{A \rightarrow B, B \rightarrow C, A \rightarrow D\}$

Canonical Cover/Minimal Cover

For a given set F of FDs, a canonical cover, denoted by F_c , is a set of FDs where the following conditions are satisfied:

- F and F_c are equivalent
- Every FD of F_c is simple. That is, the RHS of every functional dependency of F_c has only one attribute
- No FD in F_c is redundant
- The determinant or LHS of every FD in F_c is irreducible

$R = (A, B, C)$ and $F = \{A \rightarrow \{B, C\}, B \rightarrow C, A \rightarrow B, \{A, B\} \rightarrow C\}$

$F_c = \{A \rightarrow B, B \rightarrow C\}$

As canonical cover contains the functional dependencies without any redundancy, therefore finding the key of the relation becomes efficient

Database Design

Bad Database
Design/Concept of
AnomaliesFunctional
Dependency(FD)Trivial FDs and Non-Trivial
FDsArmstrong's Inference
Axioms

Logical Implication

Closure of a Set of
Functional DependenciesClosure of a Set of
Attributes

Redundancy of FDs

Canonical Cover/Minimal
Cover