# Chapter 3: Relational DataBase Design Theory

Trivial: No

Key Points:

I. FD. Key. Superkey. FD rules Definition
Minimal Basis for FDs.

II. Boyce-Codd NF
Decomposition classification

III. 3rd-NF and its synthesis Algorithm

IV. Chase validation

V. MVD.
4th-NF and its chase validation.

# Lecture 4 Functional Dependency.

Definition: Given a relation R, if its two columns (or two set of columns) X, Y satisfies: Any two tuples of R agree on (all the columns of) X, they must agree on (all the columns of) Y. then we write it as: $X \xrightarrow{(R)} Y$

X determines Y

△ FD is a constrain, an assertion.

△ Definition of Trivial FD: $X \to Y$ && $Y \subseteq X$.

△ RHS: single attribute form of FD.

e.g. Original Form: $X \to A_1 \cdots A_n$

RHS form:

$$\begin{cases} X \to A_1 \\ X \to A_2 \\ \vdots \\ X \to A_n \end{cases}$$

## Superkey and key:

Key:

1. K → all attributes of R
2. K is minimal.

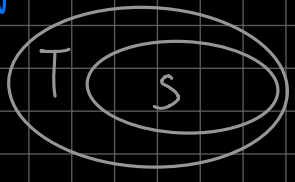
Super Key:

$K \Rightarrow$ all attributes of $R$

△ Each key is a Super Key.

# FD Reasoning (FD 推理规则):

follows from:




If every instance of $R$ satisfies $T$, then it also satisfies $S$, then we call it $S$ follows from $T$.

equivalent:

$S$ is equivalent to $T$ iff:

($S$ follows from $T$) && ($T$ follows from $S$)

# Reasoning Rules (推理规则):

- Splitting Rule: $X \to A_1 \cdots A_n \Rightarrow X \to A_1, \cdots X \to A_n$
- Combining Rule: $X \to A_1, \cdots X \to A_n \Rightarrow X \to A_1 \cdots A_n$
- Transitive Rule: $X \to Y,\ Y \to Z \Rightarrow X \to Z$
- Trivial-dependency Rule: $X \to Y \Rightarrow X \to Y - X$
  - trivial: $Y - X \subseteq X$
  - non trivial: Some $A_i \notin X$
  - completely nontrivial: all $A_i \notin X,\ (i = 1 \cdots m)$

$Y - X \Rightarrow A_1, \cdots A_m$

## Closure of Attributes: (属性闭包)

Definition: The closure of $X$: $X^+$ is the set of attributes $Y$ that $X \to Y$ follows from FD set $S$.

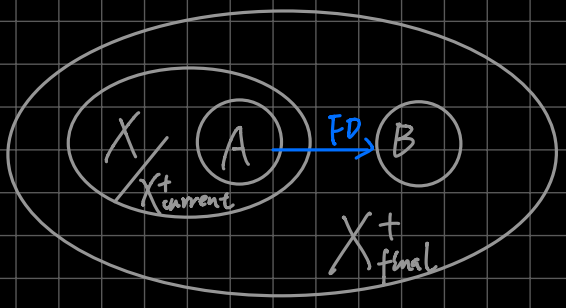
How to get $X^+$:

① Change FD to RHS form.

② Add $X$ to $X^+$ ($X^+ := X$)

③ Search for all the FD ($A \to B \in S$) that $(A \in X^+)$ && $(B \notin X^+)$, then $X^+ = X^+ \cup \{B\}$

④ Repeat ③ until no such FD can be found.




△ closure of attributes 并不改变给定的S集。

△ 计算 attribute / attributes 的 closure 时，closure 为 attribute set，与 FD 无关，因此不用考虑 Armstrong Axiom.

## Closure of FDs (函数依赖闭包)

Definition: Given a set of FDs S, infer all derived FDs.

Using the inference rules: Armstrong's Axioms.

Armstrong's Axioms:

1. Reflexivity (自反性): $Y \subseteq X \Rightarrow X \to Y$
2. Augmentation (增广律): $X \to Y \Rightarrow XZ \Rightarrow YZ$
3. Transitivity (传递性): $X \to Y, Y \to Z \Rightarrow X \to Z$

How ⇒ FD Closing set Algorithm:

Using attribute-closure algorithm to find

closure for every set of attributes. , then update the FDs by adding new FD ( $X \to Y'$ )

not exist in the current FD set S

and can't be inferred from current FD set.

using Armstrong Axioms (first two axiom ( not include the transitive axioms ))

e.g

F: $AB \to C,\ C \to D,\ D \to A$

| | Attribute closure | FD set |
|---|---|---|
| ① | $A^+ = A$ | n |
| | $B^+ = B$ | n |
| | $C^+ = ACD$ | ⇒ add $C \to A$ |
| | $D^+ = DA$ | n |
| ② | ✓ $AB^+ = ABCD$ | ⇒ add $AB \to D$ |
| | $AC^+ = ACD$ | |
| | $AD^+ = AD$ | |
| | ✓ $BC^+ = ABCD$ | |
| | ✓ $BD^+ = ABDC$ | ⇒ add $BD \to C$ |
| | $CD^+ = ACD$ | |

not add $CD \to A$ ⇒ 根据规则2., $C \to AB$ 有了.

$CD \to C$ } 根据规则1., $C, D \in CD$

$CD \to D$ }

③ ∵ $AB^+, BC^+, BD^+$ 均为 superkey,

因此所有包含此3个的 3元组均满足规则2而不能被加入

因此三元组只有1项:

$ACD^+ = ACD$

Thus, the FDs follow from F are:

$C \to A$, $AB \to D$, $BD \to C$.

## Application of Attribute-Closure Algorithm:

① Does $X \to A$ follows from S ?

$\Rightarrow$ calculate $X^+$ under S, and test if $A \in X^+$

- yes $\Rightarrow$ $X \to A$ follows from S
- no $\Rightarrow$ otherwise

② Is $\{A_1, \cdots, A_n\}$ a key ?

$\Rightarrow$ calculate $\underbrace{\{A_1, \cdots A_n\}}_{X}{}^+$ and test if ($X^+$ contains all attributes.)

&& ($S_i = X - \{A_i\}$, $S_i^+$ doesn't contain all

attributes for any $i \in \{1, 2, \ldots, n\}$)

yes, $\{A_1, A_2, \ldots A_n\}$ is key

no, otherwise

③ FDs. Closing Set Calculation ⇒ Calculate attribute-closure for any subset of X and check for new FDs and add them.

④ Find a Minimal Basis FDs.

Steps ⇒

I. Split to RHS.

II. Repeatedly remove an FD and see if the FDs remain un changed.

IiI. Repeatedly remove an attribute from a left side and see if the FDs remain un changed.

⑤ Projecting FDs.

$R \Rightarrow R_1 = \pi_L(R)$

$F \Rightarrow F_1$

follows from F

involves only attributes of $R_2$.

Steps ⇒ I. Calculate the Closure of $F_i$ and add the new FDs.

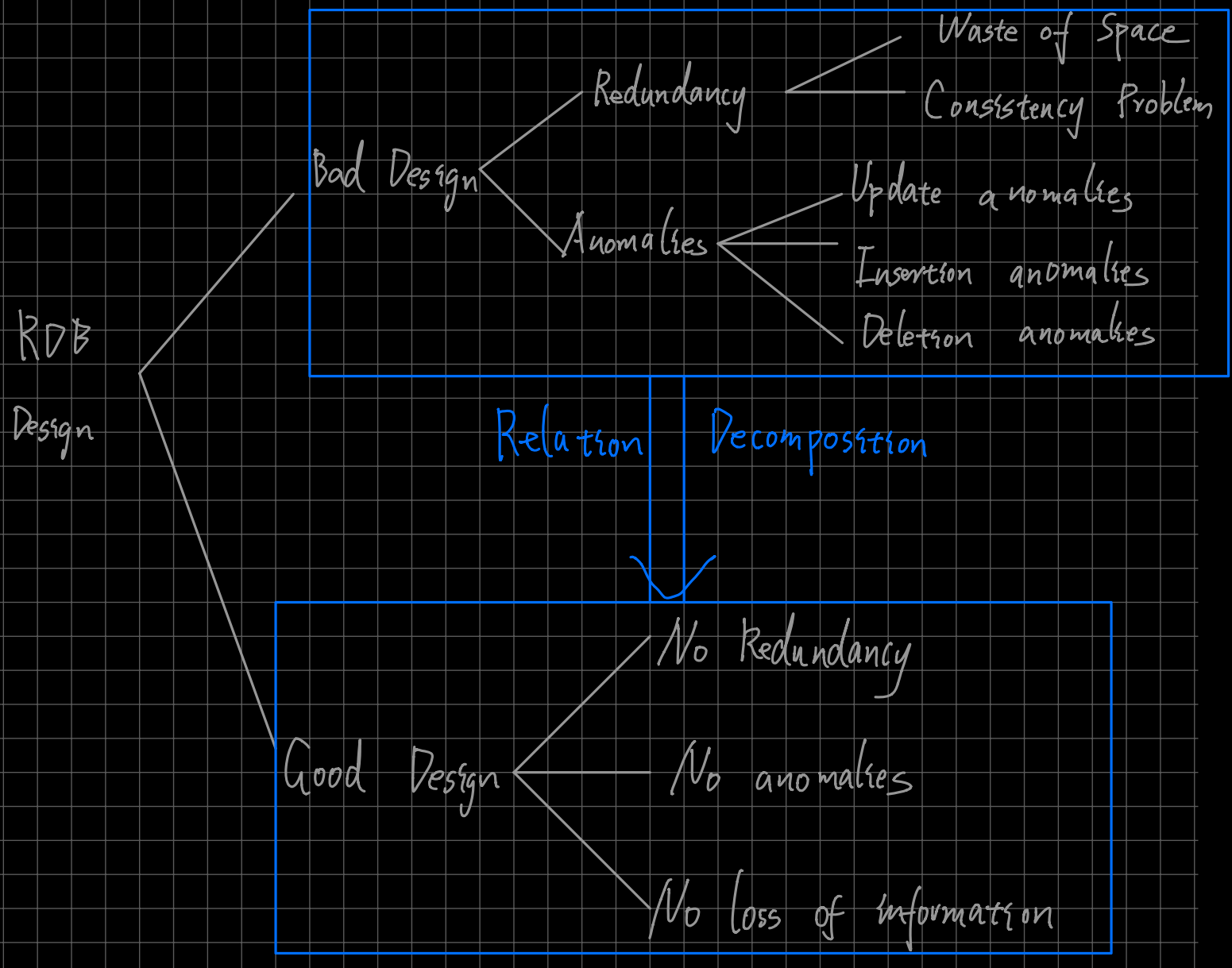
In this step we use Reflexsibility and Augumentation to eliminate some.

II. Check the Transitivity and remove the unecessary ones.

In this step we use Transitivity to eliminate some.

Thus, to get the $F_i$, we use all three methods in Amstrong's Axiom.

# Lecture 5 Normal Forms




How to do decomposition ⇒ Let the small relations after decomposing satisfies some conditions.

① in BCNF ② Can be reconstructed ⇓ using lossless join.

always ensures

BCNF ( Boyce - Codd Normal Form )

本质：在第三范式之上加上更严格约束，去除属性间不必要 FDs.

Definition: R is in BCNF iff

for every nontrivial FD: $X \to Y$, X is a superkey for R. ( $X^+ == R.$ attributes )

How to do BCNF decomposition ?

Steps:

I. Find a BCNF violation ($X \to Y$)

II Compute $X^+$.

II. Decompose R into :

① $X^+$

② $(R - X^+) \cup X$

# △ Advantages and Disadvantages of BCNF:

**Advantages:** Decomposition R into BCNF relations can always be done successfully, and it's a good design.

**Disadvantages:** The relations may be divided into too small relations in order to satisfy BCNF condition, so it lower the efficiency. (Each time you want to do a query you may need to combine a lot of small relations)

Boyce Code Normal Form:

Theme: One thing one place.

Reason: It requires all the left side of the FDs are superkey. Thus, only key can determine other attributes, which is equal to one thing one place.

thing 对象

Very similar when you put irrelevant objects/attributes in one relation, it will cause

# The Fourth Normal Form:

redundancy

redundancy will further cause consistency problems or anomalies.

Theme: One multi-value attribute one place object's feature