

# Design Theory of Relational DBs.

- What makes a good database?
  - Normalization
  - Avoid "anomalies"

## Functional Dependencies (FD)

- A FD is a constraint between two set of attributes of a relation.
- Given  $R$ , a set of attributes  $X$  in  $R$  is said to **functionally determine** another set of attributes  $Y$  in  $R$  ( $X \rightarrow Y$ ) iff 2 tuples have the same values of attributes  $X$  then they must have the same values for attributes  $Y$ .

We write them as:

$$\underline{A_1 \dots A_n} \rightarrow \underline{B_1 \dots B_m}$$

Attributes written as list.

## Example:

Title	Year	Length	Genre	Studio Name	Star Name
Star Wars	1977	124	Sci Fi	Fox	Carry Fisher
Star Wars	1977	124	Sci Fi	Fox	Mark Hamill
Star Wars	1977	124	Sci Fi	Fox	Harrison Ford
The Godfather	1972	175	Drama	Paramount	Robert Duvall
The Godfather	1972	175	Drama	Paramount	Marlon Brandon
Apocalypse Now	1979	153	War	Zoetrope	Marlon Brandon

I claim by design that

title, year  $\rightarrow$  length, genre, studioName

title, year  $\nrightarrow$  starName

## Superkey (SK)

A set of attributes  $\{A_1, \dots, A_n\}$  is a superkey of  $R$  iff  $A_1, \dots, A_n \rightarrow R$

## Candidate key (key)

A candidate key is a superkey that is minimal:

There is no proper subset  $C$  of  $\{A_1, \dots, A_n\}$  s.t.  $C \rightarrow R$

One candidate key becomes the Primary Key!

For our example:

- All attributes of  $R$  are always a SK.
- title, year, starname is a candidate

Key

title, year, starName  $\rightarrow R$ .

- Any superset of a candidate Key is a SK.

Reasoning about FD's.

- Given a relation  $R$  two sets of FDs  $A$  &  $B$  are equivalent if.

The set of instances of  $R$  that satisfy  $A$  is exactly the same that satisfy  $B$

- $A$  follows from  $B$  if every instance of  $R$  that satisfies  $B$  also satisfies  $A$ .
- $A$  &  $B$  are equivalent iff.  
 $A$  follows  $B$  and  $B$  follows  $A$

## Armstrong's Axioms (3.2 page 81)

Given relation  $R$  with subsets of attributes  
 $X, Y, Z \subseteq R$

Reflexivity (Trivial)

$Y \subseteq X$  then  $X \rightarrow Y$

Augmentation:

If  $X \rightarrow Y$  then  $XZ \rightarrow YZ$  for any  $Z$

Transitivity:

If  $X \rightarrow Y, Y \rightarrow Z$  then  $X \rightarrow Z$

Additional Rules.

They can be derived from axioms.

Union:

If  $X \rightarrow Y, X \rightarrow Z$  then  $X \rightarrow YZ$

Decomposition

If  $X \rightarrow YZ$  then  $X \rightarrow Y$  and  $X \rightarrow Z$

Ex :

Derive Union from axioms.

$$X \rightarrow Y, X \rightarrow Z$$

$$XZ \rightarrow YZ \quad \text{Augmentation}$$

$$\begin{array}{l} XX \rightarrow XZ \\ X \rightarrow XZ \end{array} \quad \left. \vphantom{\begin{array}{l} XX \rightarrow XZ \\ X \rightarrow XZ \end{array}} \right\} \text{Augmentation.}$$

$$\begin{array}{l} X \rightarrow XZ \rightarrow YZ \\ X \rightarrow YZ \quad \square \end{array} \quad \left. \vphantom{\begin{array}{l} X \rightarrow XZ \rightarrow YZ \\ X \rightarrow YZ \quad \square \end{array}} \right\} \text{Transitivity}$$

Derive decomposition

$$X \rightarrow YZ$$

$$\begin{array}{l} YZ \rightarrow Y \\ YZ \rightarrow Z \end{array} \quad \left. \vphantom{\begin{array}{l} YZ \rightarrow Y \\ YZ \rightarrow Z \end{array}} \right\} \text{Reflexivity.}$$

$$\begin{array}{l} X \rightarrow Y \\ X \rightarrow Z \quad \square \end{array} \quad \left. \vphantom{\begin{array}{l} X \rightarrow Y \\ X \rightarrow Z \quad \square \end{array}} \right\} \text{Transitivity.}$$

## Closure of attributes (3.2.4)

Given a relation  $R$  and a set  $f$  of FDs, what other FDs can be computed from a set of attributes of  $R$ ?

The closure of a set of attributes  $A_1 \dots A_n$  denoted  $\{A_1 \dots A_n\}^+$  is the set of attr. that can be derived from  $A_1 \dots A_n$  using  $f$ .

Hard to do and error prone via axioms!!

Alg:

1) Rewrite FDs in  $f$  in canonical form

2)  $X \leftarrow A_1 \dots A_n$

3) for each  $B_1 \dots B_m \rightarrow C$  in FDs  
if  $B_1 \dots B_m \subseteq X$  and  $C \notin X$   
add  $C$  to  $X$

4) Repeat (3) until  $X$  does not change

$X$  is  $\{A_1 \dots A_n\}^+$

Ex:

$R(A B C D E)$

$f = \left\{ \begin{array}{l} AB \rightarrow C \\ BC \rightarrow AD \\ D \rightarrow E \\ CF \rightarrow B \end{array} \right.$

① 

$AB \rightarrow C$	1
$BC \rightarrow A$	2
$BC \rightarrow D$	3
$D \rightarrow E$	4
$CF \rightarrow B$	5
	6

Compute  $\{AB\}^+$

$X \leftarrow AB$

First pass:

$X \leftarrow ABC$  fd 1  
 $X \leftarrow ABCD$  fd 3  
 $X \leftarrow \underline{ABCDE}$  fd 4

all attributes hence  $X$  will not change any more

$\{AB\}^+ = \{ABCDE\}$

$AB$  is a SK of  $R$ . Is it a candidate key?

Closure of attr. can help us find CKs of a relation: Compute  $\{A\}^+$  and  $\{B\}^+$

## Closure of set of FDs

Given a set  $f$  of FDs, its closure  $\{f\}^+$  is the set of all FDs derived from  $f$ .

Ex:

$$f = \left\{ \begin{array}{l} A \rightarrow B \\ B \rightarrow C \end{array} \right\} \quad A \rightarrow C \in f^+$$

Two sets  $A$  &  $B$  of FDs are equivalent iff.  $\{A\}^+ = \{B\}^+$

$$\left\{ \begin{array}{l} A \rightarrow B \\ B \rightarrow C \end{array} \right\}^+ = \left\{ \begin{array}{l} A \rightarrow B \\ B \rightarrow C \\ A \rightarrow C \end{array} \right\}^+$$

We can easily test if  $X \rightarrow Y \in f^+$

$X \rightarrow Y \in f^+$  iff  $Y \subseteq \{X\}^+$  using  $f$ .

Ex:

$$A \rightarrow C \in \left\{ \begin{array}{l} A \rightarrow B \\ B \rightarrow C \end{array} \right\}$$

$$\{A\}^+ = \{ABC\} \Rightarrow A \rightarrow ABC$$

$$\Rightarrow \begin{array}{l} A \rightarrow A \\ A \rightarrow B \\ A \rightarrow C \end{array}$$





## Basis of a relation

Given a relation  $R$  and FDs  $f$  we say that any set  $g$  s.t.  $f^+ = g^+$  is a **basis** of  $R$ .

## Minimal Basis of FDs. (3.2.7)

Any relation  $R$  has many equivalent set of FDs (many basis-es).

To avoid an explosion of FDs we usually use a minimal basis

A **minimal basis**  $B$  of a relation  $R$  is a basis of  $R$  s.t.

- 1) All FDs in  $B$  are in canonical form
- 2) If any FD is removed from  $B$ , the result is no longer a basis
- 3) If for any FD we remove one or more attr. from the left hand side the result is no longer a basis,

Ex:

$$A = \left\{ \begin{array}{l} A \rightarrow B \\ B \rightarrow A \\ B \rightarrow C \\ C \rightarrow B \end{array} \right\}$$

Given

is a minimal basis of

$$B = \left\{ \begin{array}{l} A \rightarrow B \\ A \rightarrow C \\ B \rightarrow A \\ B \rightarrow C \\ C \rightarrow A \\ C \rightarrow B \\ AB \rightarrow C \\ AC \rightarrow B \\ BC \rightarrow A \end{array} \right\}$$

$$\{B\}^+ = \{BAC\}$$

$$\{C\}^+ = \{CAB\}$$

We can generate.  $\left\{ \begin{array}{l} B \rightarrow C \\ C \rightarrow A \end{array} \right\}$

Hence  $A$  is a basis of  $B$  (and vice-versa)  
Is  $A$  minimal?

• Can we drop  $A \rightarrow B$ ?

$A \rightarrow B$  be generated from  $\left\{ \begin{array}{l} B \rightarrow A \\ C \rightarrow B \\ B \rightarrow C \end{array} \right\}$

$\{A\}^+ = \{A\}$  so no,  $A \rightarrow B$  cannot be removed.

- Can we drop  $B \rightarrow A$ ?

$B \rightarrow A$  be generated from  $\left\{ \begin{array}{l} A \rightarrow B \\ C \rightarrow B \\ B \rightarrow C \end{array} \right\}$

$\{B\}^+ = \{B, C\}$ , so no, it can not be removed.

- Repeat for  $C \rightarrow B$  and  $B \rightarrow C$ .

Yes, it is minimal.

~~XL~~

## Projection of FDs (3.2.8)

Given  $R$  and set  $F$  of FDs

the **projection of  $F$  on  $R_1 = \Pi_L R$**

is the set of FDs that follows from  $F$   
this involve only attributes in  $R_1$ .

Algorithm:

$T \leftarrow \emptyset$

for each subset  $X \in L$  compute  $\{X\}^+$

for every attribute  $A$  in  $\{X\}^+$

add  $X \rightarrow A$  to  $T$

iff  $A \in L$  and  
 $A \notin X$  (non-trivial)


Ex :  $R(ABCD)$   $F = \left\{ \begin{array}{l} A \rightarrow B \\ B \rightarrow C \\ C \rightarrow D \end{array} \right\}$

Compute FDs of  $\Pi_{ACD} R$

	closure
1) A C D	<del>A</del> <del>B</del> <del>C</del> <del>D</del>
A C	<del>A</del> <del>C</del> <del>B</del> D
A D	<del>A</del> <del>D</del> <del>B</del> C
A	<del>A</del> <del>B</del> <del>C</del> D
C D	<del>C</del> <del>D</del>
C	<del>C</del> D
D	<del>D</del>

3) Remove  
RHS att not  
in L.

4) Remove  
trivial FDs.

Result ::  $\left\{ \begin{array}{ll} AC \rightarrow D & A \rightarrow C \\ AD \rightarrow C & A \rightarrow D \\ & C \rightarrow D \end{array} \right\}$  

Is it a minimal basis? No

$\left\{ \begin{array}{l} A \rightarrow D \\ AC \rightarrow D \\ AD \rightarrow C \end{array} \right\}$  can be generated from  $\left\{ \begin{array}{l} A \rightarrow C \\ C \rightarrow D \end{array} \right\}$

Prove it!!

# Design of Relational DBs (3.3)

Title	Year	Length	Genre	Studio Name	Star Name
Star Wars	1977	124	Sci Fi	Fox	Carry Fisher
Star Wars	1977	124	Sci Fi	Fox	Mark Hamill
Star Wars	1977	124	Sci Fi	Fox	Harrison Ford
The Godfather	1972	175	Drama	Paramount	Robert Duvall
The Godfather	1972	175	Drama	Paramount	Marlon Brandon
Apocalypse Now	1979	153	War	Zoetrope	Marlon Brandon

FDs: title, year  $\rightarrow$  length, studio Name

## Anomalies

- Redundancy: Unnecessary repeated info
- Update anomalies: If we change one tuple we might have to change another: Ex: change Length of a movie.
- Deletion anomalies: If we delete a tuple we might lose other info:  
Ex: Remove M. Brandon from Ap. Now.

## Decomposing Relations

To deal with anomalies we decompose relations.

Given  $R(A_1 \dots A_n)$  a decomposition into  $S(B_1 \dots B_m)$  and  $T(C_1 \dots C_k)$  s.t.

$$1) \{A_1 \dots A_n\} = \{B_1 \dots B_m\} \cup \{C_1 \dots C_k\}$$

and

$$2) S = \pi_{B_1 \dots B_m} R \text{ and}$$
$$T = \pi_{C_1 \dots C_k} R$$

We can decompose Movies into

$S(\text{title, year, length, genre, studio Name})$

$T(\text{title, year, starName})$

title	Year	Length	Genre	Studio Name
Star Wars	1977	124	Sci Fi	Fox
The Godfather	1972	175	Drama	Paramount
Apocalypse Now	1953	153	War	Zoetrope

= S

Call this Movies2

Title	Year	StarName
Star Wars	1977	Carry Fisher
Star Wars	1977	Mark Hamill
Star Wars	1977	Harrison Ford
The Godfather	1972	Robert Duvall
The Godfather	1972	Marlon Brandon
Apocalypse Now	1979	Marlon Brandon

= T.

Call this Movies3

Good decompositions.

Given a relation R we want to decompose it into two relations S and T s.t.

1)  $R = S \bowtie T$  lossless join

2) The projection of FDs  $F_R$  of R into S ( $F_S$ ) and T ( $F_T$ )

satisfies:

$$\{F_S \cup F_T\}^+ = \{F_R\}^+$$

Dependency preserving.



# Boyce Codd Normal Form (BCNF)


A relation  $R$  is in BCNF iff  
for every non trivial FD

$$A_1 \dots A_n \rightarrow B_1 \dots B_m$$

$A_1 \dots A_n$  is a Superkey.

Ex:

Movies is not BCNF

title, year  $\rightarrow$  lenght, studioName  
is a not a SK of Movies 

If a relation  $R$  is not BCNF then  
decompose into relations  $R_1 \dots R_n$   
s.t  $R_1 \bowtie R_2 \dots \bowtie R_n = R$

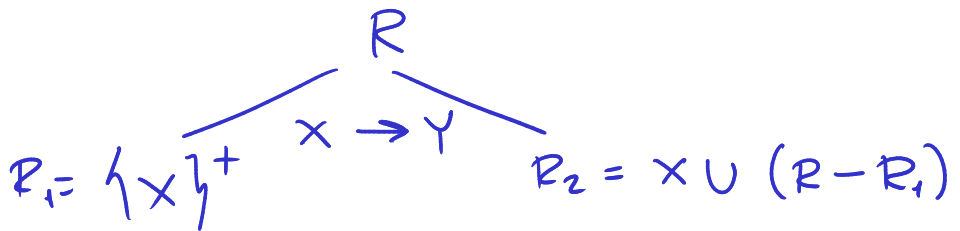
$\Rightarrow$  Loss-less join decomposition.

## Algorithm to decompose into BCNF relations

Given  $R$  and set  $F$  of FDs:

$R$  is not BCNF.

- 1) Choose one FD  $X \rightarrow Y$  not in BCNF
- 2). Decompose:



- 3) Compute FDs for  $R_1$  and  $R_2$   
(projection of FDs of  $R$  into  $R_1, R_2$ )
- 4) If  $R_1$  or  $R_2$  are not BCNF recursively decompose.

Guaranteed to be lossless join. but not FD preserving.

Any non BCNF relation has a BCNF loss-less join decomposition but not a BCNF FD preserving decomposition.

Ex:  $R(TCH)$   $H \rightarrow C$   
 $TC \rightarrow H$ .

$H \rightarrow C$  not BCNF.

$TCH$ .

$H \rightarrow C$

$\{H\}^+ =$

$H \cup \{TCH - C\}$

$HC \leftarrow \text{any 2 attr rel} \rightarrow HT$   
 is BCNF

FDs

FDs

H	C	H	C
H			AC
C			C

H	T	H	T
H			HT
T			T

$H \rightarrow C$

Decomposition  $R_1 = HC$   $FD_1 = \{H \rightarrow C\}$   
 $R_2 = HT$   $FD_2 = \emptyset$

Not FD preserving lost  $TC \rightarrow H$ .

## 3rd Normal Form (3NF)

If we cannot decompose a relation into BCNF relations that are FD preserving we are happy if they can be decompose into 3NF relations.

Any relation  $R$  with a set of FDs  $F$  has a 3NF decomposition that is loss-less join and FD preserving.

A relation  $R$  with FDs  $F$  is in 3NF if for every non trivial FD

$$A_1 \dots A_n \rightarrow B_1 \dots B_m$$

- it is a SK

or

- $C \in \{B_1 \dots B_m\}$  is either

$C \in \{A_1 \dots A_n\}$  or

$C$  is part of some candidate key.

Ex:  $R(ABCDE)$

$$F = \left\{ \begin{array}{l} AB \rightarrow C \\ C \rightarrow B \\ A \rightarrow D \end{array} \right\}$$

Is it 3NF?

$AB \rightarrow C$   $AB$  not a SK.

is  $C$  part of a CK?

Need to compute candidate keys of  $R$

• Heuristic:

$AE$  never on righthand side of FD

$\Rightarrow$  always part of a key.

Use closure of attr. to compute SKs.

all combination of attr.

closure.

$AEBCD$   
 $AEBC$   
 $AEBCD$   
 $AEBCD$   
 $AEBC$   
 $AEBCD$   
 $AEBC$   
 $AEBCD$   
 $AE$   
 $AE$

$AEBCD$   
 $AEBCD$   
 $AEBCD$   
 $AEBCD$   
 $AEBCD$   
 $AEBCD$   
 $AEBCD$   
 $AEBCD$   
 $AED$   
 $AED$

all SK.

minimal  $\Rightarrow$  Candidate keys.

$AEB$   
 $AEC$

Back to testing if  $R$  is 3NF.

$AB \rightarrow C$   $C$  is part of  $CK$ . ✓

$C \rightarrow B$   $C$  is not  $SK$ . but  
 $B$  is part of  $CK$ .

$A \rightarrow D$   $A$  is not  $SK$ .

$D$  is not part of  $CK$

$\Rightarrow R$  is not 3NF. 

Decomposition of a Relation into 3NF relations that is loss-less join and FD preserving: (Synthesis alg 3.5.2)

Given  $R$  with set of FDs  $F$

- 1) Find  $G$ , a minimal basis of  $F$
- 2) For each FD  $A_1 \dots A_n \rightarrow B$  in  $G$ .  
add a relation with schema  
 $A_1 \dots A_n B$  with FD  
 $A_1 \dots A_n \rightarrow B$
- 3) If **none** of the added relations in step 2 is a **SK** of  $R$  add another relation whose schema is a key of  $R$ .

Ex:

$R(ABCDE)$

$F = \left\{ \begin{array}{l} AB \rightarrow C \\ C \rightarrow B \\ A \rightarrow D \end{array} \right\}$   $F$  is minimal basis

$R_1 = ABC$  with  $AB \rightarrow C$

$R_2 = CB$  with  $C \rightarrow B$

$R_3 = AD$  with  $A \rightarrow D$

None of them contains a SK.

(see previous exercise, you can verify it by computing  $\{R_i\}^+$ )

We know its keys are  $AEB$  and  $AEC$

We need only one. So add

$R_4$  with no FDs.

Decomposition of  $R$  is  
 $R_1, R_2, R_3$  and  $R_4$  with  
corresponding FDs

