Designing Database

UML Data Modeling

How to represent data for application

1. Relational model (tables)
2. XML
3. Graphes
   (a) Entity-Relationship Model (E/R)
   (b) Unified Modeling Language (UML)

Both can be translated to relations automatically (or semi-automatically)

Unified Modeling Language (UML)

1. Classes
2. Associations
3. Association Classes
4. Subclasses
5. Composition & Aggregation

Classes

Name, attributes, methods

For data modeling: add primary key, delete methods

Unified Modeling Language (UML)

1. Classes
2. Associations
3. Association Classes
4. Subclasses
5. Composition & Aggregation

Associations

Relationships between objects of two classes

Multiplicity of Associations

Each object of class C_1 is related to at least m and at most n objects of class C_2

special
$m \ldots *$
$0 \ldots *$
$1 \ldots 1$ (default)

Unified Modeling Language (UML)

1. Classes
2. Associations
3. Association Classes
4. Subclasses
5. Composition & Aggregation

UML Data Modeling: Association Classes

Relationships between objects of two classes, with attributes on relationships

The sem set of information can be captured bu defferent schemas.

The same set of information can be captured into defferent schemas.

but some schemas are better than others

Database for students applying to French Universities

123 Marc lives in Nancy applies to Paris-Sorbonne, U-Lille and Lyon-I has bac $15/20$

sID	Name	Adress	Bac	University

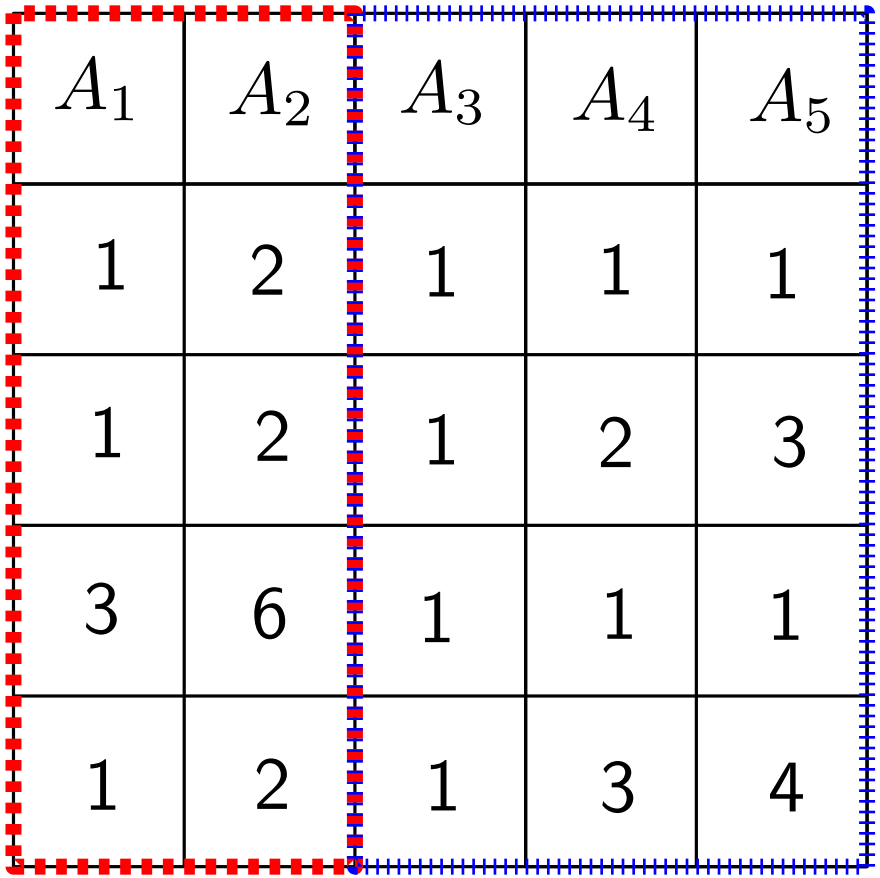
Problem on this schema (anomalies):

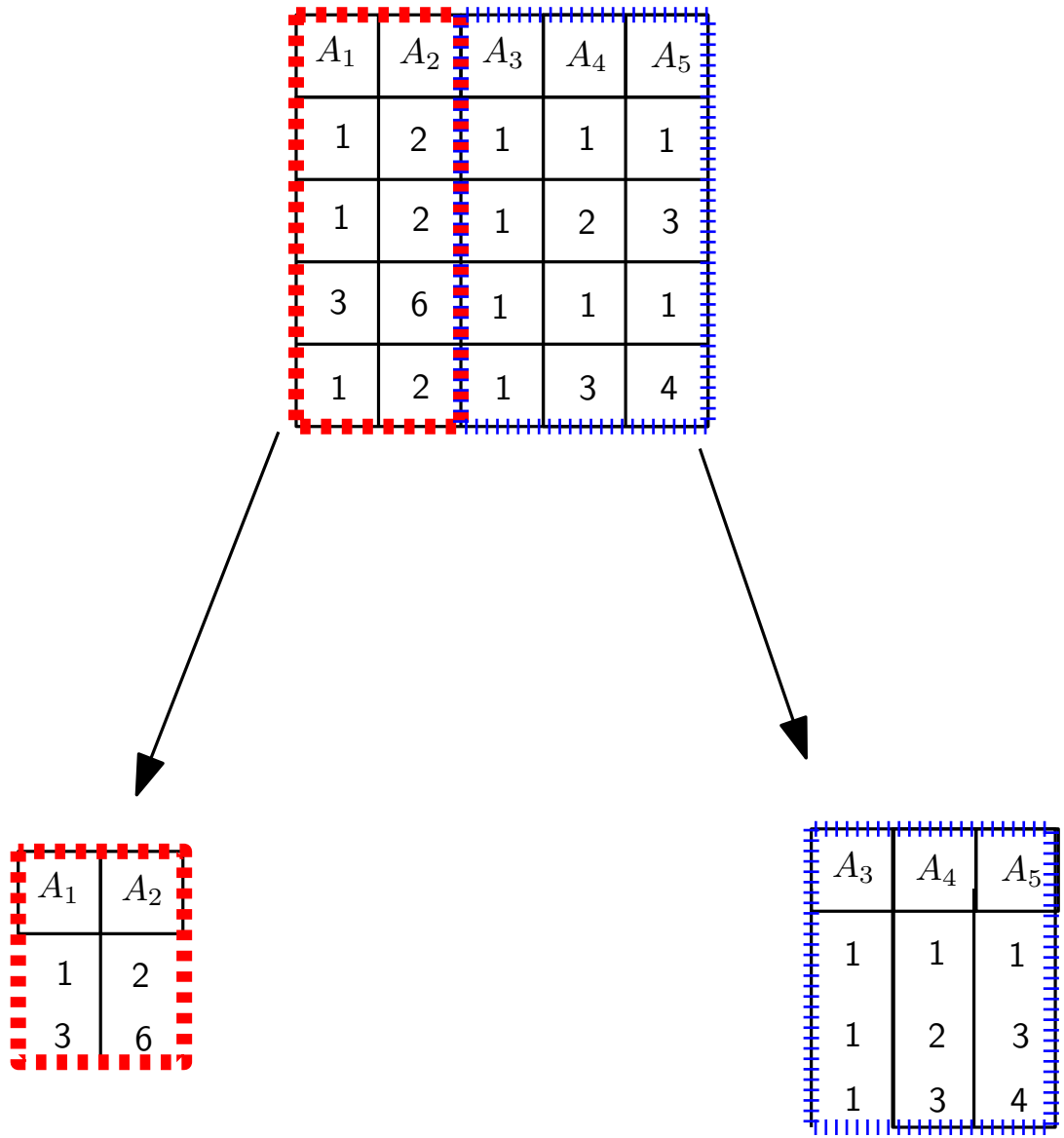
1. Redundancy
2. Update anomaly
3. Deletion anomaly

How to solve this problem?

A_1	A_2	A_3	A_4	A_5
1	2	1	1	1
1	2	1	2	3
3	6	1	1	1
1	2	1	3	4

How to solve this problem?




A_1 A_2 A_3 A_4 A_5
1 2 1 1 1
1 2 1 2 3
3 6 1 1 1
1 2 1 3 4
A_1 A_2
1 2
3 6
A_3 A_4 A_5
1 1 1
1 2 3
1 3 4

Functional Dependency

$A \rightarrow B$ we read it: B functionaly depends on A

A	B	C

Boyce-Codd normal form

A relation (table) is called Boyce-Codd normal fom if

for any $A_1, A_2, \ldots A_n \rightarrow B_1, B_2, \ldots B_m$

$A_1, A_2, \ldots A_n$ is a key

That is: $A_1, A_2, \ldots A_n$ determines the whole tuble

Student(S-id, Name, address, U-id, U-Name, U-city, Bac, Rating)

Bac $\rightarrow$ Rating

S-id $\rightarrow$ Name, adress, Bac

U-id $\rightarrow$ U-Name, U-city

U-Name, U-city $\rightarrow$ U-id

Student(S-id, Name, address, U-id, U-Name, U-city, Bac, Rating)

Bac $\rightarrow$ Rating

S-id $\rightarrow$ Name, adress, Bac

U-id $\rightarrow$ U-Name, U-city

U-Name, U-city $\rightarrow$ U-id

What is are the keys on this relation?

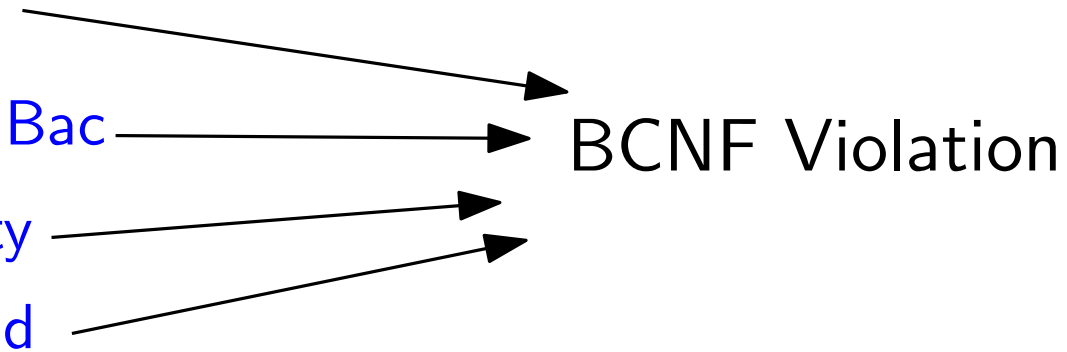
Student(S-id, Name, address, U-id, U-Name, U-city, Bac, Rating)

Bac $\rightarrow$ Rating

S-id $\rightarrow$ Name, adress, Bac

U-id $\rightarrow$ U-Name, U-city

U-Name, U-city $\rightarrow$ U-id



S-id , U-id $\rightarrow$ All others

What is are the keys on this relation?

Student(S-id, Name, address, U-id, U-Name, U-city, Bac, Rating)

Bac $\rightarrow$ Rating

S-id $\rightarrow$ Name, adress, Bac

U-id $\rightarrow$ U-Name, U-city

U-Name, U-city $\rightarrow$ U-id

Is this relation in Boyce-Codd normal form ?

Exercises

Consider the relation $R(A, B, C, D, E)$ and suppose we have the functional dependencies $A, B \rightarrow C$, $A, E \rightarrow D$, and $D \rightarrow B$. Which of the following attribute pairs is a key for R?

BCNF decomposition algorithm

Input: relation R + FDs for R

Output: decomposition of R into BCNF relations with lossless join

1. Compute keys for R
2. Repeat until all relations are in BCNF:
   (a) Pick any R with $A \rightarrow B$ that violates BCNF
   (b) Decompose R into R_1 (A, B) and R_2 (A, rest)
   (c) Compute FDs and keys for R_1 and R_2

Is BCNF always good?

Apply(S-id,U-id, hoppy)

Is BCNF always good?

Apply(S-id,U-id, hoppy)

1. Functional dependency ?
2. Keys?
3. BCNT
4. Is it a good design ?

Multivalued dependency

Relation $R(A, B, C)$

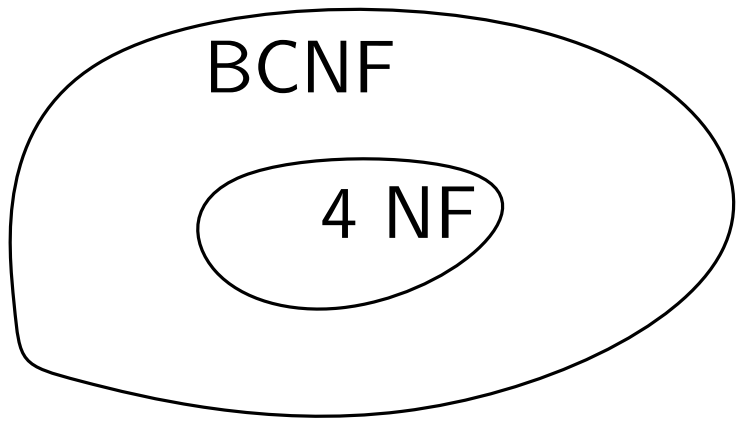
$A \twoheadrightarrow B$ if

for all L_1, L_2 in R with
$R_1[A] = R_2[A]$, then

There exists L_3 in R such that
$R_3[A] = R_1[A]$ and
$R_3[B] = R_1[B]$ and
$R_3[C] = R_2[C]$

4th Normal form

A realation is in 4th normal form, if for any $A \twoheadrightarrow B$, we have that A is a key



Apply(S-id,U-id, hoppy)

It is BNCF, but not 4 NF