Lecture 3:
Conceptual Database Design and Schema Design

April 12th, 2004

Agenda

- Project: questions?
- How to model data (E/R modeling)
- How to design a good schema (normalization).

Building an Application with a DBMS

- Requirements modeling (conceptual, pictures)
  - Decide what entities should be part of the application and how they should be linked.
- Schema design and implementation
  - Decide on a set of tables, attributes.
  - Define the tables in the database system.
  - Populate database (insert tuples).
- Write application programs using the DBMS
  - way easier now that the data management is taken care of.

Database Design

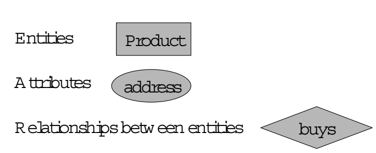
- Why do we need it?
  - Agree on structure of the database before deciding on a particular implementation.
- Consider issues such as:
  - What entities to model
  - How entities are related
  - What constraints exist in the domain
  - How to achieve good designs

Database Design Formalisms

1. Object Definition Language (ODL):
   - Closer in spirit to object-oriented models
   - I don't teach it anymore.
2. Entity/Relationship model (E/R):
   - More relational in nature.

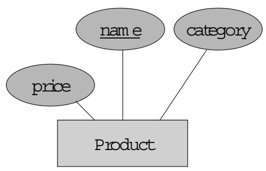
- Both can be translated (semi-automatically) to relational schemas
- ODL to OO-schema: direct transformation (C++ or Smalltalk based system).

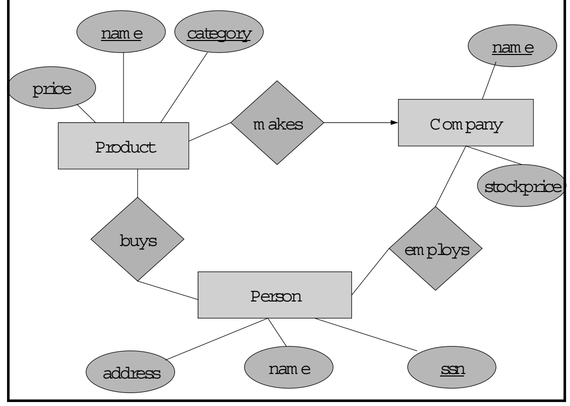
2. Entity / Relationship Diagrams



Keys in E/R Diagrams

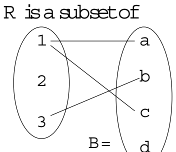
- Every entity set must have a key



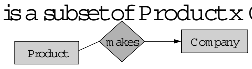


What is a Relation ?

- A mathematical definition:
  - if A, B are sets, then a relation R is a subset of A x B
- A={1,2,3}, B={a,b,c,d},
  R = {(1,a), (1,c), (3,b)}

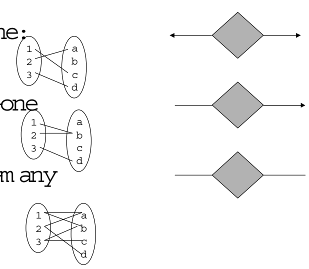


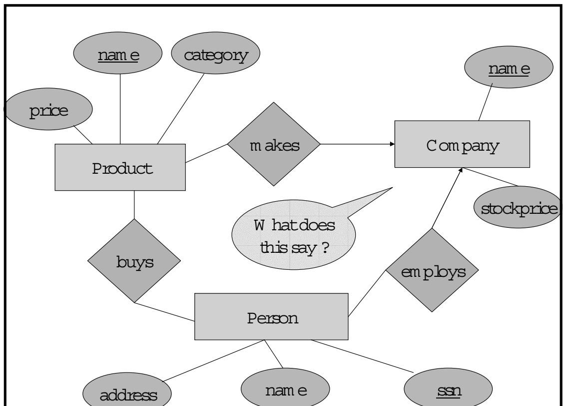
-makes is a subset of Product x Company:



Multiplicity of E/R Relations

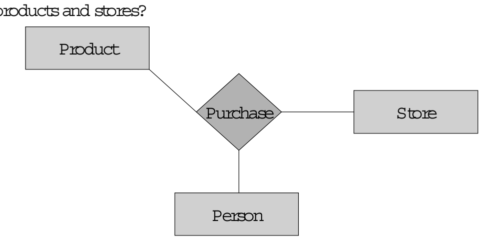
- one-one:
- many-one
- many-many





Multi-way Relationships

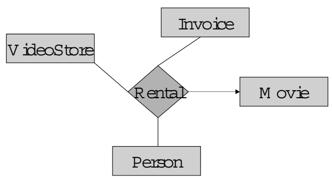
How do we model a purchase relationship between buyers, products and stores?



Can still model as a mathematical set (how ?)

Arrows in Multiway Relationships

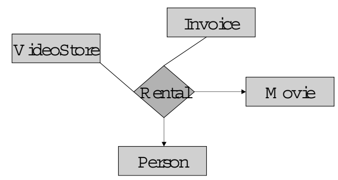
Q: what does the arrow mean ?



A: if I know the store, person, invoice, I know the movie too

Arrows in Multiway Relationships

Q: what do these arrow mean ?

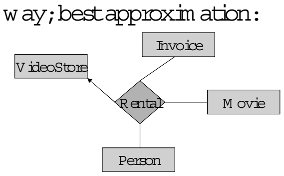


A: store, person, invoice determines movie and store, invoice, movie determines person

Arrows in Multiway Relationships

Q: how do I say: "invoice determines store" ?

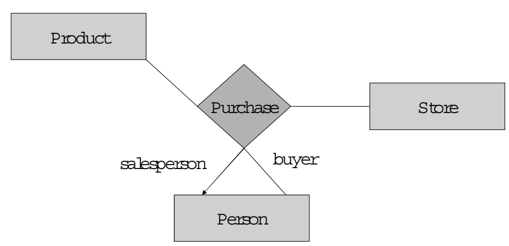
A: no good way; best approximation:



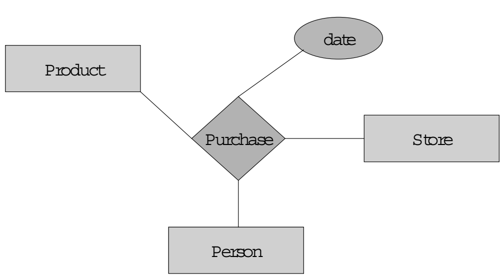
Q: Why is this incomplete ?

Roles in Relationships

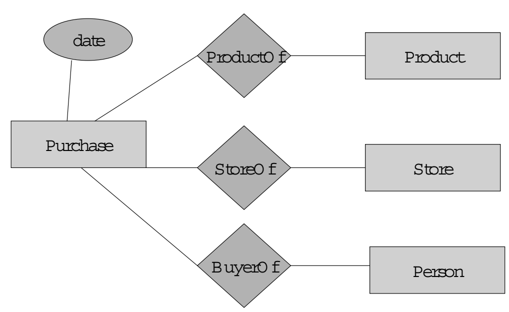
What if we need an entity set twice in one relationship?



Attributes on Relationships



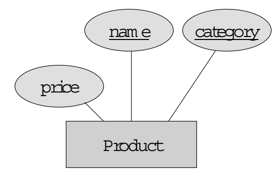
Converting Multi-way Relationships to Binary



From E/R Diagrams to Relational Schema

- Entity set relation
- Relationship relation

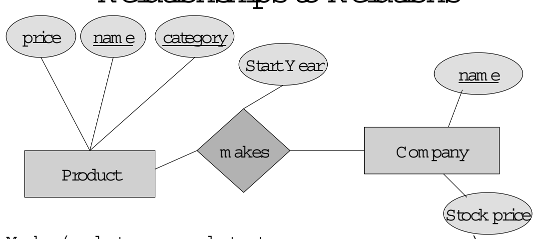
Entity Set to Relation



Product(name, category, price)

name	category	price
gizmo	gadgets	$19.99

Relationships to Relations

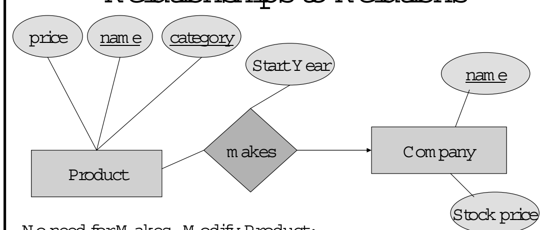


Makes(product-name, product-category, company-name, year)

Product-name	Product-Category	Company-name	Starting-year
gizmo	gadgets	gizmoWorks	1963

(watch out for attribute name conflicts)

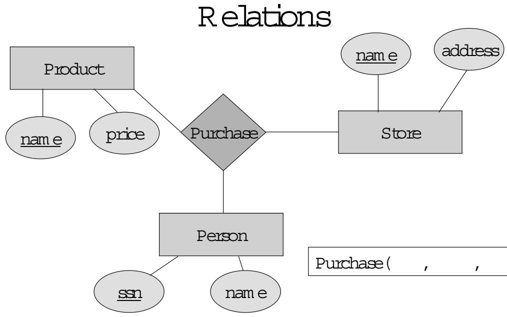
Relationships to Relations



No need for Makes. Modify Product:

name	category	price	StartYear	companyName
gizmo	gadgets	19.99	1963	gizmoWorks

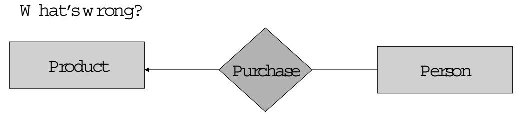
Multi-way Relationships to Relations

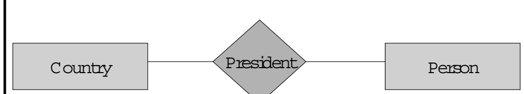


Purchase(, ,)

3. Design Principles

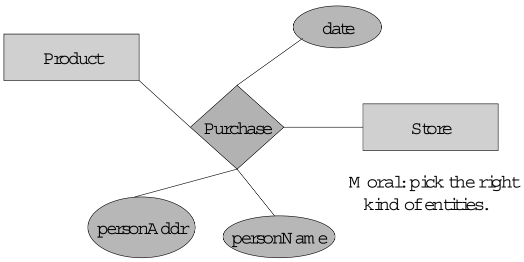
What's wrong?





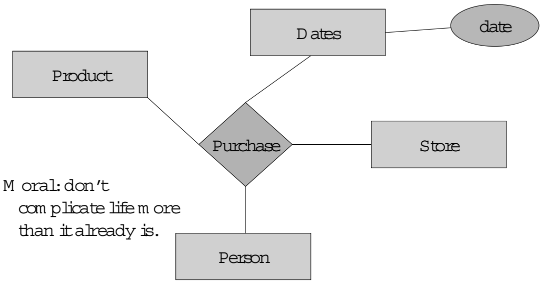
Moral: be faithful!

Design Principles: What's Wrong?



Moral: pick the right kind of entities.

Design Principles: What's Wrong?

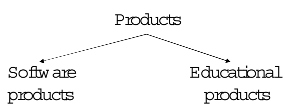


Moral: don't complicate life more than it already is.

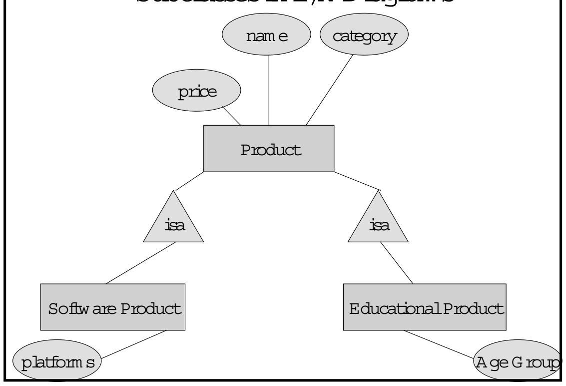
Modeling Subclasses

The world is inherently hierarchical. Some entities are special cases of others

- We need a notion of subclass.
- This is supported naturally in object-oriented formalisms.

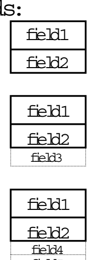


Subclasses in E/R Diagrams



Understanding Subclasses

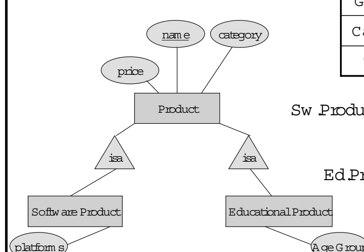
- Think in terms of records:
  - Product
  - SoftwareProduct
  - EducationalProduct



Subclasses to Relations

Product

Name	Price	Category
Gizmo	99	gadget
Camera	49	photo
Toy	39	gadget



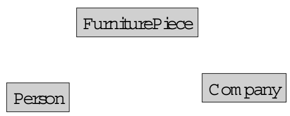
Sw Product

Name	platforms
Gizmo	unix

Ed.Product

Name	Age Group
Gizmo	todler
Toy	retired

Modeling UnionTypes With Subclasses

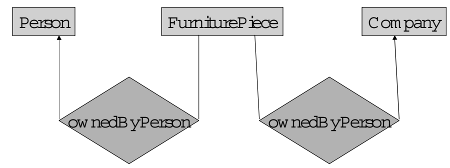


Say: each piece of furniture is owned either by a person, or by a company

Modeling Union Types with Subclasses

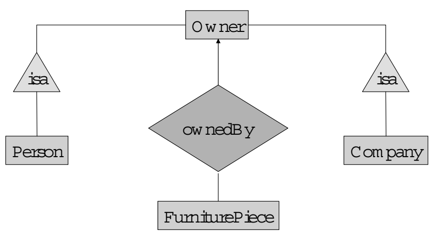
Say: each piece of furniture is owned either by a person, or by a company

Solution 1. Acceptable, imperfect (What's wrong ?)



Modeling Union Types with Subclasses

Solution 2: better, more laborious



Constraints in E/R Diagrams

Finding constraints is part of the modeling process.
Commonly used constraints:

Keys: social security number uniquely identifies a person.

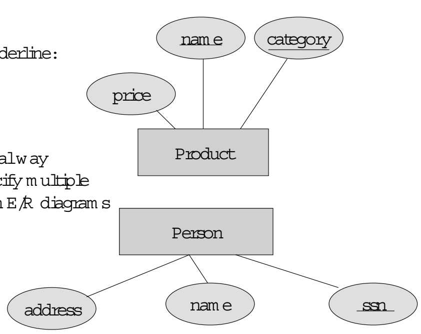
Single-value constraints: a person can have only one father.

Referential integrity constraints: if you work for a company, it must exist in the database.

Other constraints: peoples' ages are between 0 and 150.

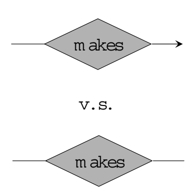
Keys in E/R Diagrams

Underline:

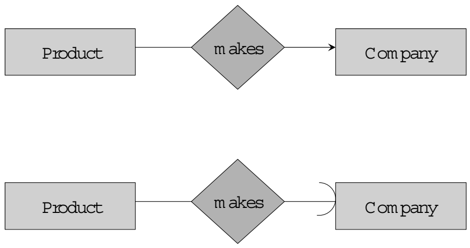


No formal way to specify multiple keys in E/R diagrams

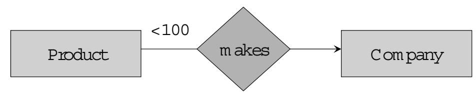
Single Value Constraints



Referential Integrity Constraints



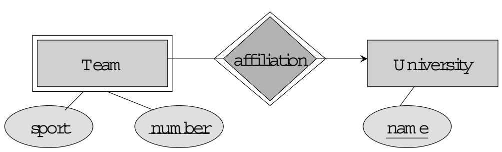
Other Constraints



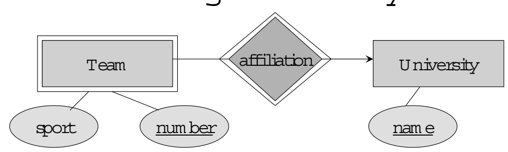
What does this mean ?

Weak Entity Sets

Entity sets are weak when their key comes from other classes to which they are related.

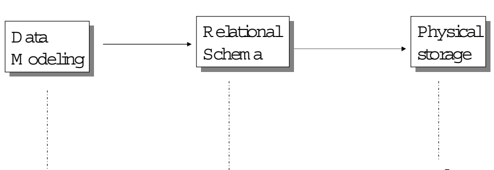


Handling Weak Entity Sets



Convert to a relational schema (in class)

The Relational Data Model



E/R diagrams

Tables:
- column names: attributes
- rows: tuples

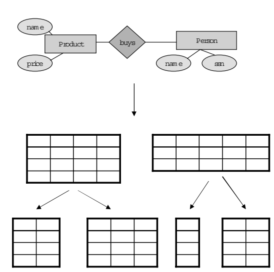
Complex file organization and index structures.

Relational Schema Design

Conceptual Model:

Relational Model: plus FD's

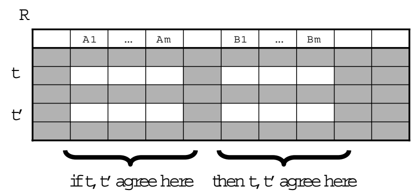
Normalization: Eliminates anomalies



Functional Dependencies

Definition: $A_1, \ldots, A_m$ $B_1, \ldots, B_n$ holds in R if:

$" t, t' . R, (t.A_1 = t'.A_1 \ \ldots \ t.A_m = t'.A_m \Rightarrow t.B_1 = t'.B_1 \ \ldots \ t.B_m = t'.B_m)$



Important Point!

- Functional dependencies are part of the schema!
- They constrain the possible legal data instances.
- At any point in time, the actual database may satisfy additional FD's.

Examples

EmpID	Name	Phone	Position
E0045	Smith	1234	Clerk
E1847	John	9876	Salesrep
E1111	Smith	9876	Salesrep
E9999	Mary	1234	Lawyer

- EmpID → Name, Phone, Position
- Position → Phone
- but Phone ↛ Position

Formal definition of a key

- A key is a set of attributes $A_1, \ldots, A_n$ s.t. for any other attribute B, $A_1, \ldots, A_n$ B
- A minimal key is a set of attributes which is a key and for which no subset is a key
- Note: book calls them superkey and key

Examples of Keys

- Product(name, price, category, color)
  name, category   price
  category   color

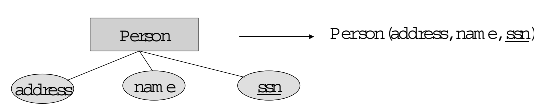
  Keys are: {name, category} and all supersets
- Enrollment(student, address, course, room, time)
  student   address
  room, time   course
  student, course   room, time

  Keys are: [in class]

Finding the Keys of a Relation

Given a relation constructed from an E/R diagram, what is its key?

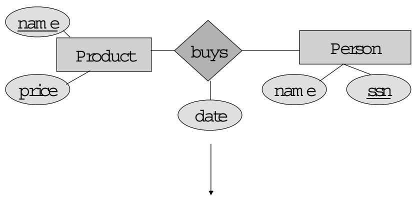
Rules:
1. If the relation comes from an entity set, the key of the relation is the set of attributes which is the key of the entity set.



Finding the Keys

Rules:

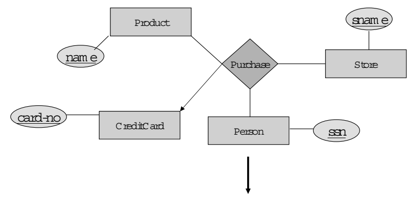
2. If the relation comes from a many-many relationship, the key of the relation is the set of all attribute keys in the relations corresponding to the entity sets



buys(name, ssn, date)

Finding the Keys

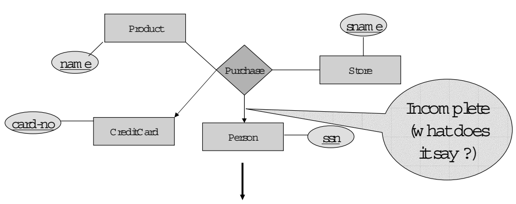
Except: if there is an arrow from the relationship to E, then we don't need the key of E as part of the relation key.



Purchase(name, sname, ssn, card-no)

Expressing Dependencies

Say: "the CreditCard determines the Person"



Purchase(name, sname, ssn, card-no)

card-no name

Finding the Keys

More rules in the book – please read !

Relational Schema Design (or Logical Design)

Main idea:

- Start with some relational schema
- Find out its FD's
  - Important also to look at inferred FD's.
- Use them to design a better relational schema

Inference Rules for FD's

$A_1, A_2, \dots A_n \rightarrow B_1, B_2, \dots B_m$

Is equivalent to

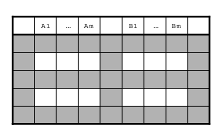
$A_1, A_2, \dots A_n \rightarrow B_1$

$A_1, A_2, \dots A_n \rightarrow B_2$

…

$A_1, A_2, \dots A_n \rightarrow B_m$

Splitting rule
and
Combing rule



Inference Rules for FD's (continued)

$A_1, A_2, \ldots A_n \longrightarrow A_i$ Trivial Rule

where i = 1, 2, …, n

Why?

	A1	…	Am	

Inference Rules for FD's (continued)

Transitive Closure Rule

If $A_1, A_2, \ldots A_n \longrightarrow B_1, B_2 \ldots, B_m$

and $B_1, B_2, \ldots B_m \longrightarrow C_1, C_2 \ldots, C_p$

then $A_1, A_2, \ldots A_n \longrightarrow C_1, C_2 \ldots, C_p$

Why?

	A1	…	Am		B1	…	Bm		C_1	…	C_p	

- Enrollment(student, major, course, room, time)

  student major

  major, course room

  course time

  What else can we infer? [in class]

Closure of a set of Attributes

Given a set of attributes {A1,… ,An} and a set of dependencies S.
Problem: find all attributes B such that:
any relation which satisfies S also satisfies:
A1,… ,An → B

The closure of {A1,… ,An}, denoted $\{A1, \ldots, An\}^+$,
is the set of all such attributes B

The closure tells us everything we can infer from A1,… ,An.

Closure Algorithm

Start with X = {A1,… ,An}.

Repeat until X doesn't change do:

if $B_1, B_2, \ldots B_n \longrightarrow C$ is in S, and
$B_1, B_2, \ldots B_n$ are all in X, and
C is not in X

then

add C to X.

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

Example

A B → C
A D → E
B → D
A F → B

Closure of {A,B}: X = {A,B, }

Closure of {A,F}: X = {A,F, }

Why Is the Algorithm Correct ?

- Show the following by induction:
  - For every B in X:
    - A1,… ,An → B
- Initially X = {A1,… ,An} -- holds
- Induction step: B1,… ,Bm in X
  - Implies A1,… ,An B1,… ,Bm
  - We also have B1,… ,Bm → C
  - By transitivity we have A1,… ,An C
- This shows that the algorithm is sound; need to show it is complete

Relational Schema Design (or Logical Design)

Main idea:

- Start with some relational schema
- Find out its FD's
- Use them to design a better relational schema

Relational Schema Design

Recall set attributes (persons with several phones):

Name	SSN	PhoneNumber	City
Fred	123-45-6789	206-555-1234	Seattle
Fred	123-45-6789	206-555-6543	Seattle
Joe	987-65-4321	908-555-2121	Westfield
Joe	987-65-4321	908-555-1234	Westfield

SSN Name, City, but not SSN PhoneNumber

Anomalies:

- Redundancy = repeat data
- Update anomalies = Fred moves to "Bellvue"
- Deletion anomalies = Fred drops all phone numbers: what is his city ?

Relation Decomposition

Break the relation into two:

Name	SSN	City
Fred	123-45-6789	Seattle
Joe	987-65-4321	Westfield

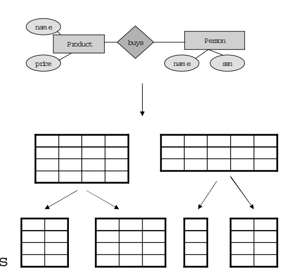
SSN	PhoneNumber
123-45-6789	206-555-1234
123-45-6789	206-555-6543
987-65-4321	908-555-2121
987-65-4321	908-555-1234

Relational Schema Design

Conceptual Model:

Relational Model: plus FD's

Normalization: Eliminates anomalies



Decompositions in General

$R(A_1, \ldots, A_n)$

Create two relations R1(B1, …, Bm) and R2(C1, …, Cp)

such that: B1, …, Bm ∪ C1, …, Cp = A1, …, An

and:

R_1 = projection of R on $B_1, \ldots, B_m$
R_2 = projection of R on $C_1, \ldots, C_p$

Incorrect Decomposition

- Sometimes it is incorrect:

Name	Price	Category
Gizmo	19.99	Gadget
OneClick	24.99	Camera
DoubleClick	29.99	Camera

Decompose on : Name, Category and Price, Category

Incorrect Decomposition

Name	Category
Gizmo	Gadget
OneClick	Camera
DoubleClick	Camera

Price	Category
19.99	Gadget
24.99	Camera
29.99	Camera

When we put it back:

Cannot recover information

Name	Price	Category
Gizmo	19.99	Gadget
OneClick	24.99	Camera
OneClick	29.99	Camera
DoubleClick	24.99	Camera
DoubleClick	29.99	Camera

Normal Forms

First Normal Form = all attributes are atomic

Second Normal Form (2NF) = old and obsolete

Third Normal Form (3NF) = this lecture

Boyce Codd Normal Form (BCNF) = this lecture

Others...

Boyce-Codd Normal Form

A simple condition for removing anomalies from relations:

A relation R is in BCNF if:

Whenever there is a nontrivial dependency $A_1, \ldots, A_n$ B in R , $\{A_1, \ldots, A_n\}$ is a key for R

In English (though a bit vague):

Whenever a set of attributes of R is determining another attribute, should determine all the attributes of R.

Example

Name	SSN	PhoneNumber	City
Fred	123-45-6789	206-555-1234	Seattle
Fred	123-45-6789	206-555-6543	Seattle
Joe	987-65-4321	908-555-2121	Westfield
Joe	987-65-4321	908-555-1234	Westfield

What are the dependencies?
SSN Name, City
What are the keys?
{SSN, PhoneNumber}
Is it in BCNF?

Decompose it into BCNF

Name	SSN	City
Fred	123-45-6789	Seattle
Joe	987-65-4321	Westfield

SSN Name, City

SSN	PhoneNumber
123-45-6789	206-555-1234
123-45-6789	206-555-6543
987-65-4321	908-555-2121
987-65-4321	908-555-1234

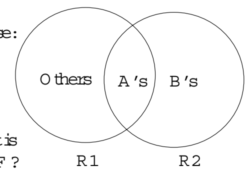
Summary of BCNF Decomposition

Find a dependency that violates the BCNF condition:

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

Heuristics: choose $B_1, B_2, \ldots B_m$ "as large as possible"

Decompose:



Is there a 2-attribute relation that is not in BCNF ?

Continue until there are no BCNF violations left.

Example Decomposition

Person(name, SSN, age, hairColor, phoneNumber)

SSN name, age

age hairColor

Decompose in BCNF (in class):

Step 1: find all keys

Step 2: now decompose

Other Example

- R(A,B,C,D) A → B, B → C
- Key: A,D
- Violations of BCNF: A → B, A → C, A → BC
- Pick A → BC: split into R1(A,BC) R2(A,D)
- What happens if we pick A → B first?

Correct Decompositions

A decomposition is lossless if we can recover:

R(A,B,C)

↓ Decompose

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

↓ Recover

R'(A,B,C) should be the same as R(A,B,C)

R' is in general larger than R. Must ensure R' = R

Correct Decompositions

- Given R(A,B,C) s.t. A → B, the decomposition into R1(A,B), R2(A,C) is lossless

3NF: A Problem with BCNF

Unit	Company	Product

FD's: Unit → Company; Company, Product → Unit
So, there is a BCNF violation, and we decompose.

Unit	Company

Unit → Company

Unit	Product

No FDs

So What's the Problem?

Unit	Company
Galaga99	UW
Bingo	UW

Unit	Product
Galaga99	databases
Bingo	databases

No problem so far. All local FD's are satisfied.
Let's put all the data back into a single table again:

Unit	Company	Product
Galaga99	UW	databases
Bingo	UW	databases

Violates the dependency: company, product -> unit!

Solution: 3rd Normal Form (3NF)

A simple condition for removing anomalies from relations:

> A relation R is in 3rd normal form if :
>
> Whenever there is a nontrivial dependency $A_1, A_2, ..., A_n \rightarrow B$ for R , then $\{A_1, A_2, ..., A_n\}$ a super-key for R,
> or B is part of a key.