The Relational Model

Objectives

- Relational Model
- Keys

The Relational Model

- Introduced in E.F. Codd's 1970 paper "A relational Model of the Data for Large Shared Data Banks"
- Suatu model data yang meletakkan /menggambarkan data dalam bentuk relasi (tabel)

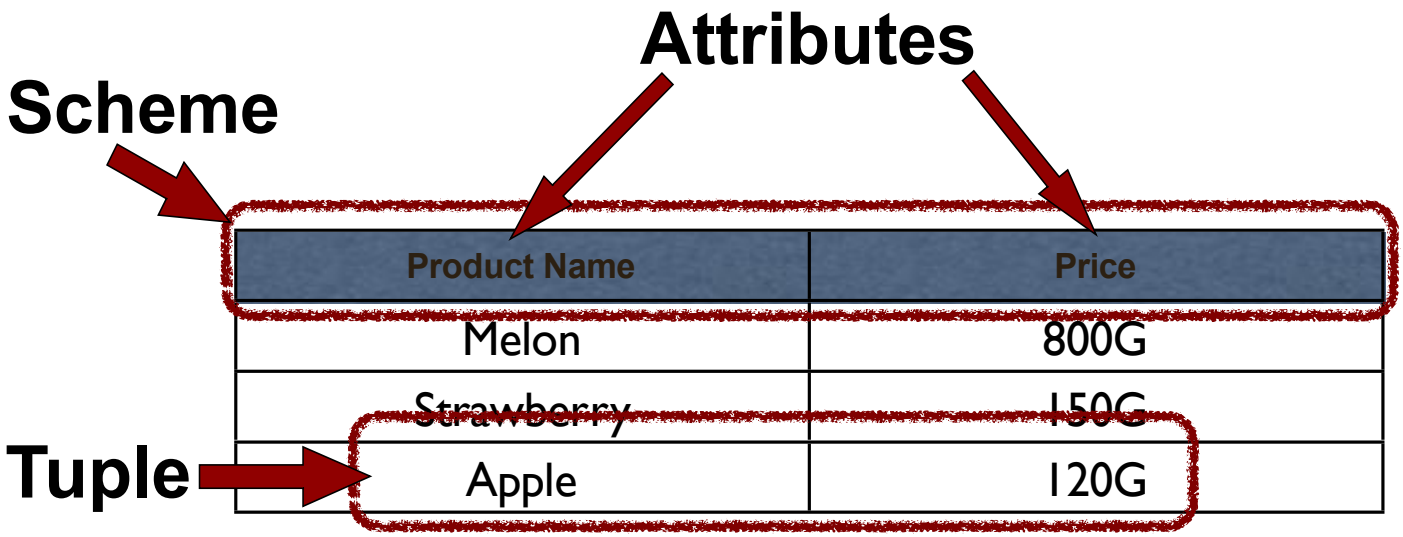
The Relational Model

- 3 main components
  - Data Structure (how data is represented)
    - Data-data diorganisasikan dalam bentuk table berdimensi dua dan yang terdiri atas baris dan kolom
  - Data integrity (what data is allowed)
    - Penentuan aturan-aturan bisnis dalam perusahaan yang diterapkan dalam database dengan tujuan menjaga integritas dan konsistensi data ketika dimanipulasi
  - Data manipulation (what you can do with the system)
    - Berkaitan dengan operasi untuk memanipulasi data (menambah, mengubah dan menghapus data)

Relational Data Structure

- Data is stored in **RELATIONS** (tables)
- Each relation has a **SCHEME** (heading)
- The scheme defines the relation's **ATTRIBUTES** (columns)
- Data takes the form of **TUPLES** (rows)

Relational Data Structure



The Database Terms

More formally:

- A **scheme** is a set of attributes
- A tuple assigns a value to each attribute in its scheme
- A relation is a set of tuples with the same scheme

Naming Conventions

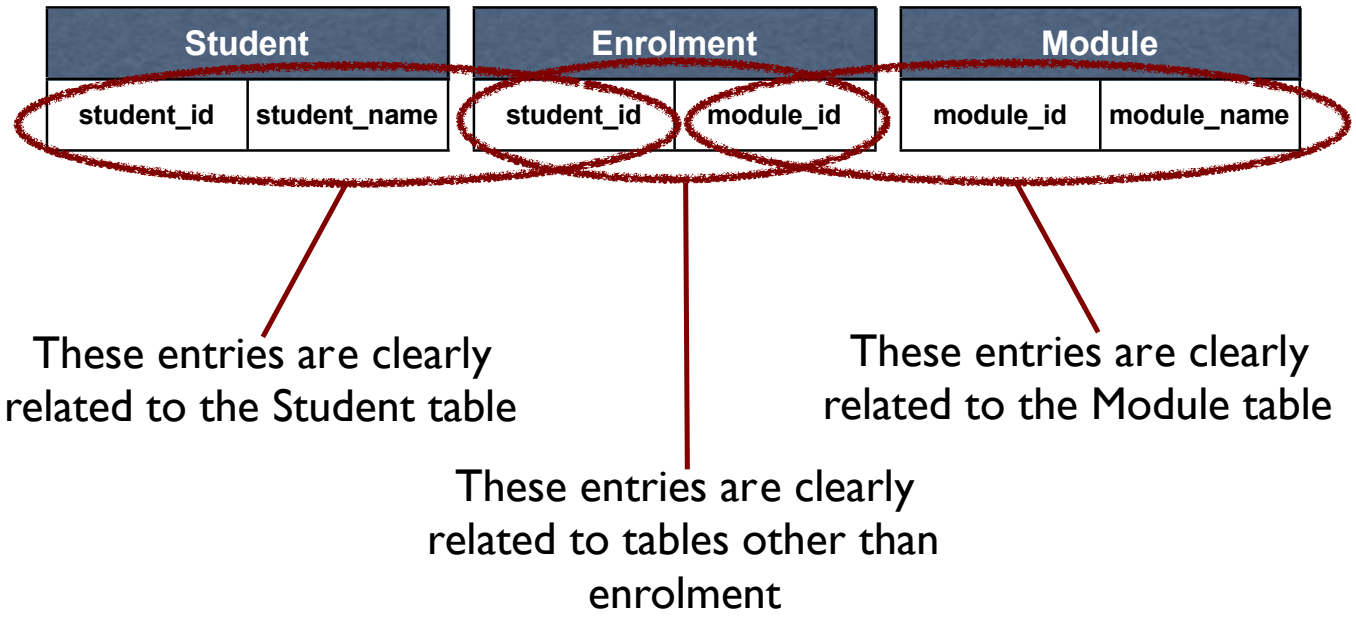
- Consistent naming conventions can help remind you of the databases structure
  - short lower case - separating words with an underscore
  - assign each table a unique prefix
- This facilitates the use of foreign keys (e.g. product_id)

Example

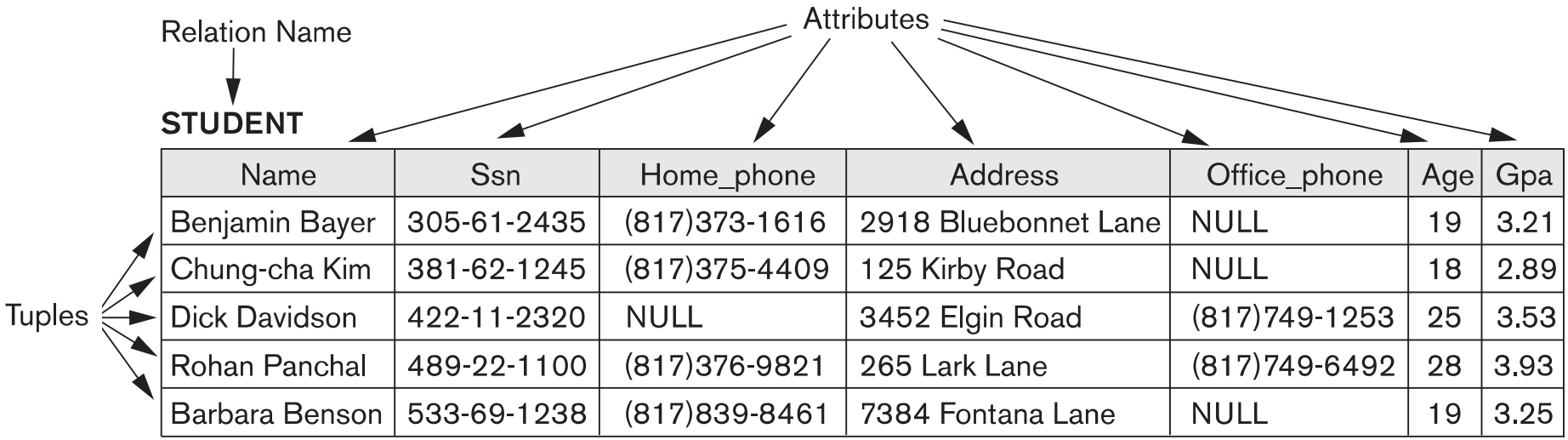
product		
product_id	product_name	product_price
101	Melon	800G
102	Strawberry	150G
103	Appe	120G

Example

- Consistent three tables - Student, Enrolment and Module



Example



Name	Ssn	Home_phone	Address	Office_phone	Age	Gpa
Benjamin Bayer	305-61-2435	(817)373-1616	2918 Bluebonnet Lane	NULL	19	3.21
Chung-cha Kim	381-62-1245	(817)375-4409	125 Kirby Road	NULL	18	2.89
Dick Davidson	422-11-2320	NULL	3452 Elgin Road	(817)749-1253	25	3.53
Rohan Panchal	489-22-1100	(817)376-9821	265 Lark Lane	(817)749-6492	28	3.93
Barbara Benson	533-69-1238	(817)839-8461	7384 Fontana Lane	NULL	19	3.25

Properties of Relations

- The relation has a name that is distinct from all other relation names in the relational schema;
- each cell of the relation contains exactly one atomic (single) value;
- each attribute has a distinct name;
- the values of an attribute are all from the same domain;

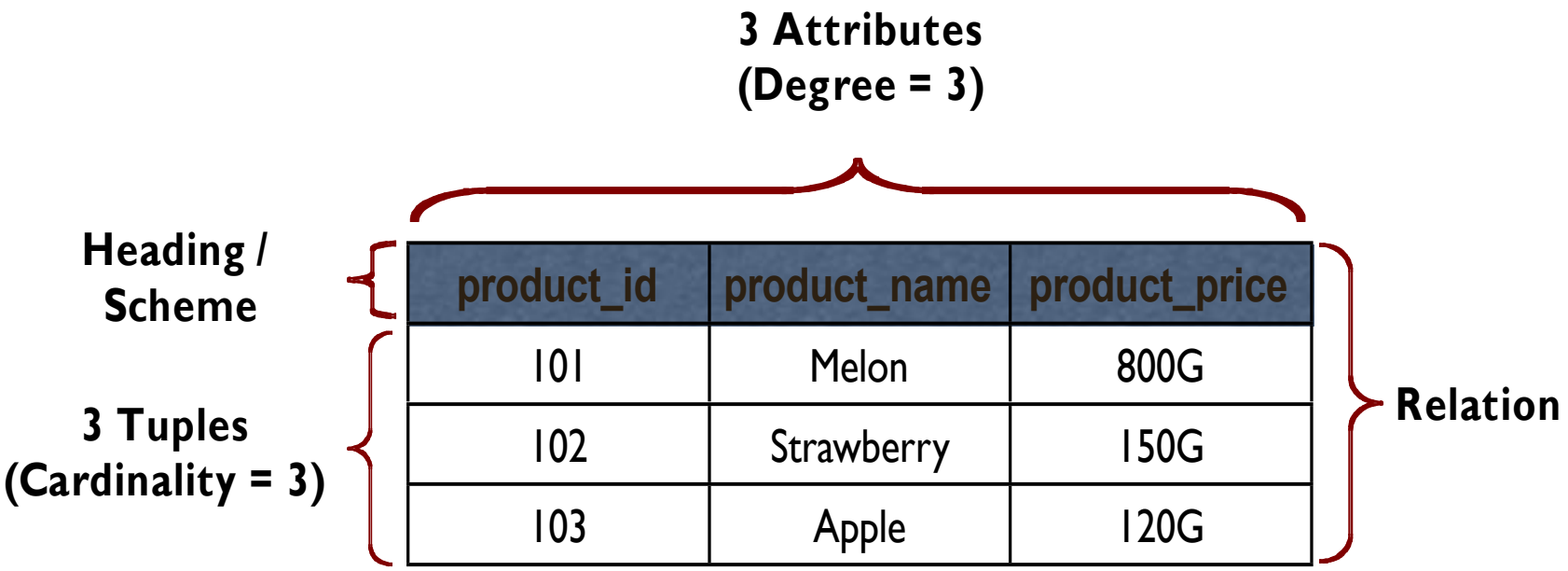
Properties of Relations

- each tuple is distinct; there are no duplicate tuples;
- the order of attributes has no significance;
- the order of tuples has no significance, theoretically. (However, in practice, the order may affect the efficiency of accessing tuples.)

Properties of Relations

- Since relations are sets of tuples, the tuples of a relation are *unique* and *unordered*
- The number of tuples in a relation is called the ***cardinality*** of the relation
- Since schemes are sets of attributes, the attributes of a relation are *unique* and *unordered*
- The number of attributes in a relation is called the relation's ***degree***

A Relation (table)



Relational Data Integrity

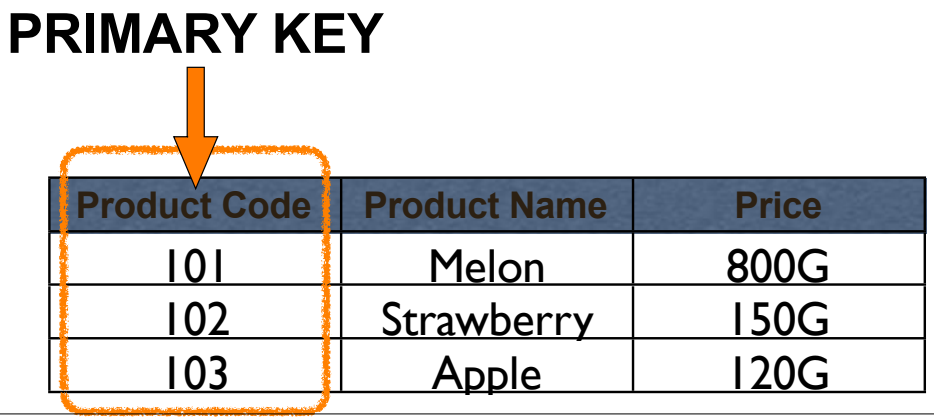
- Data integrity controls what data can be in a relation. This relates to
  - **Domains** restrict the possible values of attributes
  - **Candidate and Primary Keys** identify tuples within a relation
  - **Foreign Keys** link relations to each other

Attributes and Domains

- A **domain** is given for each attribute
- The domain lists the possible values for that attribute
- Each tuple contains a value for each attribute from the domain of the attribute
- Examples
  - An ‘age’ might have to come from the set of integers between 0 and 150
  - A ‘department’ might come from a given list of strings
  - A ‘notes’ field might allow any string at all

Keys

- CANDIDATE KEYs are unique fields
- One candidate key is usually declared to be the PRIMARY KEY - which is a field that serves to identify data
- A FOREIGN KEY is a key in another table



Candidate Keys

- Candidate Keys
  - A set of attributes in a relation is called a Candidate Key if and only if:
    - The values of the set of attributes uniquely identify a tuple (*uniqueness*)
    - No proper subset of the set has the uniqueness property (*minimality*)

Candidate Keys

- Candidate keys are:
  - {Nim} and {Nama, Tanggal_Lahir}
- Not unique:
  - {Nama} and {Tanggal_Lahir}
- Not minimal:
  - {Nim, Nama} {Nim, Tanggal_Lahir} and {Nim, Nama, Tanggal_Lahir}

Nim	Nama	Tanggal_Lahir
55	Ridwan Gani	20/03/2001
56	Slamet	03/12/2000
57	Fika Damayanti	17/08/2002
58	Rahmanda	06/06/2001
59	Slamet	04/05/2000

Primary Keys

- One Candidate Key is usually chosen to be used to identify tuples in a relation
- This is the **Primary Key**
- Often a special ID attribute is used as the primary key

Primary Keys

- Either {Nim} or {Nama, Tanggal_Lahir} could be used as the primary key
- Nim is a better primary key since it is a single column, and it will always be unique

Nim	Nama	Tanggal_Lahir
55	Ridwan Gani	20/03/2001
56	Slamet	03/12/2000
57	Fika Damayanti	17/08/2002
58	Rahmanda	06/06/2001
59	Slamet	04/05/2000

NULLs and Primary Keys

- Missing information can be represented using NULLs
- A NULL represents a missing or unknown value
- *Entity Integrity* : a Primary Key cannot contain NULLs

NULLs and Primary Keys

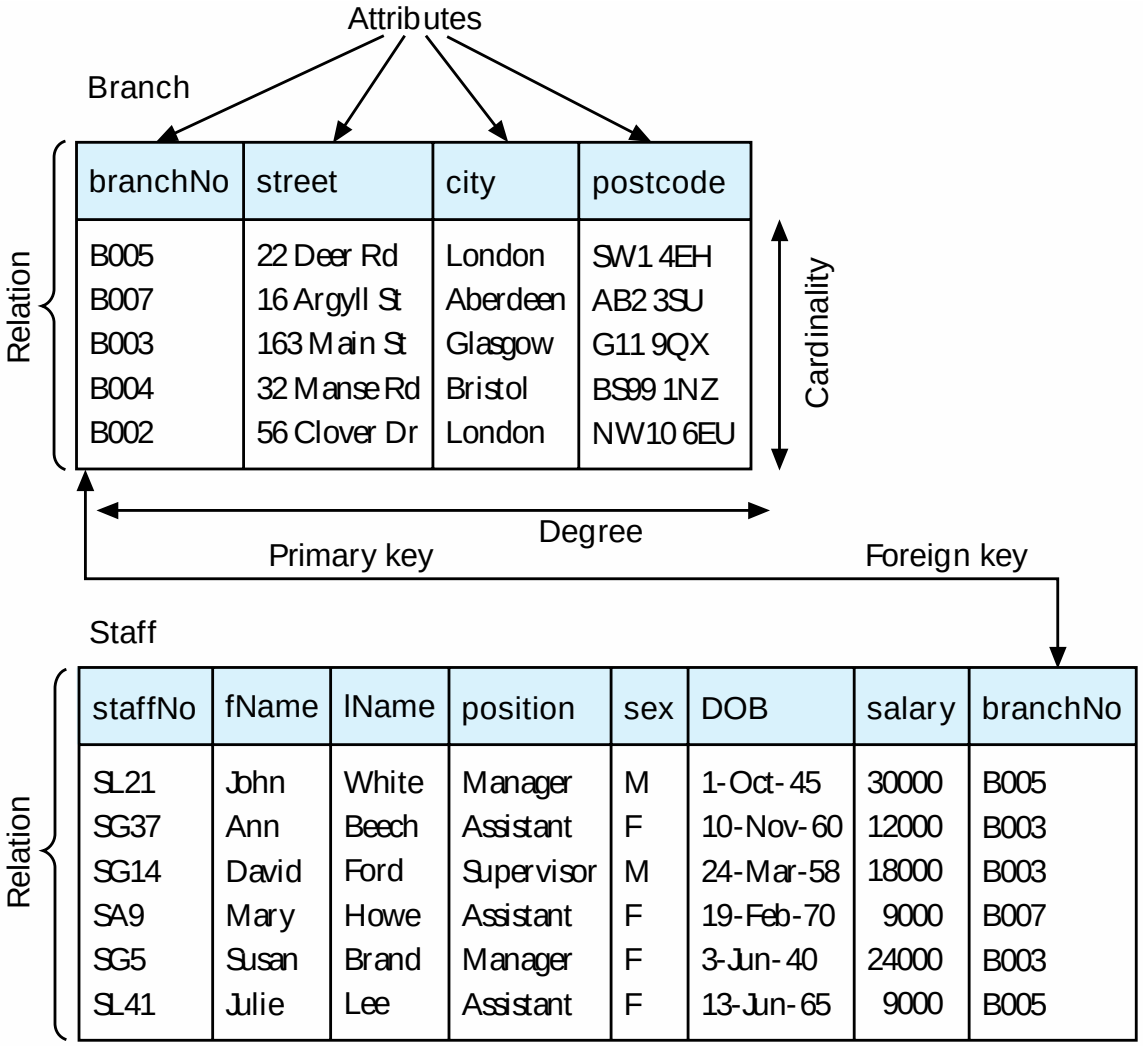
- NULLs can cause problems
  - a computer has no way of knowing whether a value is unknown or missing
  - this means that NULLs are intrinsically uncertain
  - uncertainty is difficult to compute

ID	Fname	Mname	Lname
s139	John	Arthur	Smith
s140	Mary	NULL	Jones
s141	John	NULL	Brown
s142	Jane	Anne	Smith

Foreign Keys

- Foreign Keys are used to link data in two relations.
- A set of attributes in the first (**Referencing**) relation is a Foreign Key if its value always matches a Candidate Key value in the second (**Referenced**) relation
- This is called **Referential Integrity**

Foreign Keys Example 1



Foreign Keys Example 2

department

dept_id	dept_name
13	Marketing
14	Accounts
15	Personnel

employee

empl_id	empl_name	dept_id
15	John Smith	13
16	Mary Brown	14
17	Mark Jones	13
18	Jane Smith	NULL

- dept_id is a candidate key for department
- dept_id is a foreign key for employee - each dept_id value is either NULL or matches an entry in the department relation.

Foreign Keys Example 3

student		
stud_id	stud_first	stud_last
S139	John	Smith
S140	Mary	Smith
S141	John	Brown
S142	Jane	Jones

enrolment		
stud_first	stud_last	en_module
John	Smith	G64DBS
John	Smith	G54ADS
Mary	Smith	G51HOC
John	Brown	G64PMM

- {stud_first, stud_last} is a candidate key for student - no entries have the same value for both stud_first and stud_last
- {stud_first, stud_last} is a Foreign Key in enrolment - each stud_first,stud_last pair matches exactly one entry in the Student relation

Referential Integrity

- When relations are updated, referential integrity can be violated
- This usually occurs when a referenced tuple is updated or deleted
- There are a number
  - RESTRICT - stop the user from doing it
  - CASCADE - let the changes flow on
  - NULLIFY - make values NULL
  - Triggers - actions defined by the user

Referential Integrity - Example

- What happens if:
  - Marketing's dept_id is changed to 16 in department?
  - The entry for Accounts is deleted from Department?

department

dept_id	dept_name
13	Marketing
14	Accounts
15	Personnel

employee

empl_id	empl_name	dept_id
15	John Smith	13
16	Mary Brown	14
17	Mark Jones	13
18	Jane Smith	NULL

RESTRICT

- RESTRICT stops any action that violates integrity
  - You cannot update or delete Marketing or Accounts
  - You can change Personnel as it is not referenced

department

dept_id	dept_name
13	Marketing
14	Accounts
15	Personnel

employee

empl_id	empl_name	dept_id
15	John Smith	13
16	Mary Brown	14
17	Mark Jones	13
18	Jane Smith	NULL

CASCADE

- CASCADE allows the changes made to flow through
  - If Marketing's dept_id is changed to 16 in department, then the dept_ids for John Smith and Mark Jones also change
  - If Accounts is deleted then so is Mary Brown

department

dept_id	dept_name
~~13~~ 16	Marketing
~~14~~	~~Accounts~~
15	Personnel

employee

empl_id	empl_name	dept_id
15	John Smith	~~13~~ 16
~~16~~	~~Mary Brown~~	~~14~~
17	Mark Jones	~~13~~ 16
18	Jane Smith	NULL

NULLIFY

- NULLIFY sets problem values to NULL
  - If Marketing's dept_id changes then John Smith's and Mark Jones' dept_ids are set to NULL
  - If Accounts is deleted, Mary Brown's dept_id becomes NULL

department

dept_id	dept_name
~~13~~ 16	Marketing
~~14~~	~~Accounts~~
15	Personnel

employee

empl_id	empl_name	dept_id
15	John Smith	~~13~~ NULL
16	Mary Brown	~~14~~ NULL
17	Mark Jones	~~13~~ NULL
18	Jane Smith	NULL