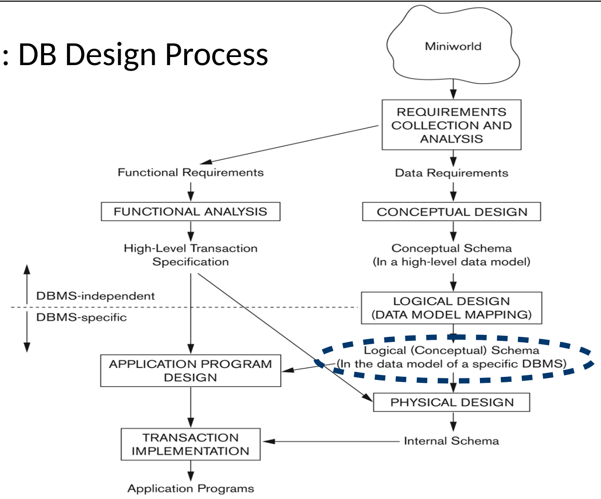
Database Technology

Topic 2: Relational Databases

Olaf Hartig
olaf.hartig@liu.se

LINKÖPING UNIVERSITY

Recall: DB Design Process

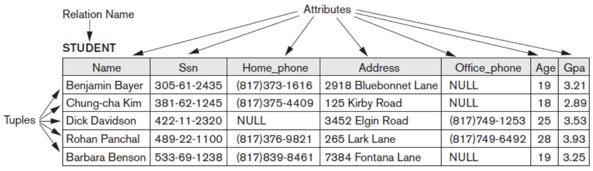

Miniworld
REQUIREMENTS COLLECTION AND ANALYSIS
Functional Requirements
Data Requirements
FUNCTIONAL ANALYSIS
CONCEPTUAL DESIGN
High-Level Transaction Specification
Conceptual Schema (In a high-level data model)
DBMS-independent
DBMS-specific
LOGICAL DESIGN (DATA MODEL MAPPING)
Logical (Conceptual) Schema (In the data model of a specific DBMS)
APPLICATION PROGRAM DESIGN
PHYSICAL DESIGN
TRANSACTION IMPLEMENTATION
Internal Schema
Application Programs

Relational Data Model

LINKÖPING UNIVERSITY

Relational Model Concepts

- Relational database: represent data as a collection of *relations*
  - Think of a relation as a table of values


Relation Name
Attributes
Tuples

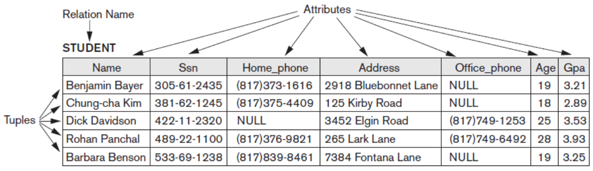
STUDENT

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

- Each row (*tuple*) represents a record of related data values
  - Facts that typically correspond to a real-world entity or relationship
- Each column (*attribute*) holds a corresponding value for each row
  - Columns associated with a data type (*domain*)
  - Each column header: *attribute name*

Relational Model Concepts (cont'd)

- Relational database: represent data as a collection of *relations*
  - Think of a relation as a table of values


Relation Name
Attributes
Tuples

STUDENT

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

- **Schema** describes the relation
  - Relation name, attribute names and domains
  - Integrity constraints
- **Instance** (also called **state**) denotes the *current* contents of the relation
  - *Set* of tuples

Domains

- **Domain** is a set of *atomic* values
  - { 0, 1, 2, … }
  - { Jo Smith, Dana Jones, Ashley Wong, Y. K. Lee, … }
- **Atomic**: Each value indivisible
- Domains specified by **data type** rather than by enumeration
  - Integer, string, date, real, etc.
  - Can be specified by format
    - e.g., *(ddd)ddd-dddd* for phone numbers
      (where *d* represents a digit)

Schemas and Attributes

- **Relation schema**
  - A relation name *R* and a list of attributes *A1*, *A2*, ... , *An*
  - Denoted by *R*(*A1*, *A2*, ..., *An*)
- **Attribute** *Ai*
  - Name of a role in the relation schema *R*
  - Associated with a domain **dom(*Ai*)**
  - Attribute names do not repeat within a relation schema, but domains can repeat
- **Degree** (or **arity**) of a relation
  - Number of attributes *n* in its relation schema

NULL Values

- Each domain may be augmented with a special value called NULL
  - Represent the values of attributes that may be unknown or may not apply to a tuple
  - If an attribute of a tuple is NULL, we cannot make any assumption about the value for that attribute (for that tuple)
- Interpretations for NULL values
  - Nothing is known about the value
  - Value exists but is (currently) not available
  - Value undefined (i.e., attribute does not apply to this tuple)
- For instance, Ashley's telephone number is NULL could mean
  - Ashley doesn't have a phone
  - Ashley has a phone but we don't know the number (perhaps withheld)
  - Ashley has a phone that has no number

Quiz

- A relation schema consists of:

A) relation name, attribute names and domains, and tuples;

or

B) relation name, attribute names and domains, and restrictions;

or

C) relation name, tuples, and NULL values.

Quiz

- A relation schema consists of:

A) relation name, attribute names and domains, and tuples;

or

B) relation name, attribute names and domains, and ~~restrictions;~~
integrity constraints

or

C) relation name, tuples, and NULL values.

Integrity Constraints

What are Integrity Constraints?

- **Constraints** are restrictions on the permitted values in a DB state
  - Derived from the rules in the miniworld that the DB represents

1. **Inherent model-based constraints** (also called **implicit constraints**)
   - Inherent in the data model, enforced by DBMS
   - e.g., duplicate tuples are not allowed in a relation
2. **Schema-based constraints** (also called **explicit constraints**)
   - Can be expressed in schemas of the data model, enforced by DBMS
   - e.g., films have only one director
   - Our focus here
3. **Application-based** (also **semantic constraints** or **business rules**)
   - Not directly expressed in schemas
   - Expressed and enforced by application program
   - e.g., this year's salary increase can be no more than last year's

Uniqueness Constraints

- Let *R* be a relation and *K* be a (sub)set of attributes of *R*
- If we specify the uniqueness constraint for K, then for any pair of tuples in *R*, the tuples must have a different value for at least one of the attributes in *K*
- Uniqueness must hold in all valid instances of *R*
- Uniqueness serves as a constraint on updates

Superkeys and Candidate Keys

- A set *K* of attributes of *R* is called a *superkey* of R if it has the

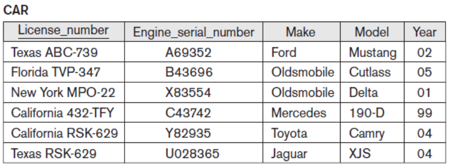
  Uniqueness property: no two distinct tuples have the same values across all attributes in *K* (i.e., we may define a uniqueness constraint for *K*)

- *K* is called a *candidate key* of *R* if, additionally, it also has the

  Minimality property: no proper subset of *K* has the uniqueness property

Primary Key

- There may be *more than one* candidate key in a relation
- **Primary key**: a particular candidate key is *chosen* as the primary
  - Diagrammatically, underline its attribute(s)
  - Tuples cannot have NULL for any primary key attribute
- Other candidate keys are designated as **unique**
  - Non-NULL values cannot repeat, but values may be NULL



CAR

License_number	Engine_serial_number	Make	Model	Year
Texas ABC-739	A69352	Ford	Mustang	02
Florida TVP-347	B43696	Oldsmobile	Cutlass	05
New York MPO-22	X83554	Oldsmobile	Delta	01
California 432-TFY	C43742	Mercedes	190-D	99
California RSK-629	Y82935	Toyota	Camry	04
Texas RSK-629	U028365	Jaguar	XJS	04

Figure 3.4
The CAR relation, with two candidate keys: License_number and Engine_serial_number.

Other Schema-Based Integrity Constraints

- **Entity integrity constraint**: No primary key value can be NULL
- **Domain constraint**: declared by specifying the datatype of attributes
- **Referential integrity constraint**
  - Specified between two relations
  - Allows tuples in one relation to *refer to* tuples in another
  - Maintains consistency among tuples in two relations
  - **Foreign key** rules:
    - Let *PK* be the primary key in a relation *R1* (i.e., set of attributes in its relational schema declared to be primary key)
    - Let *FK* be a set of attributes for another relation *R2*
    - The attribute(s) *FK* have the same domain(s) as the attribute(s) *PK*
    - Value of *FK* in a tuple *t2* of the current state of *R2* either occurs as a value of *PK* for some tuple *t*1 in the current state of *R1* or it is NULL

Diagramming Referential Constraints

- Show each relational schema
  - Underline primary key attributes in each
- Directed arc from each foreign key to the relation it references

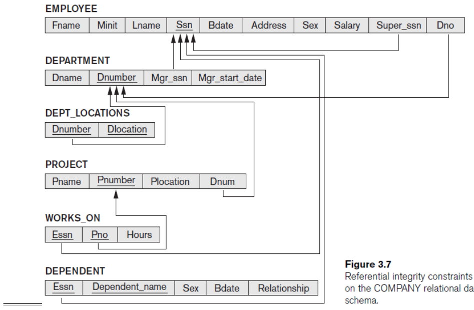


Figure 3.7
Referential integrity constraints displayed on the COMPANY relational database schema.

Quiz

- Consider the following two relations

Instructor

ID	Name	Office
4	Jennifer	B308
35	Paul	B311
12	Kim	E112

Course

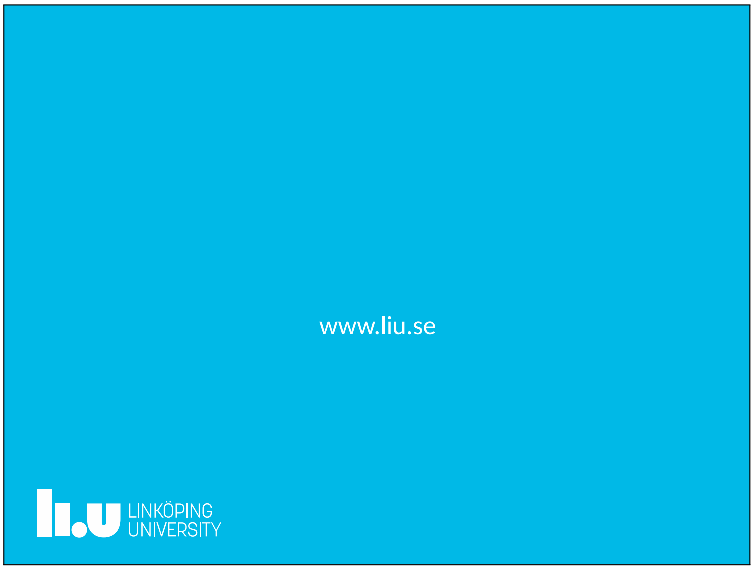
CourseID	Year	Instructor
cid444	2012	35
cid598	2013	4
cid444	2013	35

- Which of the following statements are correct and which are wrong?

  (a) We can insert a new *Course* tuple (cid598,2017,2).

  (b) We can modify the two cid444 *Course* tuples by changing their *Instructor* value to 12.

  (c) We can modify the cid598 *Course* tuple by changing its *CourseID* value to cid444.



www.liu.se

LINKÖPING
UNIVERSITY