

# Fundamentals/ICY: Databases

2013/14

***Week 4: Friday***

*John Barnden*

*Professor of Artificial Intelligence*

*School of Computer Science*

*University of Birmingham, UK*

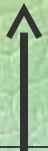


**Reminder**



# Typical Approach to Phone Numbers

NAME	<i>PHONE</i>	EMPLOYER	AGE
Chopples	0121-414-3816	E12345	37
Blurp	01600-719975	E54321	21
Rumpel	07970-852657	E12345	88



*(There should  
really be a FIRST  
NAME as well)*



But the following is *possible* ...

NAME	<i>PHONE ID</i>	EMPLOYER	AGE
Chopples	<i>ABC123</i>	E12345	37
Blurp	<i>ABC137</i>	E54321	21
Rumpel	<i>DEF678</i>	E12345	88

↑  
*There should  
really be a FIRST  
NAME as well*

<i>PHONE ID</i>	AREA CODE	BODY
<i>ABC123</i>	0121	414-3816
<i>ABC137</i>	01600	719975
<i>DEF101</i>	0121	440-5677
<i>DEF678</i>	07970	852657



**New**



# Some Operations on Individual Tables

- ◆ Creating a new empty table of a particular “shape” (mainly, particular column names and value-types for the columns)
- ◆ Changing the “shape” of an existing table (e.g., adding/deleting a column, or changing the type of a column)
- ◆ Adding a row or rows to a table
- ◆ Deleting a row or rows (question: how identified?)
- ◆ Updating values in an individual cell (column specified by name; but how identify the row?)



# More Operations on Individual Tables

- ◆ Retrieving values from an individual cell; doing calculations on them
- ◆ Retrieving the values in the cells in some or all columns for some or all rows
- ◆ Calculating statistics concerning values in particular columns across all rows, a subset of rows, or several subsets of rows (count, max, min, average, standard deviation, ...)
- ◆ Ordering rows in different ways in displays of a table.



# Operations on Coordinated Tables

- ◆ Need to be able to combine data from related tables in a variety of ways. *E.g.:*
  - Join tables together in various ways
  - Select things from one table on the basis of information in others
- ◆ Need to ensure consistency between related tables. *E.g.:*
  - Deletion of something in one table may require deletions from or other modifications to other tables.



# **ENTITIES, RELATIONSHIPS & ATTRIBUTES**

*(Introduction)*



# Entities

◆ Basically, entities are just things of the “important types” that we judged above to merit tables. So we had *entity types* such as:

- People
- Employing Organizations
- Phone Stations (as opposed to just phone numbers as such)

◆ So what the entity types are in a given working environment are partly a matter of judgment, as explained earlier.

But we'll see that in designing a DB we may need to introduce new, not immediately obvious, entity types.

◆ “*Entities*” are, or should be, the things of a type: *e.g.*, *individual* people. An entity is represented by a *row* in the appropriate table.



# Entity Terminology

## ◆ *Unfortunately:*

“**entity**” is often used to mean entity type.

“**entity set**” is often used for entity type.

“**entity occurrence**” is often used to mean individual entity.



# Relationships

- ◆ These are the relationships between entity types, such as
  - A **person being employed by an organization**
  - A **person having a phone station**
- ◆ Have to think about both directions of a relationship: e.g., both *employed-by* and *employs*.
- ◆ CAUTION: Tables are also called “relations” [hence “relational” DB] (much more on this later). This is to do with the **internals** of tables/entities rather than with “relationships” between entities.



# Relationship Connectivity

- ◆ Relationships are importantly categorized as to **uniqueness or multiplicity** of entities at either end – “*connectivity*.”

*Has big effect on DB design.*

- *1:1* (“one to one”): e.g., the people/phone-stations relationship, if each person has *at most one* phone station and each phone station is assigned to *at most one* person.
- *M:N* (“many to many”): e.g., the employs relationship, assuming a person *may* have *more than one* employing organization (**or none**) and an organization *may* have *more than one* employee (**or none**). (Don’t take “many” seriously – just means possibly more than one.)
- *1:M* (“one to many”): e.g., the employs relationship, if an organization *may* have *more than one* employee (**or none**) but a person has *at most one* employing-org.



# Relationship Cardinality

- ◆ Relationships can be further specified as to “how many entities allowed or required at either end” – *cardinality*.

*Also has significant effect on DB design.*

- ◆ In a relationship from entity type A to entity type B, a minimum and a maximum can be specified for the number of B entities for each A entity.
- ◆ A maximum greater than 1 can only be specified if the relationship from A to B is 1:M or M:N. (So the notions of connectivity and cardinality are not properly separated).
  - E.g., could be specified that a person can only be **employed by** up to **five** organizations.
- ◆ Most normally, the important choice for the minimum is between none and one. E.g., the minimum for **employed-by** could be **none**, but the minimum for **employs** could be **one**. But the minimum number of wheels for a car could be specified to be three.
- ◆ If the minimum is none, then B is *optional for A*. Otherwise, it is *mandatory for A*.



# Attributes

- ◆ Attributes of entities of a given type are the names of the different pieces of information that need to be stored for entities of that type. So they're just the **column names** for the table for the entity type.
  - E.g., entities of the type “people” could have the following attributes: **person ID number, last name, first name, phone number, age.**
- ◆ Note: Attributes include artificial ones like the employer identity numbers (EMPL. NUM.) that we introduced in an example above. These may have no significance outside the DB itself.
- ◆ Relationships are represented by associative linking by means of shared attributes. (For now, will always assume that the same attribute name is used in each of the tables involved.)



# Attribute Determination

- ◆ REMEMBER: *Rows in a table are uniquely determined (picked out) by the values in some set of columns, i.e. **the values of some collection of attributes**.*

That is, given some values for those attributes, there is at most one entity that has those values for those attributes, at any given time.

- ◆ Hence, that collection of attributes *determines* all the other attributes.
- ◆ That is, given some values for the determining attributes, there's at most one value for each of the other attributes, at any given time.



# Attribute Determination, contd.

- ◆ More generally, a collection of one or more attributes *determines* another attribute A if only one value for A is possible given the values for the former attributes.

*E.g.*, the collection DAY-NUMBER, MONTH and YEAR specifying birth-date in a table about people could determine DAY-NAME, even though it doesn't determine other attributes such as NATIONALITY: several people could have the same birth-date but be of different nationalities.

- ◆ We alternatively say that DAY-NAME is *functionally dependent on* DAY-NUMBER, MONTH and YEAR.