

# Fundamentals/ICY: Databases 2013/14

## ***Week 5: Friday***

*(slides added at front on the day)*

*John Barnden*

*Professor of Artificial Intelligence*

*School of Computer Science*

*University of Birmingham, UK*



# Notes/Suggestions about the Week 5 SQL and Lab Work

## ◆ What much of it is about is “joining”

- [later on in lectures; textbook section 9.2]

## ◆ Much is about “subqueries”

- [textbook section 9.3]



# Extra Things to Try with Joining

- ◆ `SELECT * FROM course, staff;`
- ◆ `SELECT * FROM staff, staff;`
- ◆ `SELECT * FROM staff AS s1, staff AS s2;`
- ◆ `SELECT s1.sid, s2.sid FROM staff AS s1, staff AS s2;`
- ◆ `SELECT COUNT(*) FROM staff;`  
`SELECT COUNT(*) FROM staff AS s1, staff AS s2;`
  - What's the relationship between those two numbers?
- ◆ `SELECT COUNT(*) FROM staff, lecturing;`



# Extra Things to Try with Joining, contd

- ◆ `SELECT s1.sid, s2.sid FROM staff AS s1, staff AS s2  
WHERE s1.sid = s2.sid;`
- ◆ `SELECT s1.sid, s2.sid FROM staff AS s1, staff AS s2  
WHERE s1.sid > s2.sid;`
- ◆ `SELECT staff.sid, lecturing.* FROM staff, lecturing  
WHERE staff.sid > lecturing.sid;`
- ◆ `SELECT staff.sid, lecturing FROM staff, lecturing  
WHERE staff.sid > lecturing.sid;`



**Reminder of Monday**



# 1:1 Connectivity between Tables

## People

<u>PERS-ID</u>	NAME	<i>PHONE</i>	<i>EMPL ID</i>	AGE
9568876	Chopples	0121-414-3816	E22561	37
2544799	Blurp	01600-719975	E85704	21
1698674	Rumpel	07970-852657	E22561	88
5099235	Biggles		E22561	29

*1:1: that is, no more than one phone allowed per person, and vice versa.*

## Phones

<u>PHONE</u>	TYPE	STATUS
0121-414-3816	office	OK
01600-719975	home	FAULT
0121-440-5677	home	OK
07970-852657	mobile	UNPAID

Note: the representation is still *asymmetric* in that the People table mentions phones but not vice versa – *symmetry would create extra redundancy*.

*NB: Biggles has no phone listed, and 0121-440-5677 has no person recorded. Suggests a possible reason for not combining such tables.*



# 1:M Connectivity between Tables

## People

<u>PERS-ID</u>	NAME	PHONE	EMPL ID	AGE
9568876	Chopples	0121-414-3816	E22561	37
2544799	Blurp	01600-719975	E85704	21
1698674	Rumpel	07970-852657	E22561	88
1800748	Dunston	0121-414-3886	E22561	29

*More than one employee allowed per organization, but no more than one employer per person.*

*NOTE direction of use of the foreign key. Why so??*

## Organizations

<u>EMPL ID</u>	EMPL NAME	ADDRESS	NUM EMPLS	SECTOR
E48693	BT	BT House, London, ...	1,234,5678	Private TCOM
E85704	Monmouth School	Hereford Rd, Monmouth, ...	245	Private 2E
E22561	University of Birmingham	Edgbaston Park Rd, ....	3023	Public HE



**New**



# M:N Connectivity between Tables

- ◆ *IF* we represent M:N connectivity in a similar way to 1:M, then we can expect that
  - in the People table: some people will each have several employers listed
  - *or* in the Organizations table: some organizations will each have several employees listed
  - *or* both.
- ◆ This is a problem. Why?



# M:N Connectivity between Tables, contd.

- ◆ Because of this problem, an M:N relationship is usually broken up into two 1:M relationships.
- ◆ This means introducing an extra “**bridging**” or “linking” or “composite” entity type (hence table) to stand between the two original ones.



**M:N -- a person may be employed by more than one organization  
and an organization may employ more than one person**

PEOPLE

*does not have*  
an EMPL ID attrib.

EMPLOYMENTS

each = *person-id*  
+  
*organzn-id*

**possibly plus other  
attributes**

ORGANIZATIONS

*does not have*  
a PERS ID attrib





# M:N Connectivity between Tables using a Bridging Entity Type

## People

<u>PERS-ID</u>	NAME	AGE
9568876	Chopples	37
2544799	Blurp	21
1698674	Rumpel	88
1800748	Dunston	29

## Employments

<u>PERS-ID</u>	<u>EMPL ID</u>	START
9568876	E22561	15-06-99
9568876	E85704	23-11-03
1698674	E22561	23-11-03
1800748	E22561	07-07-97

## Organizations

<u>EMPL ID</u>	EMPL NAME	ADDRESS	NUM EMPLS	SECTOR
E48693	BT	BT House, London, ...	1,234,5678	Private TCOM
E85704	Monmouth School	Hereford Rd, Monmouth, ...	245	Private 2E
E22561	University of Birmingham	Edgbaston Park Rd, ....	4023	Public HE



# Relationship Participation

## ◆ *Optional* [in a particular direction, X to Y]:

- an X entity *does not require* a corresponding Y entity occurrence
- *i.e.* the **minimum** number of Ys per X is 0

## ◆ *Mandatory* [in a particular direction, X to Y]:

- an X entity *requires* a corresponding Y entity occurrence
- *i.e.* the **minimum** number of Ys per X is 1 or more



# Strong (or Identifying) Relationships

- ◆ A relationship **from** entity type A **to** entity type B, mediated by having A's primary key (PK) as a foreign key in B, is **strong** when *B's PK contains A's PK*.
  - So, B entities are defined *in terms of* A entities.
- ◆ Includes the case of B's PK just being the same as A's PK.
- ◆ E.g., A = Customers, B = Dependants, where
  - A's PK is: **CUST\_ID**
  - B's PK is: **CUST\_ID, FIRST\_NAME, CONNECTION**.
  - So a PK value in B could be **(1698674, Mary, child)** , meaning that this entity is the **child** called **Mary** of person **1698674** in the Customer table.



# a Strong Relationship

Customers (the “A” type)

<u>CUST-ID</u>	NAME	PHONE	EMPL ID	AGE
9568876	Chopples	0121-414-3816	E22561	37
2544799	Blurp	01600-719975	E85704	21
1698674	Rumpel	07970-852657	E22561	88
1800748	Dunston	0121-414-3886	E22561	29

*Strong relationship  
going from A to B*

*(we could say: “B is  
strongly dependent on  
A”)*

Dependants (the “B” type)

<u>CUST-ID</u>	<u>FIRST NAME</u>	<u>CONNECTION</u>	LIVES_WITH
2544799	John	civil partner	TRUE
1698674	Mary	child	FALSE
1698674	Mary	spouse	FALSE
1698674	David	child	TRUE



# Weak (or Non-Identifying) Relationships

- ◆ A relationship is *weak* when it isn't strong!

So, most relationships are weak.

- ◆ Note that strength/weakness is *directional*: the People to Dependants relationship (above) is *strong*, but the Dependants to People relationship is *weak*.
- ◆ *Exercise*: Can a relationship be weak in both directions?
- ◆ *Exercise*: Can a relationship be strong in both directions?



# Weak *Entity Types*

◆ A *weak* entity type **E** is one such that there is a relationship **R** from some other entity type **F** to **E** that satisfies the following two conditions:

- **R** is strong.
- **E** is existence-dependent on **F** via **R**.

That is, an **E** entity *e* can only exist in the database if some **F** entity in the database bears relationship **R** to *e*.

This actually just adds the requirement that  $R^{-1}$  (the **R** relationship but in the **E** to **F** direction) be *mandatory*.



# Weak Entity Types, contd.

- ◆ So on a previous slide, Dependants is *weak*, because there is a **strong** relationship to it from Customers, and Dependants is **existence-dependent** on Customers via this relationship.
  - **Mary's** existence in the database as a member of Dependants relies on the existence of **customer 1698674** in the database.

But this **doesn't mean** Mary would vanish from the world if customer 1698674 left the database or even if that customer were to vanish from the world.

And indeed Mary could herself be an entity in type Customers, and could remain so even if 1698674 left the DB or really vanished from the world.



# Strong *Entity Types*

- ◆ A ***strong*** entity type is one that is not weak! ....
- ◆ So, in particular, any entity type that receives only weak relationships from other entity types is strong.
  - *BUT this is NOT the ONLY way for an entity type to be strong. [Exercise]*
- ◆ So the usual case is for an entity type to be strong.
- ◆ And any entity type that is not existence-dependent on anything is strong.



# Mental Exercises for You

- ◆ What about the Employments bridging type we introduced?
- ◆ Although it might look at first as though a strong relationship necessarily implies existence dependence, it doesn't. Why not?
- ◆ One entity type can be existence-dependent on another without therefore being weak. Why?