|  |
| --- |
| **RELATIONSHIPS & MODELLING**  **ENTITY RELATIONSHIP DIAGRAMS**  **E/R DIAGRAMS** |

**RELATIONSHIPS & MODELLING**

**ENTITY RELATIONSHIP DIAGRAMS**

## E/R DIAGRAMS

# After figuring out the entities (later became tables) we need to develop a

PRELIMINARY MODEL

We need to show RELATIONSHIPS between entities or logical associations

So far we have

1 found all the attributes

2 found at least one entity

3 assigned each of the attributes to one of the entities

At the same time the attributes were reviewed. Some attributes were not being stored in entities, such as calculated attributes (GPA). Also care was taken not to forget an attribute and not to assign the attribute to more than one entity.

We took a break from the process of designing to demonstrate the notation used for showing entities and attributes.

At this time before the process of normalizing the entities is shown, one other notation is required. An Entity Relationship Diagram or ERD or ER diagram can be used to graphically represent the system. This graphic representation shows all the entities in the system and how they relate to each other. Once again there are differences in notation, although all are very similar and self-explanatory. The notation used in these notes is very simple. It does not try to show all the details that some methods do. It is kept overly simple so that it can be easily done in WORD.

**ASSUME 2 ENTITIES**

STUDENT

COLLEGE

PROGRAM

COLLEGE PROGRAMS STUDENT

#### Contains data Contains data

about programs about students

- computer - name

- nursing - address

- sports - age

ARE THERE ANY LOGICAL CONNECTIONS between these two entities?

If there is then

**HOW IS DATA RELATED BETWEEN ENTITIES**

Relational databases have **no physical links (pointers) between existing entities**. That means data must be designed so that the DBMS can build relationships or links upon request. This is done through a **common field**.

Example 1: We have two tables with sample data, STUDENTS with 2 examples and PROGRAMS with 3 sample pieces of data. What program does the student 1111 take?

STUDENTS

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| ID | LAST NAME | FIRST | ADDRESS | GENDER | OFFICE | PROGRAM NO. |
| 1111 | TARR | RON | ADDRESS | M | 2099 | **17** |
| 2222 | RYAN | MARY | ADDRESS | F | 2122 | **13** |

PROGRAMS

|  |  |  |
| --- | --- | --- |
| PROGRAM NO. | DEPARTMENT NAME | SCHOOL NO. |
| **10** | ACCOUNTING & FINANCE | 100 |
| **13** | COMPUTER STUDIES | 200 |
| **17** | ADMINISTRATION | 100 |

Example 2: Some of the employees of a company are salespeople. These salespeople receive a commission if their customer makes a purchase. How are the entities CUSTOMER and EMPLOYEE related in order to know which salesperson will get the commission?

Given the two entities as follows:

CUSTOMER [CID, CNAME, BILLING-ADDRESS, SHIPPING ADDRESS … **SALESREP-ID** ]

EMPLOYEE [**EID**, ENAME, EADDRESS, GENDER, SALESREP-Y-N, …]

The attribute SALESREP-ID in CUSTOMER will match an EID in EMPLOYEE

NOTE: Often the common fields have the same name, but they do not have to. It is essential that it be the same datatype.

E/R DIAGRAMS - LOGICAL DESIGN MODELS

The notation used to **model the logical design** is an E/R diagrams. Again these diagrams are meant as simple pictorials. (If you specialize in the database area, an excellent choice, you will also see STAR and SNOWFLAKE design diagrams)

# **NOTATION**

**Every entity is enclosed in boxes.**

If there are two entities DOCTORS and PATIENTS, then they are enclosed in a box as follows

PATIENTS

DOCTORS

**If a relationship exists between entities, join them with a line.**

PATIENTS

DOCTORS

The line will be modified to show the kind of relationship.

BIG CAUTION

DON'T JOIN EVERY BOX

There will be more extra fields than data. One box connected to 50 boxes requires 50 common fields or 50 foreign keys. (More about foreign keys later)

3 TYPES OF RELATIONSHIPS

HOW TO FIND OUT WHICH ONE

Patients

Doctors

1 ASK … IS THERE A RELATIONSHIP

Yes … Join with line

2 ASK … WHAT TYPE …

First Question: Start with any entity – i.e. DOCTOR

IF I LOOK AT 1 OCCURRENCE OF DOCTOR

CAN THEY HAVE NO PATIENTS

ONE PATIENT

MANY PATIENTS

Yes … they can have all of them or many

###### Draw joins like this

The arrow at this stage shows a ratio of one doctor to many patients. We always show the largest number. It is possible that a specific doctor in the table may have no patients or one patient.

Next Question: Go to the other entity - PATIENT

IF I LOOK AT 1 OCCURRENCE OF PATIENT

CAN THEY HAVE MORE THAN ONE DOCTOR

Yes … they can have all of them or many

Draw the join like this

The line now will have an arrow at both ends meaning MANY

This is a MANY to MANY

ANOTHER 2 RELATIONSHIPS

Note: In some cases a double arrow at each end means many, so does crows feet and a few other symbols. I will accept any style as long as it is consistent.

MEN

WOMEN

- Assume the relationship is MARITAL

- Assume Canadian Laws

SAME 2 QUESTIONS

Ask from both sides

There will be only 1 occurrence from the entity men

(perhaps none)

If I look at one woman

or

for each occurrence in

the women entity

The relationship is ONE to ONE

ASIDE: This type of relationship in organizations is not common2 MORE ENTITIES

INVOICE

CUSTOMER

1 IS THERE A RELATIONSHIP

🡺 If YES then draw a line to join them

2 ASK QUESTIONS

For 1 occurrence of customer

there can be none 🡺

one 🡺 invoices ?

many 🡺

ASK OTHER SIDE

For 1 occurrence of invoice

there can be none 🡺

one 🡺 customers ?

many 🡺

3 APPLY THE ARROWS

This relationship is a ONE to MANY

- THE MOST COMMON IN DATABASE DESIGN

**WORKSHOP**

What is the relationship between these pairs of entities.

### Between

INVOICE and INVOICE LINE

JOCKEYS HORSES

COLLEGE ONTARIO

FUNDING GOVERNMENT

DOCTORS COLLEGE OF ART

**LAYOUT SUGGESTIONS**

There are layouts of the ERD that are easier to read than others. The left example is easier to read the relationship than the right example. Where possible with one to many relationships the one entity should be placed above the many entity. The instructor will provide other suggestions.

**LAYOUT SUGGESTIONS**

(cont’d)

##### B

##### MANY TO MANY

### These are NOT supported by most DBMS. This demonstrates the difference between conceptual design and physical design. It means that more entities are required before implementation can occur.

**C**

**IS THERE ONE ENTITY CENTRAL TO ALL OF THE SYSTEM**

Example students, then put this box near the middle