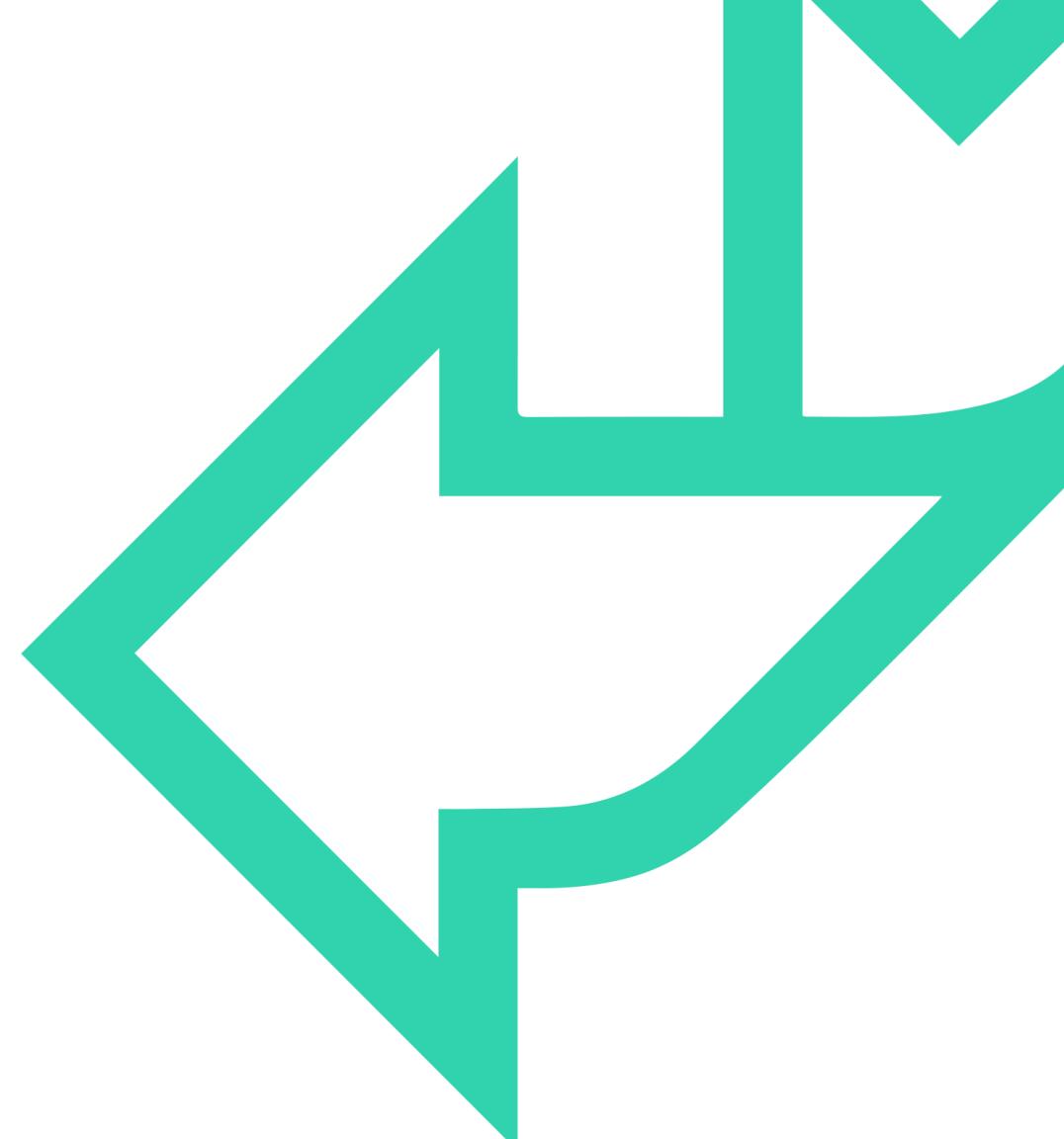




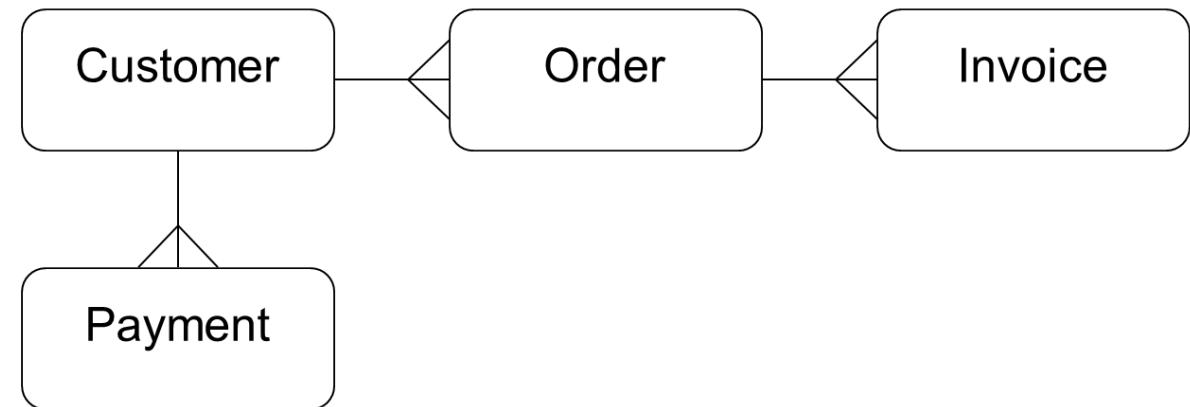
# Databases and SQL

Tutor: Rob Menzies  
Email: [rob.menzies@qa.com](mailto:rob.menzies@qa.com)

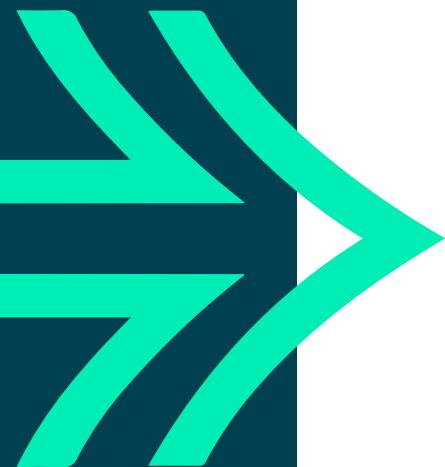


# Entity Relationships

# ENTITY RELATIONSHIP DIAGRAMS (ERD)



# AN ENTITY



Invoice

# RELATIONSHIPS



## *Relationships*

Employees work in a department

A truck is a type of vehicle

Lawyers advise clients

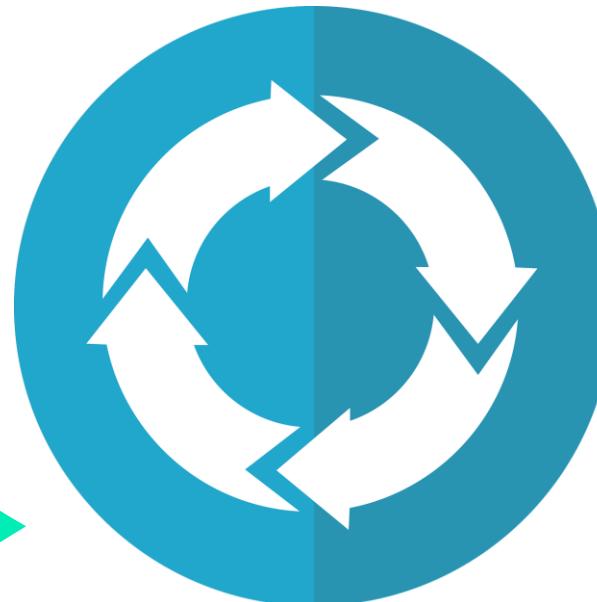
# BENEFITS OF E-R MODELLING



## Using the client requirements

- Start with Entities
- Then add the relationships
- Specify the attributes

**Iterate**



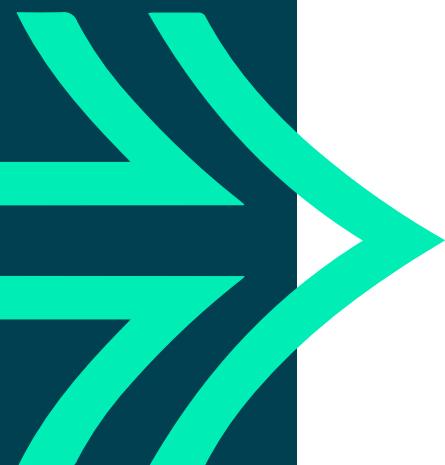
**Simplified Model**

**Storage Independence**

**Independent of  
Access Method**

**Readable  
by Analyst and End User**

# OPTIONALITY AND NAMING RELATIONSHIPS



*Mandatory:*



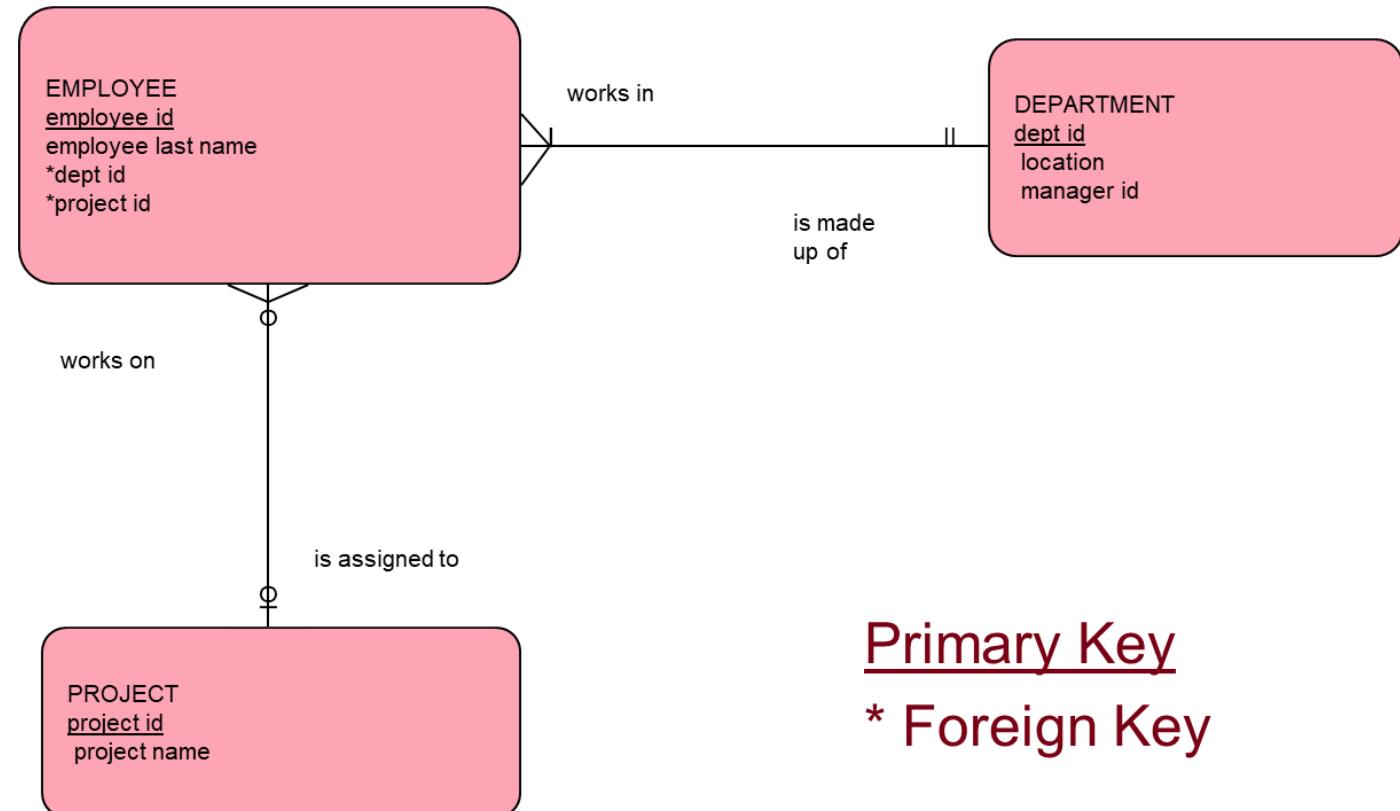
*Conditional:*



*Optional:*

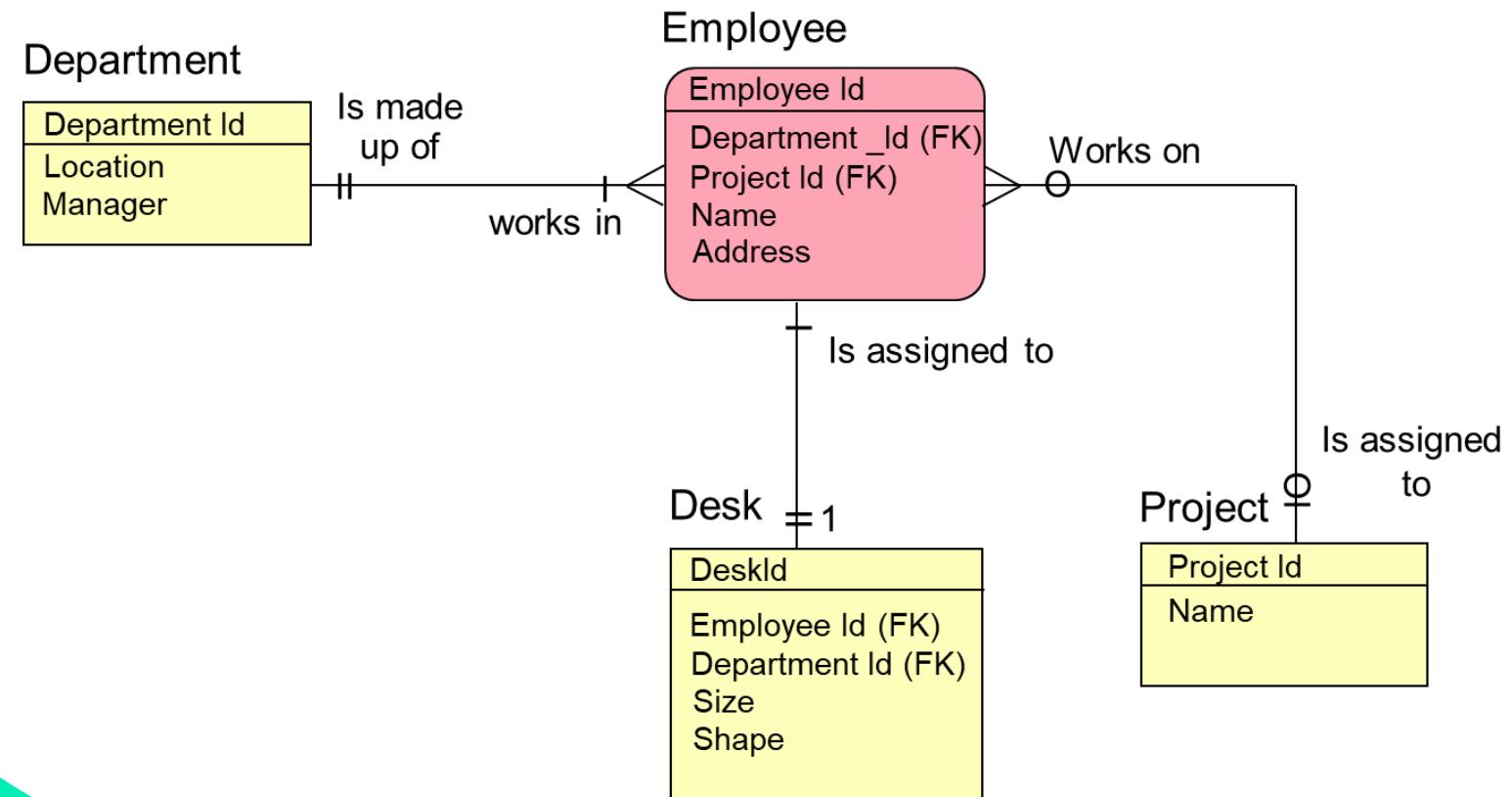


# IE NOTATION: KEYS



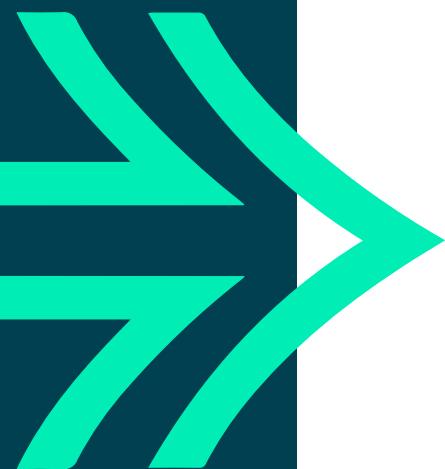
Primary Key  
\* Foreign Key

# AN EXAMPLE



# APPROACH

- 1. Identify candidate entities**
- 2. Identify relationships between candidate entities**
- 3. Draw data model**
- 4. Review and refine**



# ONE TO ONE AND ONE TO MANY

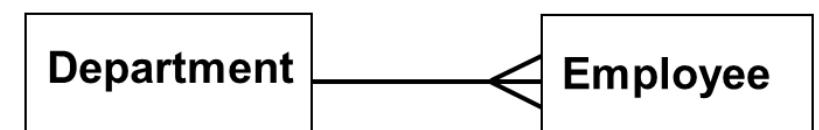


## SE crow's-foot (Bachman) Diagrams

“One to One”

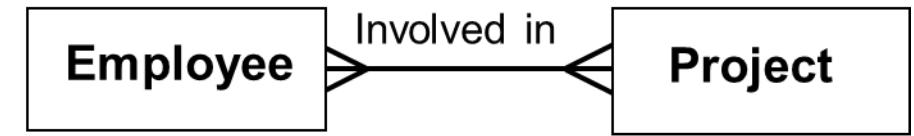


“One to Many”

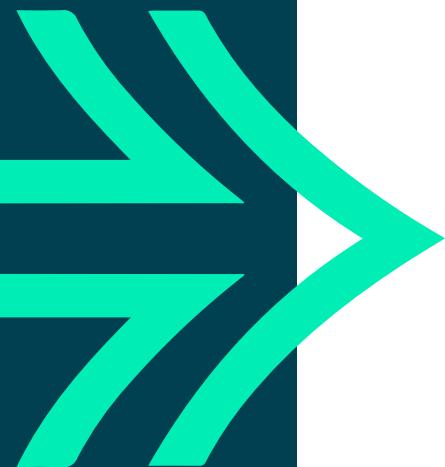


# MANY TO MANY

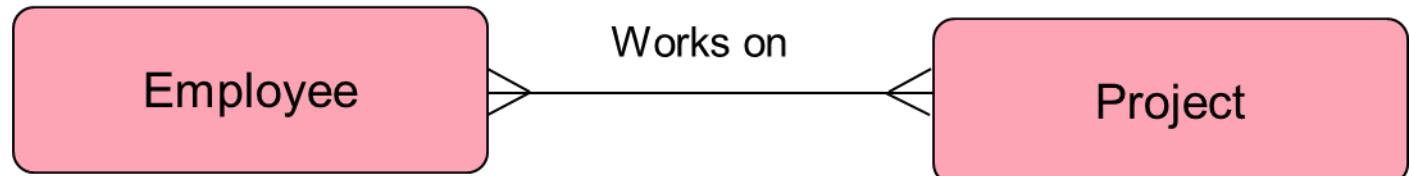
“Many to Many”



# THE MANY-TO-MANY PROBLEM



How can we store this data?



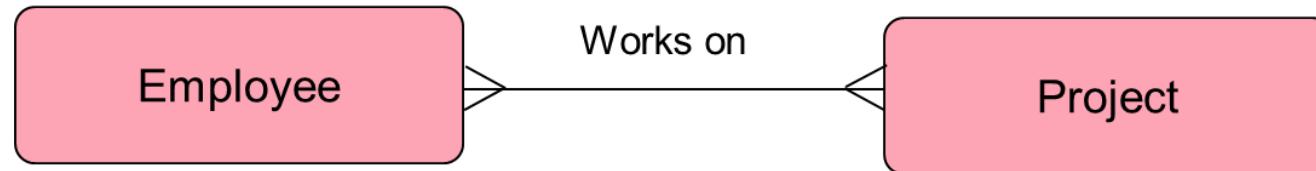
Employee	Project
E1	A,B,C
E2	D,E
E3	A,E
E4	B,E
E5	A,C,E
E6	D



# THE MANY-TO-MANY PROBLEM (CONT.)



How can we store this data?



Perhaps we should store it like this?

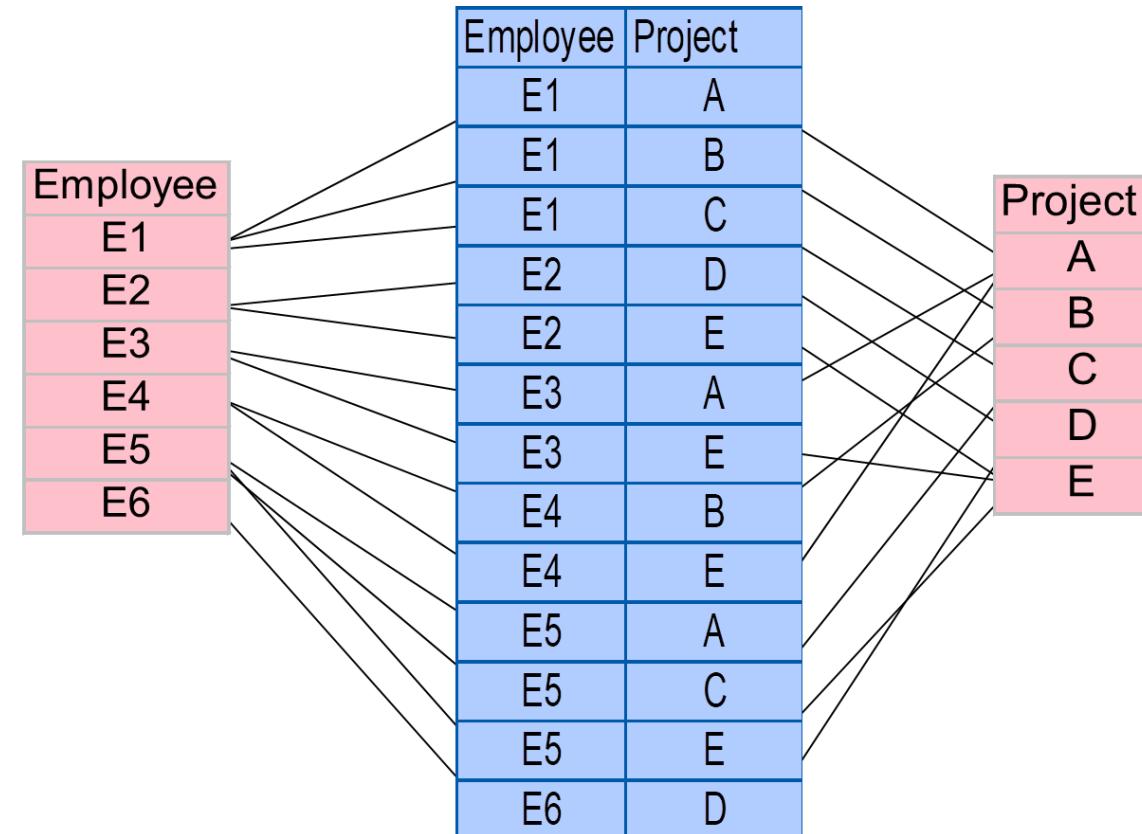
Project	Employee
A	E1,E3,E5
B	E1,E4
C	E1,E5
D	E2,E6
E	E2,E3,E4,E5



It depends what questions we wish to ask

# THE MANY-TO-MANY SOLUTION

How about this?



# THE MANY-TO-MANY PROBLEM EXAMPLE 2

How can we store this data?

Student ID	Class ID	Student Name
1	3, 5, 9	John
2	1, 4, 5, 9	Debbie

Perhaps we should store it like this?

Student ID	Class ID 1	Class ID 2	Class ID 3	Student Name
1	3	5	9	John
2	1	4	5	Debbie

Both of these will cause either problems of data redundancy or will simply be confusing.  
Even if we try a one to many there will be repeated (redundant data) in one of the tables

# THE MANY-TO-MANY SOLUTION



How about this?

Student ID	Student name
1	John
2	Debbie

Student ID	Class ID
1	3
1	5
1	9
2	1
2	4
2	5
2	9

Class ID	Class name
1	English
2	Maths
3	Spanish
4	Biology
5	Science
6	Programming
7	Law
8	Commerce
9	Physical Education

We use a joining or bridging table to avoid many to many relationships now we have no data redundancy

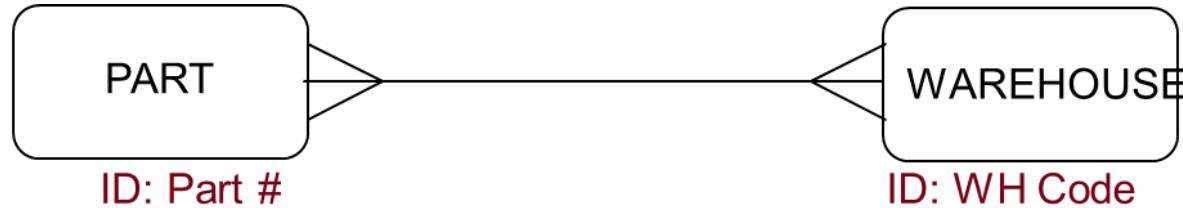
# MANY TO MANY RELATIONSHIPS EXAMPLE 2



To which entity do you assign the attribute “quantity-on-hand”?

If you imagine we have 10 warehouses and the total quantity of parts we have in stock is divided amongst these warehouses, how does any individual warehouse know how many parts are available at any time?

# MANY TO MANY RELATIONSHI PS EXAMPLE 2



To which entity do you assign the attribute “quantity-on-hand”?



ATTRIBUTES: Quantity on hand  
Re-order Level etc.