



DATABASE DESIGN

Julie Allen 2024

ENTITY RELATIONSHIP MODELLING

Today's Agenda:

- The design process
- Entity Relationship Modelling Concepts
 - Entities
 - Attributes
 - Relationships
 - Cardinality



1ST STEP: WHAT ARE THE DATABASE REQUIREMENTS?

Analyse the Problem Domain...

What data do we need to store?

What events happen that create or have a need for data?

Who needs what – people, departments, organisations?

What queries and reports will be needed?

THE DESIGN PROCESS

Conceptual Design

A data model is produced which reflects the nature of the application data.
The model does not specify any implementation details.
The outcome of this phase is a high level Entity Relationship Model, which is not concerned with attributes and data types.

Logical Design

In this stage the Entity Relationship Diagrams are refined to give a more detailed structure, adding detail in the form of attributes, relationships, and data types.
We carry out the process of Normalisation.
This phase is independent of any DBMS.

Physical Design

Produce the database schema: specify table structures, data types, constraints, etc.
Decide on a DBMS.
Decisions made regarding data storage, access control, etc.

ENTITY RELATIONSHIP MODELLING CONCEPTS

Entities

An object, person, thing, concept.

E.g. student, course, module, department, employee, customer, order, vehicle, order, place.

Relationships

A relationship is an association between two or more entities.

E.g. students take courses, a department has employees, a customer places orders.

Attributes

Entities are defined by a set of characteristics or properties which we call attributes.

E.g. a student has a student number, a name, an address, a course, etc.

ENTITY RELATIONSHIP DIAGRAM - ERD

ERDs are the most widely used technique for designing data models.

There are many different notations used, e.g. Chen, Bachman, Crow's Foot, etc.

This module uses the 'Crow's Foot' notation as it is the simplest and easiest way to learn Entity Relationship Modelling.



ENTITY RELATIONSHIP DIAGRAM

Entities are shown as rectangles, and are thought of as a singular noun.

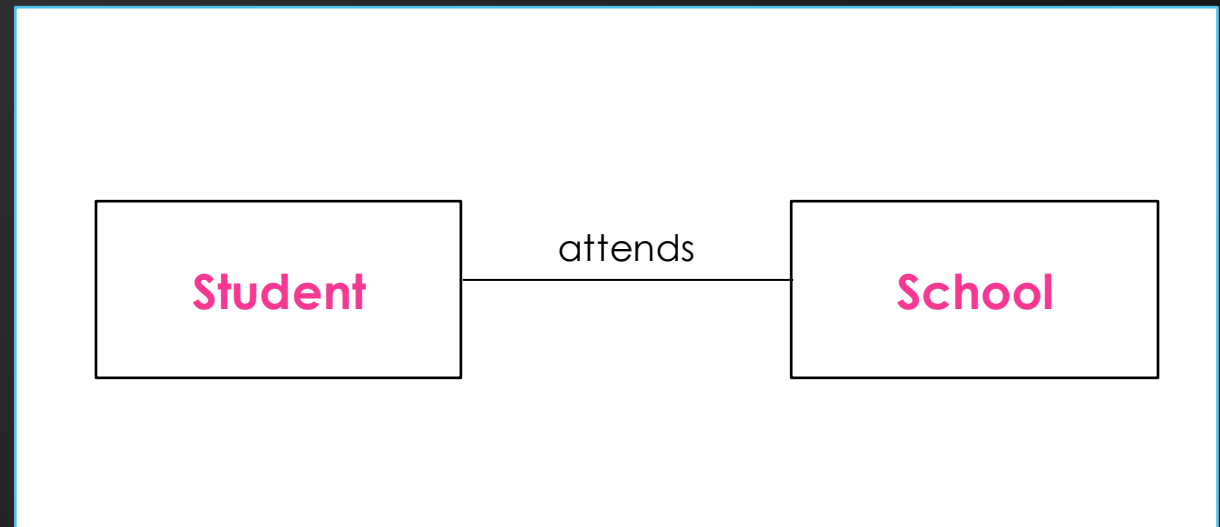
Relationships are shown as a connecting line between the entities and are thought of as verbs.

We can read this relationship as...

Each student attends a school.

And the other way round...

Each school has attending students (sounds more awkward).

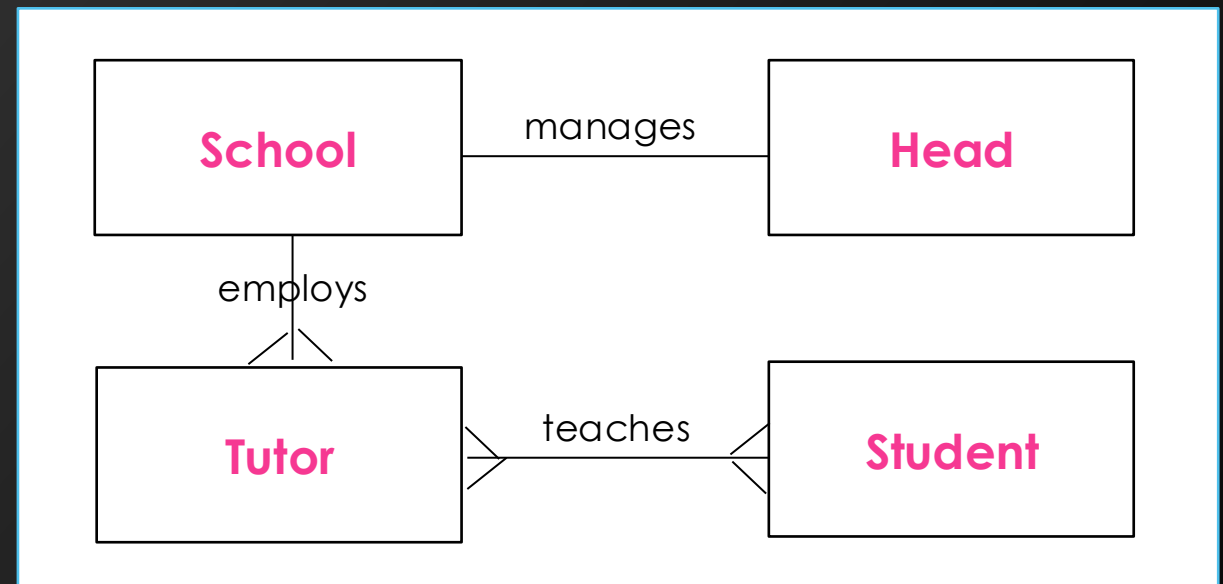


It is important to understand that each entity box is a type of 'thing' and refers to a set of that thing, e.g. the student entity refers to a set of students, not just one.

ENTITY RELATIONSHIP DIAGRAM

A more elaborate ERD to model – school, head, tutor, and student...

Note that the end of some of the lines have been shown as 'Crow's Foot' notation.



Crow's Foot notation is used to specify the 'cardinality' of the relationship

The Crow's Foot end means 'MANY'

CARDINALITY

Cardinality applies a numerical context to the relationship between two entities...

We can interpret and verbalise a relationship between two entities in the following way:

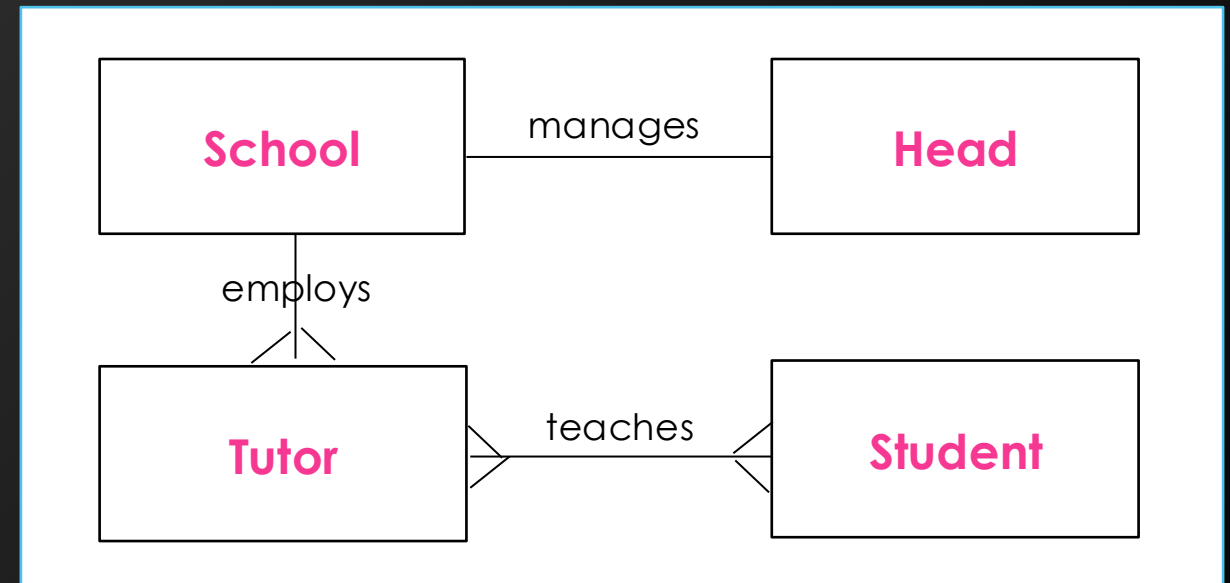
School – Tutor:

Starting at the School end we can say –

one school employs **many** tutors.

Starting at the Tutor end we can say –

one tutor is employed by **one** school.



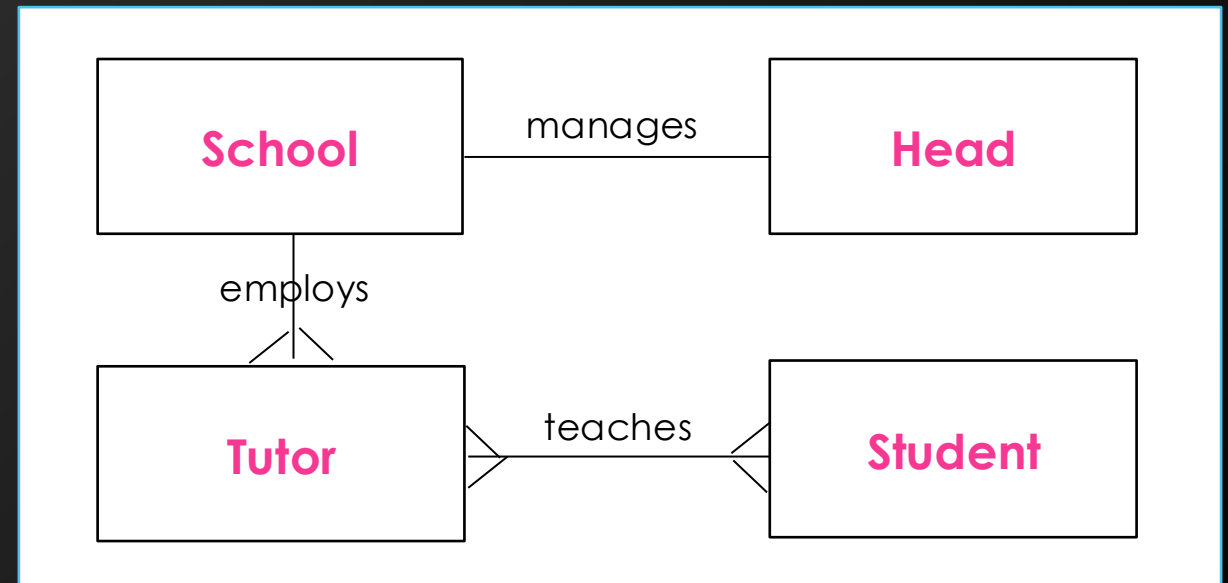
The Crow's Foot end means 'MANY'

CARDINALITY

There are three relationship cardinalities that can exist between two entities:

1:1 One-to-one
1:n One-to-many
M:n Many-to-many

1:1 - One Head manages one School
1:n - One School employs many Tutors
m:n - Many Tutors teach many Students

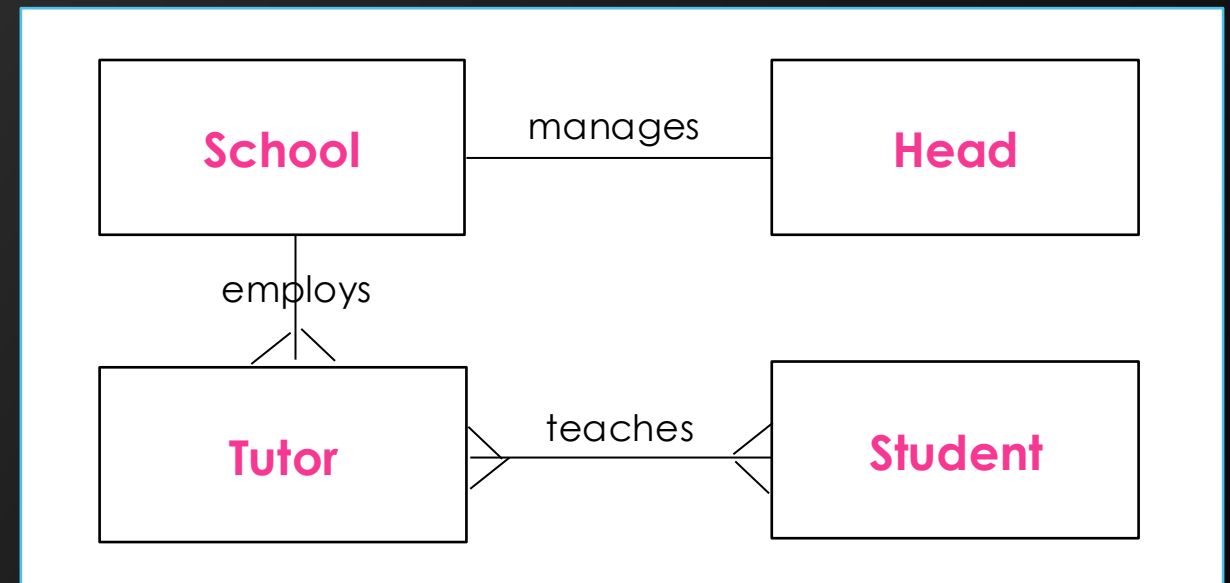


CARDINALITY

But what if ...

A Tutor can also be employed by more than one school?

Or a School can also be managed by more than one Head, e.g. Head of Computing, Head of Engineering?



How do we show these differences on our diagram?

CARDINALITY

We can further refine the cardinality with a minimum and a maximum...

There are three symbols used to represent the Cardinality:

The RING represents 'zero'

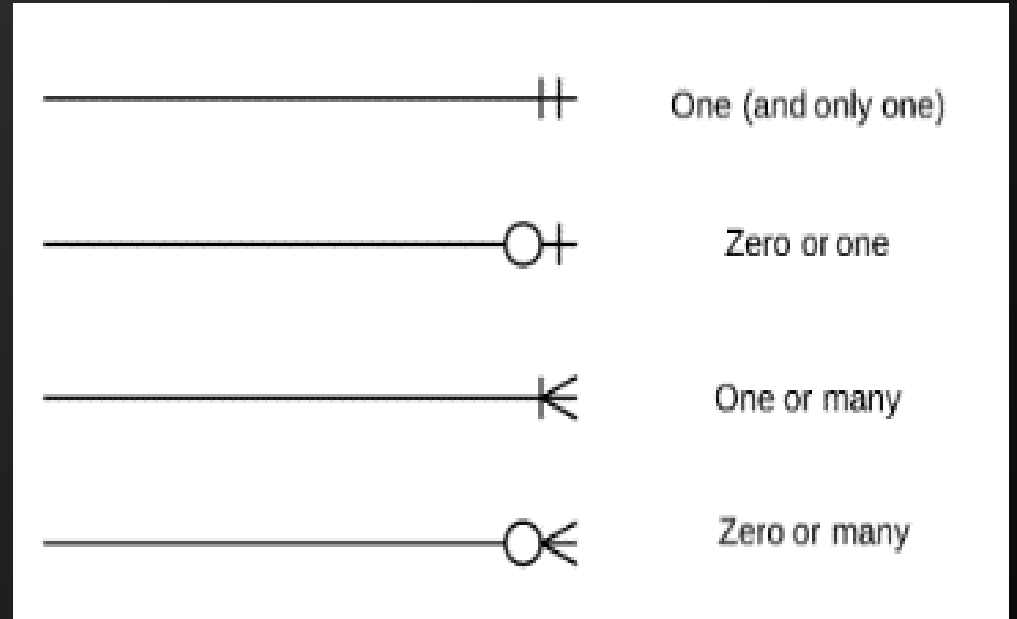
The LINE represents 'one'

The CROW'S FOOT represents 'many'

Examples:

A School must have at least one Head, but it could have more than one. This would be a 'One or Many'.

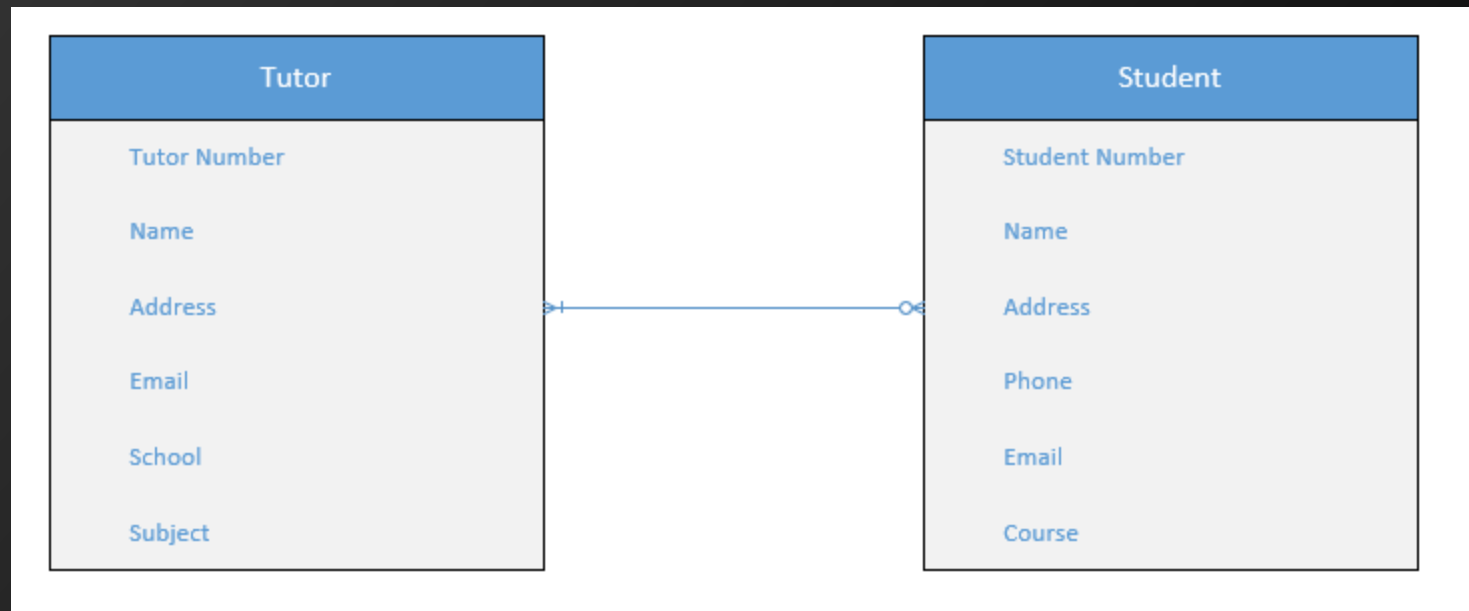
A Head can manage only one School. This would be a 'One and Only One'.



ATTRIBUTES

The first level ERD's only model the structure of the system. The next step is to identify the attributes or properties of each entity...

Tutor and Student entities with added attributes.



This has been kept simple for ease of understanding, in reality there may be many more attributes.

KEYS

A **PRIMARY** Key is a column that uniquely identifies every row/record in a table.

A **FOREIGN** Key is a column that relates to the primary key in another table.

A **COMPOSITE** Key is a primary key made up of more than one column



KEYS

In our EMP and DEPT table the **PRIMARY KEYS** were EMPNO (emp) and DEPTNO (dept).

In order to create the relationship between the two tables a copy of DEPTs Primary Key column (deptno) was placed in the EMP table. This new column is known as a **FOREIGN KEY**.

EMP

EMPNO	ENAME	JOB	MGR	HIREDATE	SAL	COMM	DEPTNO
7839	KING	PRESIDENT	-	11/17/2001	82000	-	10
7782	CLARK	MANAGER	7839	06/09/2009	27500	-	10
7934	MILLER	CLERK	7782	01/23/2006	13200	-	10
7566	JONES	MANAGER	7839	04/02/2012	26850	-	20
7788	SCOTT	ANALYST	7566	12/09/2008	19500	-	20
7902	FORD	ANALYST	7566	12/03/2014	21500	-	20
7369	SMITH	CLERK	7902	12/17/2002	13750	-	20
7876	ADAMS	CLERK	7788	01/12/2011	11900	-	20
7698	BLAKE	MANAGER	7839	05/01/2005	24000	-	30
7499	ALLEN	SALESMAN	7698	02/20/2006	19000	6300	30
7521	WARD	SALESMAN	7698	02/22/2018	18500	4500	30
7654	MARTIN	SALESMAN	7698	09/28/2008	15650	3400	30
7844	TURNER	SALESMAN	7698	09/08/2011	18500	0	30
7900	JAMES	CLERK	7698	12/03/1998	11950	-	30
4422	STEVENS	MANAGER	7839	02/14/2019	24750	-	50
3258	GREEN	SALESMAN	4422	07/24/2003	18200	2750	50
6548	BARNES	CLERK	4422	01/16/2018	11950	-	50

DEPT

DEPTNO	DNAME	LOC
10	ACCOUNTING	LONDON
20	RESEARCH	PRESTON
30	SALES	LIVERPOOL
40	OPERATIONS	MANCHESTER
50	MARKETING	LUTON

MANY TO MANY RELATIONSHIPS

Each Student can take MANY Modules.

Each Module has MANY Students.



Why do we think this might be a problem?

MANY TO MANY RELATIONSHIPS

Student

Student ID	Name	Module ID
101	Zayne	CO1409, CO1507, CO1301
102	Sarah	CO1409, CO1301, CO1605
103	Michael	CO1507, CO1301, CO1605

Module

Module ID	Name	Student ID
CO1409	Programming	101, 102
CO1605	Databases	102, 103
CO1507	Networking	101, 103
CO1301	Games	101, 102, 103

In relational databases multi-valued columns are not allowed.

MANY TO MANY RELATIONSHIPS

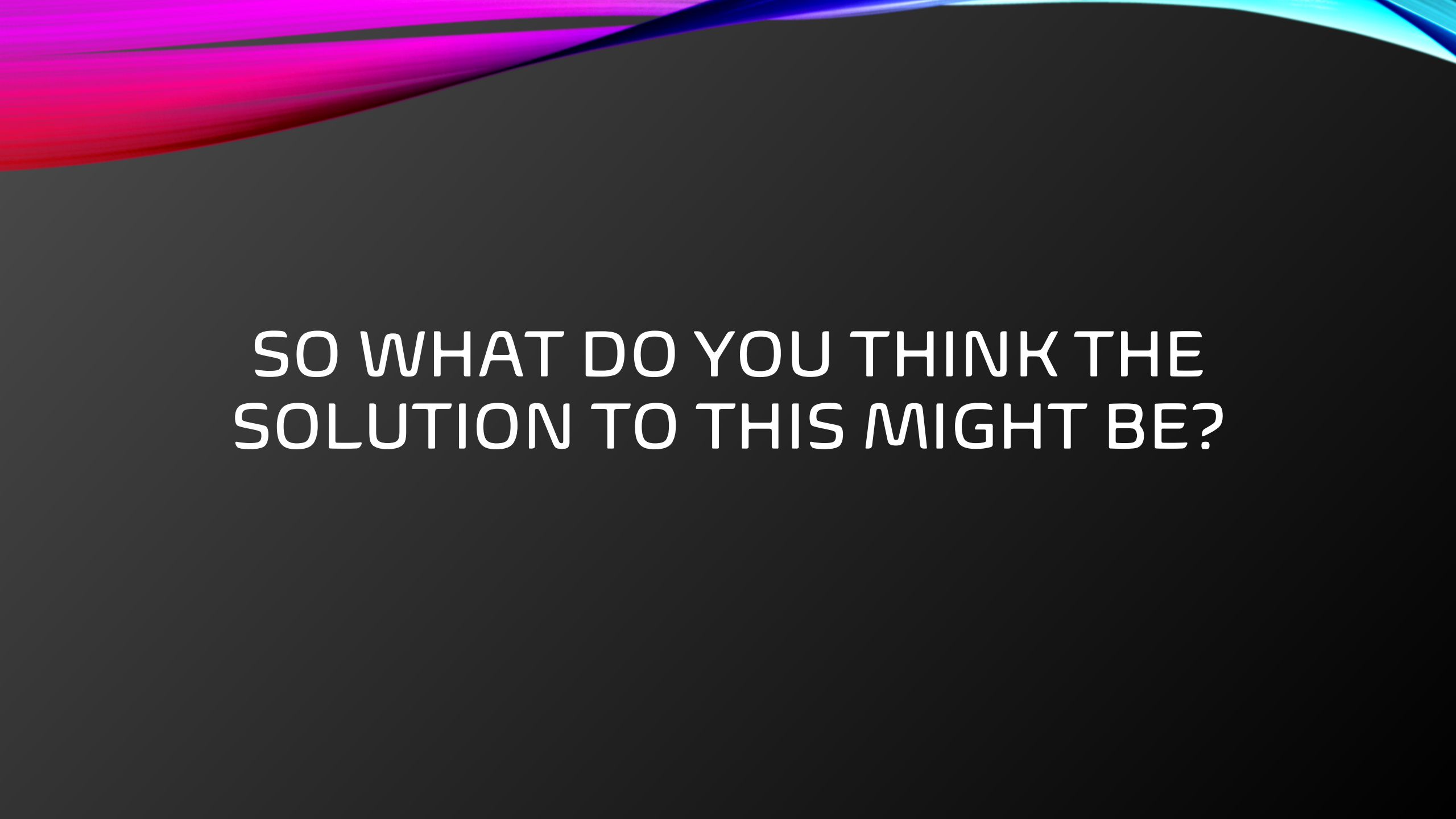
Student

Student ID	Name	Module ID
101	Zayne	CO1409
101	Zayne	CO1507
101	Zayne	CO1301
102	Sarah	CO1409
102	Sarah	CO1301
102	Sarah	CO1605
103	Michael	CO1507
103	Michael	CO1301
103	Michael	CO1605

Module

Module ID	Name	Student ID
CO1409	Programming	101
CO1409	Programming	102
CO1605	Databases	102
CO1605	Databases	103
CO1507	Networking	101
CO1507	Networking	103
CO1301	Games	101
CO1301	Games	102
CO1301	Games	103

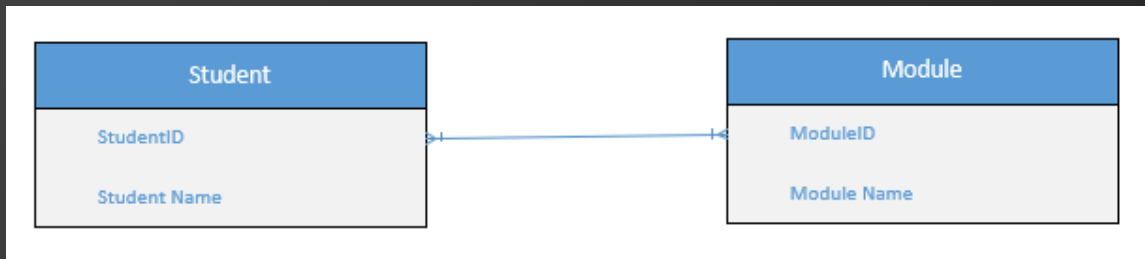
In this situation we are violating the PRIMARY KEY unique constraints.



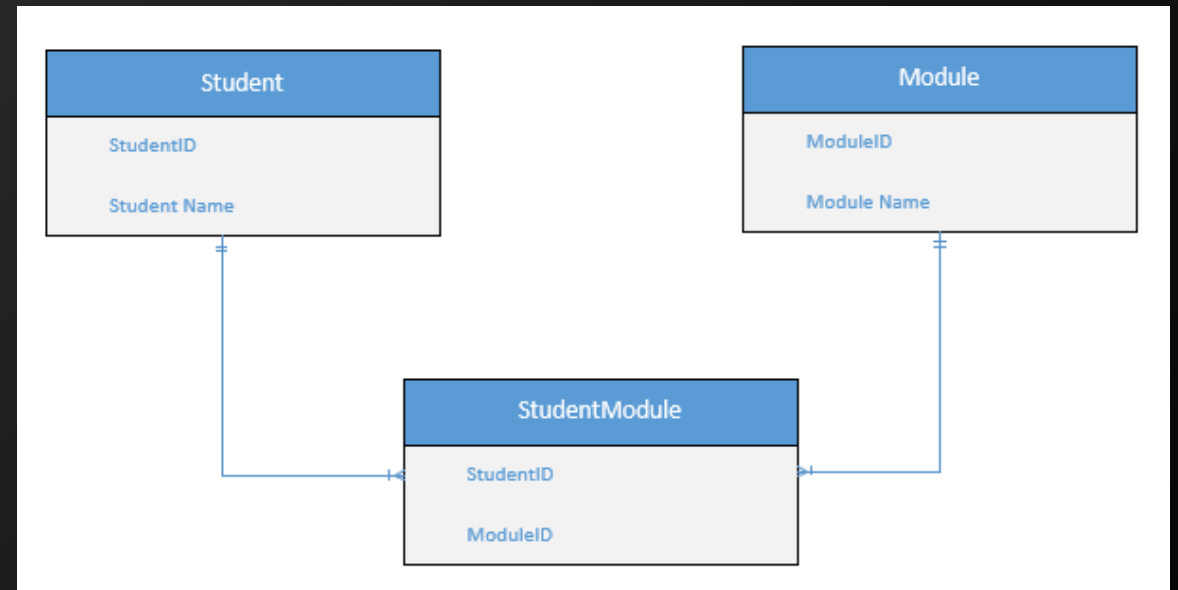
SO WHAT DO YOU THINK THE
SOLUTION TO THIS MIGHT BE?

JOINING TABLE

Before...



After...



The creation of the JOINING table has turned a many-to-many relationship into two one-to-many relationships.

JOINING TABLE

Student

Student ID	Name
101	Zayne
102	Sarah
103	Michael

Module

Module ID	Name
CO1409	Programming
CO1605	Databases
CO1507	Networking
CO1301	Games

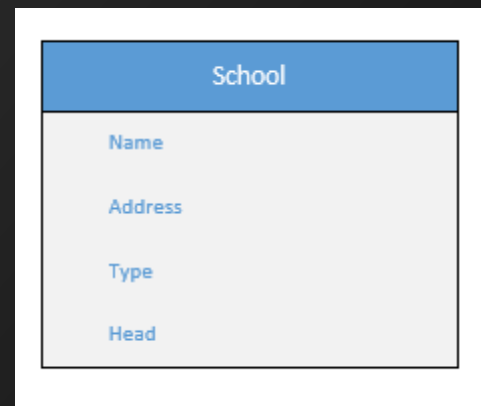
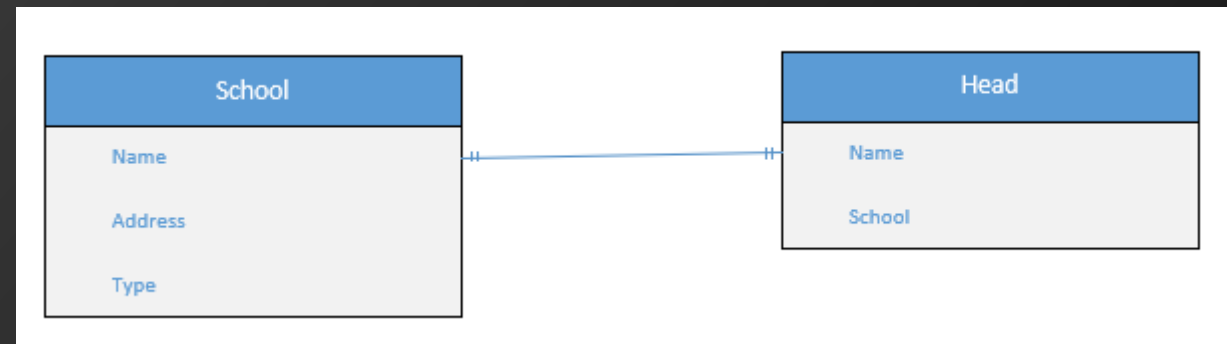
Student_Module

Student ID	Module ID
101	CO1409
101	CO1507
101	CO1301
102	CO1409
102	CO1301
Etc.	Etc.
103	CO1507, CO1301, CO1605

ONE TO ONE RELATIONSHIPS

One School is managed by One Head.

One Head manages One School.



The Head would be better as an attribute of the School entity.

SUMMARY

- Before starting any Entity Relationship Modelling you must have a really good understanding of the application domain.
- Entity Relationship Diagrams model the nature of data within the domain of a database application.
- Entities are identifiable things or concepts that are relevant to the application.
- Relationships show associations between entities.
- Relationships have cardinality – one-to-one, one-to-many, many-to-many.
- Entities have attributes (properties).
- Many-to-many relationships cannot be modelled and should be broken down into two one-to-many relationships.
- Often, one-to-one relationships can be resolved by making one entity an attribute of the other.

RESOURCES

- Richie, C. (2008). *Database Principles and Design*. 3rd Ed. Thompson Learning