**Shape

Description automatically generated with medium confidenceUniversity of Information Technology**

**Homework 02: Entity Relationship Modeling**

****

**DATABASE SYSTEMS DESIGN, IMPLEMENTATION AND MANAGEMENT**

**Instructor: Nguyen Dinh Thuan**

**Students: Group 2**

**March 12, 2023**

**Table of Contents**

[**Question:** 2](#_Toc129557045)

[**1. What two conditions must be met before an entity can be classified as a weak entity? Give an example of a weak entity.** 2](#_Toc129557046)

[**2. What is a strong (or identifying) relationship, and how is it depicted in a Crow’s Foot ERD?** 3](#_Toc129557047)

[**4. What is a composite entity, and when is it used?** 4](#_Toc129557048)

[**5. Suppose you are working within the framework of the conceptual model in Figure Q 4.5.** 6](#_Toc129557049)

[**Problem:** 11](#_Toc129557050)

[**4. Create an ERD based on the Crow’s Foot notation using the following requirements**: 11](#_Toc129557051)

[**References:** 13](#_Toc129557052)

[**Task and member assignment table:** 13](#_Toc129557053)

# **Question:**

## **1. What two conditions must be met before an entity can be classified as a weak entity? Give an example of a weak entity.**

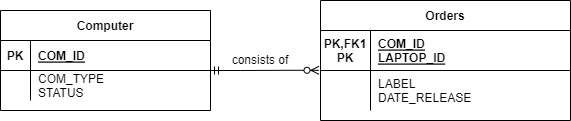
**Answer:**

([1],128) A weak entity is one that meets two conditions:

1. The entity is existence-dependent; it cannot exist without the entity with which it has a relationship.   
2. The entity has a primary key that is partially or totally derived from the parent entity  
in the relationship.

**Example:**

Consider there is a “Laptop” entity in computer-selling shop system. “Laptop” is one of the PC that is derived from “Computer” entity, which mean “Laptop” represents single personal computer and is dependent on “Computer” entity, which represents all personal computers.



**Compare with ChatGPT:**

Graphical user interface, text, application

Description automatically generated

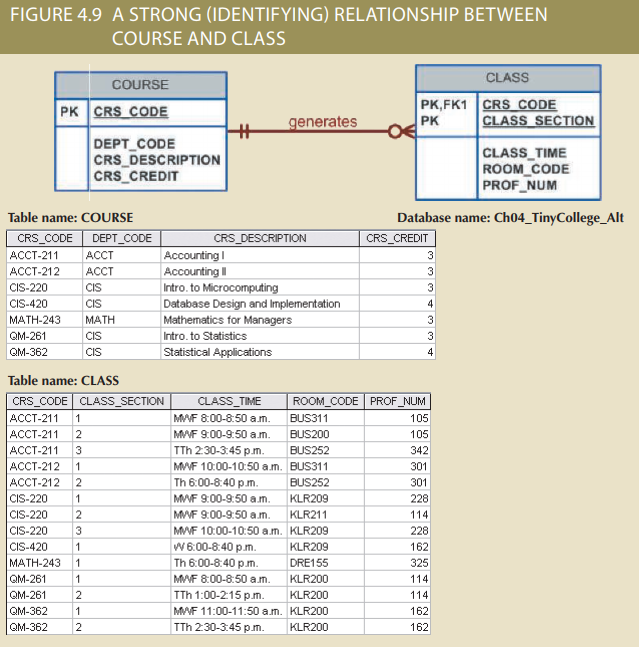
## **2. What is a strong (or identifying) relationship, and how is it depicted in a Crow’s Foot ERD?**

**Answer:**

A relationship that occurs when two entities are existence-dependent; from a database design perspective, this relationship exists whenever the primary key of the related entity contains the primary key of the parent entity.( [1],129)

**For example**, suppose the 1:M relationship between COURSE and CLASS is defined as:  
COURSE (**CRS\_CODE**, DEPT\_CODE, CRS\_DESCRIPTION, CRS\_CREDIT)  
CLASS (**CRS\_CODE, CLASS\_SECTION**, CLASS\_TIME, ROOM\_CODE, PROF\_NUM)  
In this case, the CLASS entity primary key is composed of CRS\_CODE and CLASS\_SECTION. Therefore, a strong relationship exists between COURSE and CLASS because CRS\_CODE (the primary key of the parent entity) is a primary key component in the CLASS entity. In other words, the CLASS primary key did inherit a primary key component from the COURSE entity. (Note that the CRS\_CODE in CLASS is *also* the FK to the COURSE entity). ([1],129)

The Crow’s Foot notation depicts the strong (identifying) relationship with a solid line  
between the entities, as shown in Figure 4.9. ([1],129)



**Compare with ChatGPT:**

Graphical user interface, text, application

Description automatically generated

## **4. What is a composite entity, and when is it used?**

**Answer:**

([1],139) The ER model uses the associative entity to represent an M: N relationship between two or more entities.

This associative entity, also called a composite or bridge entity:

+ Is in a 1:M relationship with the parent entities and is composed of the primary key attributes of each parent entity.

+ May also contain additional attributes that plays no role in the connective process.

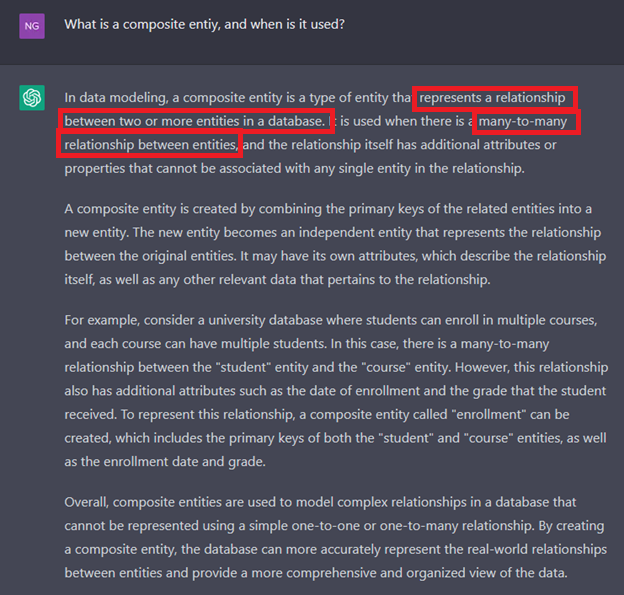
Ảnh có chứa bàn

Mô tả được tạo tự động

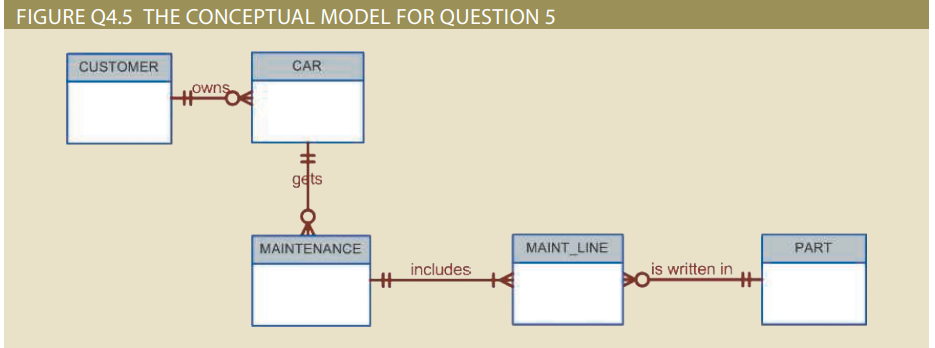
Figure: CONVERTING THE M: N RELATIONSHIP INTO TWO 1:M

In the above Figure there are 2 tables, STUDENT and CLASS. thereby creating a new table with the name ENROLL aka composite entity that is combined from the primary keys in the STUDENT and CLASS tables. in the aggregate entity can also have its own properties independent of other tables.

**Compare with ChatGPT:**



## **5. Suppose you are working within the framework of the conceptual model in Figure Q 4.5.**



*FIGURE Q4.5 THE CONCEPTUAL MODEL FOR QUESTION 5*

Given the conceptual model in Figure Q4.5:

**a. Write the business rules that are reflected in it.**

**Answer:**

The business rules for figure Q4.5 that are reflected in it can be like:

1. A customer owns zero or multiples cars.
2. Each car is owned by one and only one customer.
3. Each car also can have many maintenance records.
4. Each maintenance record is generated by one and only one car.
5. Some cars haven’t generated maintenance record yet.
6. Each maintenance record can use many parts.
7. A part may be used in many maintenance records.
8. Each maintenance record included one or more than one maintenance task.
9. A car part may be written in multiples maintenance lines.

**b. Identify all the cardinalities**.

**Answer:** Before Identifying all the cardinalities, we will examine the term “cardinality “first :

([1],125) Cardinality expresses the minimum and maximum number of entity occurrences associated with one occurrence of the related entity. In the ERD, cardinality is indicated by placing the appropriate numbers beside the entities, using the format (x, y). The first value represents the minimum number of associated entities, while the second value represents the maximum number of associated entities.

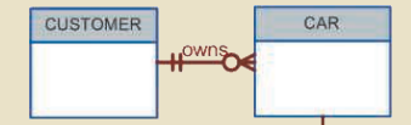
Table

Description automatically generated

*Figure: TABLE 4.3 ([1],132)*

**For example**, to be more clearly as also solving problem 4.b we use the following cuts from *figure Q4.5*

*The first one:*



The relationship between CUSTOMER and CAR in here is ‘own’. The relationship on the right side is denoted as double line segment so we can understand it mean ‘one and only one.’

🡪 The cardinality will be (1, 1) next to the CUSTOMER entity indicates that each car is owned by one and only one CUSTOMER.

The relationship on the left side is denoted as 0 with the three-pronged so we can understand it mean ‘zero or many’.

🡪 The cardinality will be (0, N) next to the CAR entity in the “CUSTOMER owns CAR” indicates that each customer owns from 0 up to many cars.

*The second one:*

Diagram

Description automatically generated

The relationship between CAR and MAINTENANCE in here is ‘gets. The relationship on the right side is denoted as double line segment so we can understand it mean ‘one and only one.’

🡪 The cardinality will be (1, 1) next to the CAR entity indicates that each MAINTENANCE record is gotten by one and only one CAR.

The relationship on the left side is denoted as 0 with the three-pronged so we can understand it mean ‘zero or many’.

🡪 The cardinality will be (0, N) next to the MAINTENANCE entity in the “CAR gets MAINTENANCE” indicates that each CAR gets maintenance record from 0 up to many records.

*The third one:*

Diagram, timeline

Description automatically generated

The relationship between MAINTENANCE and MAINT\_LINE in here is ‘include’. The relationship on the right side is denoted as double line segment so we can understand it mean ‘one and only one.’

🡪 The cardinality will be (1, 1) next to the MAINTENANCE entity indicates that each MAINT\_LINE is included in one and only one MAINTENANCE record.

The relationship on the left side is denoted as line segment with the three-pronged so we can understand it mean ‘one or many’.

🡪 The cardinality will be (1, N) next to the MAINT\_LINE entity in the “MAINTENANCE includes MAINT\_LINE” indicates that each MAINTENANCE record includes at least one MAINT\_LINE.

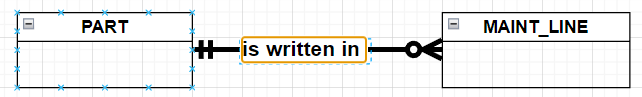
*The last one:*

Diagram

Description automatically generated

*Figure: Original ERD*

We turn this ERD into the following because MAINT\_LINE here is the associative entity between MAINTENCE and PART, so we must draw it like that.



*Figure: After transforming*

The relationship on the left side is denoted as double line segment so we can understand it mean ‘one and only’.

🡪 The cardinality will be (1, 1) next to the PART entity in the “PART is written in MAINT\_LINE” indicates that different MAINT\_LINE may write in for same car PART.

The relationship on the right side is denoted as 0 with the three-pronged so we can understand it mean ‘0 or many.’

🡪 The cardinality will be (0, N) next to the MAINT\_LINE entity indicates that a car PART may be written in by different MAINT\_LINE.

After identifying all the cardinalities, we denoted the figure once again:

Diagram

Description automatically generated

*Figure: Q4.5 After modifying*

**Compare with ChatGPT:**

**a)**

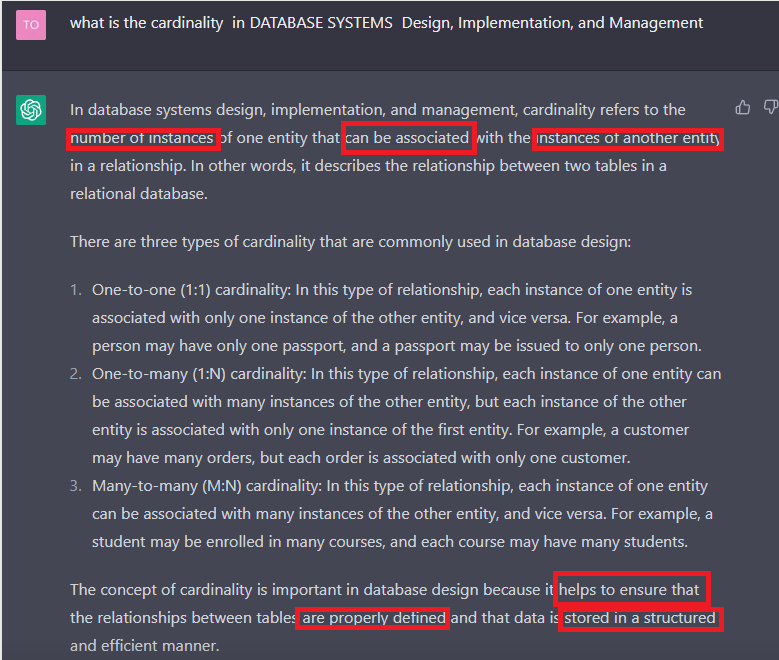
**Text

Description automatically generated**

*Figure: Business Rule*

🡪In general, ChatGPT gives the results from the question almost exactly. Since it does not provide additional cardinalities, there are some differences. After the comparison, ChatGPT tends to give both sides a mandatory relationship but not optional relationship like the Q4.5 given.

**b)**



*Figure: What is the cardinality?*

🡪 When compared to the definition from the textbook, the answer provided by ChatGPT is similar and simple to understand.

classified

# **Problem:**

## **4. Create an ERD based on the Crow’s Foot notation using the following requirements**:

* An INVOICE is written by a SALESREP. Each sales representative can write many invoices, but each invoice is written by a single sales representative.

Shape

Description automatically generated with medium confidence

* The INVOICE is written for a single CUSTOMER. However, each customer can have many invoices.

Shape

Description automatically generated with medium confidence

* An INVOICE can include many detail lines (LINE), each of which describes one product bought by the customer.

Shape

Description automatically generated with medium confidence

* The product information is stored in a PRODUCT entity.

Shape

Description automatically generated with medium confidence

* The product’s vendor information is found in a VENDOR entity.

Shape

Description automatically generated with medium confidence

Final ERD

Graphical user interface, application

Description automatically generated

# **References:**

[1] Database System Design, Implementation, and Management 12th edition

# **Task and member assignment table:**

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **ID** | **Task** | **Status** |
| Nguyễn Minh Thành\* | 20521920 | Q5 | Done |
| Nguyễn Văn Tân | 20521880 | Q4 | Done |
| Quách Vinh Quang | 20521811 | Q1, Q2 | Done |
| Tống Trường Thịn | 20521958 | Problem 4 | Done |

Note \*: Leader