*Ali Masuth*
*IT-214-005*

# Happy Learning Lunch Time!
## Database Design Report 2

This is a continuation of a private database for a local kindergarten, Happy Learning. The kindergarten now aims to keep track of information regarding student lunch.

https://www.freeimages.com/photo/boy-with-lunch-tray-in-school-cafeteria-1890766

# Conceptual Analysis

**Main Entities:**
After reviewing the provided description, I have identified three new entities that now play a role in this database: LUNCH_ITEM, LUNCH_ORDER, and LUNCH_ORDER_ASSIGNMENT.

**Main Relationships:**
By introducing three additional entities into the database, we've also introduced two new relationships.

Firstly, on a daily basis, a caretaker takes on the responsibility of creating many lunch orders for their children, and each order is linked to only one caretaker, establishing a one-to-many relationship known as "CARETAKER orders LUNCH_ORDER."

Secondly, within a lunch order, there can be one or more lunch items, and each lunch item can be associated with one or more lunch orders. This dynamic creates a many-to-many relationship.

Given that this relationship is many-to-many, we need to establish a bridge entity. This bridge entity is called "LUNCH_ORDER_ASSIGNMENT," which contains both the lunch order ID and the lunch item code as its composite primary key.

Furthermore, it's important to note that in a many-to-many relationship, we're effectively connecting two one-to-many relationships with each other. Specifically, "LUNCH_ORDER" is connected to "LUNCH_ORDER_ASSIGNMENT" as a one-to-many relationship, and "LUNCH_ITEM" is similarly linked to "LUNCH_ORDER_ASSIGNMENT" as a one-to-many relationship. Both relationships use the caption “is assigned to.”

**Relationship Type:**
The relationship between CARETAKER and LUNCH_ORDER is considered a weak relationship since the primary key of CARETAKER is not the primary key of LUNCH_ORDER. Each lunch order has its own unique identification, so it wouldn't be logical for an order, which is meant to be distinct, to share its primary key with that of the caretaker.

In the case of the many-to-many relationship between LUNCH_ORDER and LUNCH_ITEM, the relationship is strong. This is because the primary keys of the bridge entity are the primary keys of the individual entities. As a rule of thumb, it's important to note that all many-to-many relationships are inherently strong.

## Entities Analysis

**Name:**
LUNCH_ITEM

**Description:**
The lunch menu will consist of 25 lunch items, each of which will have the following attributes: a 4-character code, a name, and the associated calorie count.

**Attributes:**

| **Name:** | **Domain & Size** | **Simple/ Composite** | **Single/ Multi-Value** | **Required/ Optional** | **Unique/ Duplicated** |
|---|---|---|---|---|---|
| CODE | CHAR(4) | Simple | Single-Value | Required | Unique |
| NAME | VARCHAR (25) | Simple | Single-Value | Required | Unique |
| CALORIES | INT | Simple | Single-Value | Required | Duplicated |

For this entity, a simple primary key was provided, which was a 4-character unique code.

No foreign keys were used within this entity; however, the primary key of the entity served as a foreign key for its bridge entity "LUNCH_ORDER_ASSIGNMENT."

---

**Name:**
LUNCH_ORDER

**Description:**
The lunch order entity's purpose is to record student orders. Each order is assigned a unique ID for easy tracking, and it also includes the caretaker's ID. Although an entity that tracks orders would contain foreign keys that refer to some specific lunch items. In this case, there are no such foreign keys. This is because the bridge entity already handles this relationship, as the connection between a lunch order and a lunch item is already a many-to-many relationship.

**Attributes:**

| **Name:** | **Domain & Size** | **Simple/ Composite** | **Single/ Multi-Value** | **Required/ Optional** | **Unique/ Duplicated** |
|---|---|---|---|---|---|
| ID | INT | Simple | Single-Value | Required | Unique |
| CARETAKER_ ID | INT | Simple | Single-Value | Required | Duplicated |

As just mentioned, a surrogate primary key, “LUNCH_ORDER_ID," was included in this entity to avoid potential data anomalies.

As for foreign keys included, all lunch orders must have a caretaker’s ID and a lunch item’s code is included in the bridge entity.

___________

**Name:**
LUNCH_ORDER_ASSIGNMENT (Bridge Entity)

**Description:**
A bridge entity that was made to connect a person’s lunch order and the lunch item they have purchased. Both the lunch order ID and the 4-character lunch item code are included in this entity.

**Attributes:**

| **Name:** | **Domain & Size** | **Simple/ Composite** | **Single/ Multi-Value** | **Required/ Optional** | **Unique/ Duplicated** |
|---|---|---|---|---|---|
| LUNCH_ORDER _ID | INT | Simple | Multi-Value | Required | Duplicated |
| LUNCH_ITEM_ CODE | CHAR(4) | Simple | Multi-Value | Required | Duplicated |

## Relationship Analysis

**Relationship Name:**
CARETAKER orders LUNCH_ORDER

**Relationship Type:**
This relationship is defined as a one-to-many relationship because a caretaker can place multiple lunch orders for their children while each individual lunch order is exclusively linked to a single caretaker.

**Relationship Strength:**
The primary key of a lunch order is not the primary key of the caretaker. Therefore, this relationship is a weak relationship.

**Entities Participation:**
The participation of the caretaker entity is mandatory because a lunch order cannot be made without a caretaker. The lunch order entity is mandatory as well because even if a caretaker skips a day without placing an order, the order from the previous day will instead be given to the child.

**Special Cardinality:**
There are no special cardinalities to be reported.

**Foreign Keys:**
The lunch order references the caretaker's ID as its foreign key because, according to the given description, the order must store the caretaker who made the order.

**Relationship Name:**
LUNCH_ORDER is assigned to LUNCH_ORDER_ASSIGNMENT and LUNCH_ITEM is assigned to LUNCH_ORDER_ASSIGNMENT

**Relationship Type:**
This relationship is a many-to-many relationship because one order will contain many lunch items, and one lunch item will be associated with many orders.

**Relationship Strength:**
This is a strong relationship because the primary keys of the two entities are featured in their bridge entity.

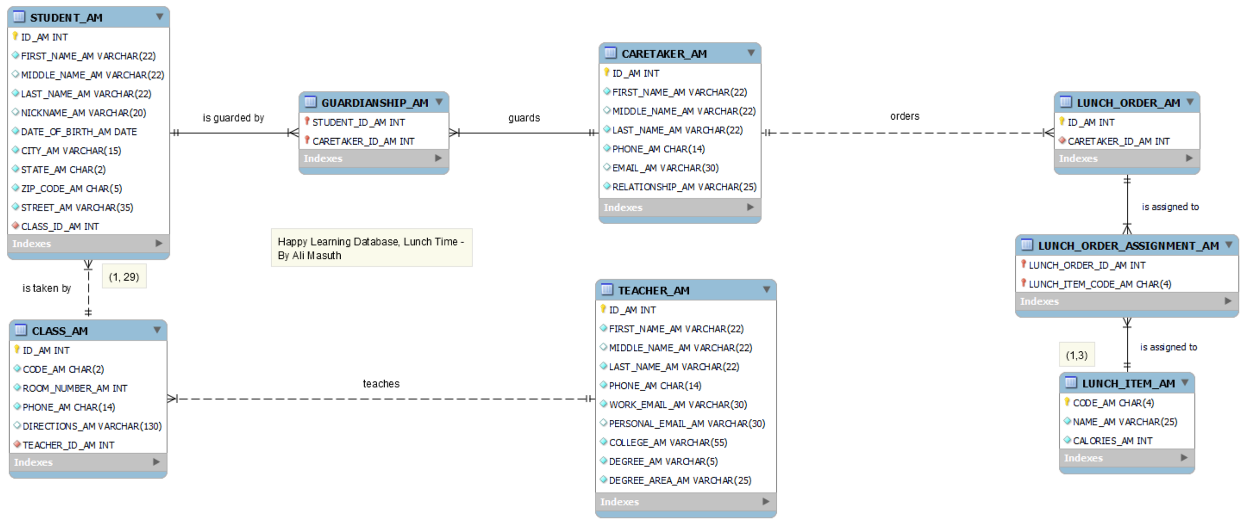
**Entities Participation:**
Both entities are required to participate in this relationship because a lunch order cannot exist without having lunch items, and a lunch item cannot be sold without being part of an order.

**Special Cardinality:**
The cardinality of the lunch item entity is (1, 3) because one order consists of a maximum of 3 items.

**Foreign Keys:**
The lunch order ID and the lunch item code are used in the bridge entity.

# Entity Relationship Diagram

## Main Screenshot:



## Entity Screenshots:

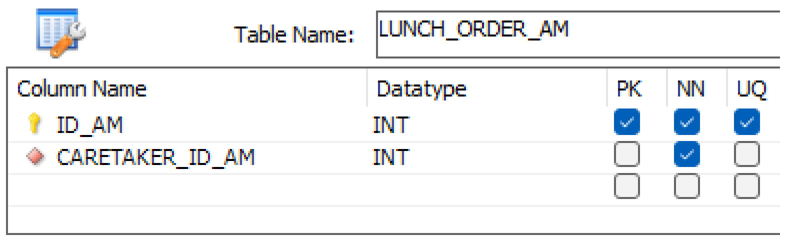


Table Name: LUNCH_ORDER_AM

| Column Name | Datatype | PK | NN | UQ |
|---|---|---|---|---|
| ID_AM | INT | ✓ | ✓ | ✓ |
| CARETAKER_ID_AM | INT | ☐ | ✓ | ☐ |
| | | ☐ | ☐ | ☐ |

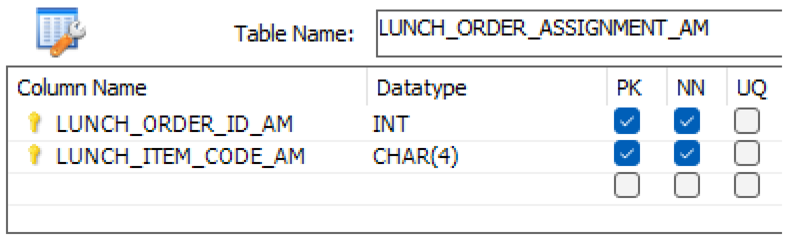


Table Name: LUNCH_ORDER_ASSIGNMENT_AM

| Column Name | Datatype | PK | NN | UQ |
|---|---|---|---|---|
| LUNCH_ORDER_ID_AM | INT | ✓ | ✓ | ☐ |
| LUNCH_ITEM_CODE_AM | CHAR(4) | ✓ | ✓ | ☐ |
| | | ☐ | ☐ | ☐ |

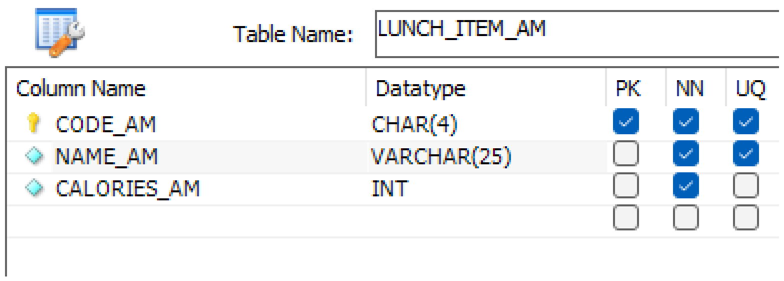


Table Name: LUNCH_ITEM_AM

| Column Name | Datatype | PK | NN | UQ |
|---|---|---|---|---|
| CODE_AM | CHAR(4) | ✓ | ✓ | ✓ |
| NAME_AM | VARCHAR(25) | ☐ | ✓ | ✓ |
| CALORIES_AM | INT | ☐ | ✓ | ☐ |
| | | ☐ | ☐ | ☐ |