**[Discussion 5A: The relational data model and relational database constraints](https://blackboard.jhu.edu/webapps/blackboard/content/launchLink.jsp?course_id=_207662_1&content_id=_7973838_1&mode=view)**

Please provide your thoughts related to the following questions:

* Why are tuples in a relation not ordered? Why are duplicate tuples not allowed in a relation?
* Discuss the various reasons that lead to the occurrence of NULL values in relations.
* Discuss the entity integrity and referential integrity constraints. Why is each considered important?
* Define foreign key.
* What is this concept used for?

Relational databases are based off the relational model of data, which is based off the concept of a relation. A database can be represented as a collection of relations in the relational model, where each relation is a table of values, with each value in a row being assigned to a key in a tuple. Each tuple in a relation is unordered, greatly simplifying the process for adding and deleting tuples (as if the tuples were ordered you would have to keep track of the index of each tuple in the relation). Further, duplicate tuples are not allowed in a relation since it violates the specifications of the relational integrity, specifically the constraint that no two tuples can have the same primary key values at any relational state of the database.

There are a few instances where a tuple may have a NULL value in a relation. NULL may indicate that no value has been specified, one example may be a new customer on a site may have not set up their profile completely. Another instance where NULL may appear in a relation is if the value is unknown, perhaps in the case of an optional field that was not filled out, or in a result column where the result is still being computed. Another instance where you may see NULL in a relation is if the value is not applicable, perhaps in the case where you have a relation that stores car sales, one value in the tuple may be cosigner, however not every car needs a cosigner. These are a few cases where NULL may be a valid attribute in a tuple.

There are some constraints to relations that must be considered. The first is the entity integrity constraint, which specifies that no primary key can be null. If the primary key was null, then the tuple could not be identified, hence why the entity integrity constraint exists. Another is the referential integrity constraint, which specifies that the primary key of a parent relation migrates to a child relation as a foreign key to maintain referential integrity between the two relations. If there was no foreign key on the child entity, there would be no way to track the parent in the parent relation, which is why the referential integrity constraint is important to tracking child entities in a relation.

The foreign key is a referencing relation that references the primary key of the referenced relation. For example, if you are selling cars, you may have a relation that describes each car sold, and a relation that describes each of the customers. If a car is sold to child entity customer, then the customer would get the primary key of the car sold added as a foreign key. This allows the database user to track which cars get sold to which customers.

[**Discussion 5B: Your experience in Experience in the Relational Data Model, Relational Database Constraints, and Relational Algebra**](https://blackboard.jhu.edu/webapps/blackboard/content/launchLink.jsp?course_id=_207662_1&content_id=_7973839_1&mode=view)

Please provide your experience related to the following questions:

* Any experience in Relational Database Constraints. If yes, how did you specify these constraints?
* Any experience in Relational Algebra used in the past?
* Any differences between relational algebra operations and relational languages SQL?
* Any experience on query optimizer at work?
* If you don’t have any experience, please provide scenarios in your database project that you will incorporate various relational database constraints.

As someone new to databases, I have no experience with relational database constraints, no experience with relational algebra, no experiences with SQL, and no experience with query optimizers at work. Thus there should be much to learn for me as I progress through this class.

In lieu of applicable database experience, I will discuss some places in the database project where I will likely use some of the various relational database constraints. For reference, I am planning on doing a database project which will store team and player statistics for a hockey league, and in preparation for the database project I will try to have examples which apply directly to my personal project. There are four types of constraints:

* Domain Constraint: The domain constraint specifies a data type and range associated with a domain. For example, in hockey a player must wear a number between 1 and 99. Thus, in the field for player number, the value must be an integer between 1 and 99. For first and last name for each player, we can specify a constraint of 30 characters, since names do not often surpass 30 characters.
* Key Constraints and Constraints on Null: Key constraints include the constraint that no two tuples can have the same unique identifier or primary key. To satisfy this constraint, for entities such as player and team, on player/team creation each entity should be assigned a unique key as an identifier to satisfy the key constraint. Some attributes may not accept a null value. Attributes relating to statistics for a given player for a given season should not accept a null value. Statistics like games played, goals, penalty minutes, etc. should all have numeric values of 0 or greater, and keys such as these should be initialized to 0 on player creation.
* Entity Integrity Constraints: The entity integrity constraint specifies that no primary key can be null. As discussed above, as each player/team is created, a unique identifier should be created for the player/team, and specifying that on creation will maintain the entity integrity constraint.
* Referential Integrity Constraints: The referential integrity constraint specifies that the primary key of a parent relation should migrate to the child relation as a foreign key to maintain the referential integrity between the two relations. One example of this specific to my database project is each player (child) tuple should have a foreign key specifying which team (parent) they are a part of. This will make it simple to aggregate which players play for which team.