Chapter 2: Relationships

Lab: Many-to-Many Relationship

Back to the Question

Correct Options:

- **stores.address.street** and **items.description**

  There is one given street address for a store and one description for an item.

  However, the **stores** and **items** entities have a Many-to-Many relationship per the Crow's Foot symbol. To be more precise, it says that a given item can be sold at 0 to Many stores (as represented by the **items.sold_at** array), and a store can sell 0 to Many items (as represented by one **stores._id** being stored in 0 to Many **items.sold_at** arrays).

  The two directional One-to-Many readings give us a Many-to-Many relationship.

  In this case, linking is used to establish the relationship.

- **items.sold_at** and **items.reviews.body**

  From an **items** point of view there are possibly *Many* values of **sold_at** and *Many* values of **reviews.body**.

  So going from one of those two fields to another, we will traverse a Many-to-One relationship, then a One-to-Many relationship, making the whole relationship a Many-to-Many relationship.

  Another way to look at it is to understand that a given review will be indirectly linked to all the stores that sell the product and that a given store has multiple reviews pertaining to the products it sells.

- **users.credit_cards.number** and **items.reviews.body**

  A user writes many reviews.

  Reading the relationship from the other end, a review is written by only one user, however, there are many credit cards identifying this user, due to the fact that a user has an array of credit cards. So a review for a given user can be can be indirectly linked to many credit cards.

  Those two readings give us again a Many-to-Many relationship between those 2 fields.

One thing to understand from all of the above examples of Many-to-Many relationships is that traversing a relationship through either a noted One-to-Many CRD relationship or through entering an array will identify a directional relationship as One-to-Many.

This means that you don't really need to understand the domain of a system to identify potential One-to-Many relationships. Reading the relationships between collections and accounting for the traversal of arrays is enough to identify relationships.

Getting One-to-Many relationships from both directions will identify the relationship as a Many-to-Many relationship.

Incorrect Options:

- **items.title** and **items.reviews.body**

  A given item, identified by its title here, has many reviews, represented here by an array.

  However, a given review only applies to one item.

  This means we have a One-to-Many relationship between the **items.title** and **items.reviews.body**. Going from the **items.reviews.body** to the **items.title** we walk out of the array, not in, so this is One-to-One in this direction, meaning for a given **reviews.body** there is only one **product.description** to which it applies.

- **users.shipping_address.street** and **items.reviews.body**

  For a given **users.shipping_address.street** there are many related **items.reviews.body** values, so One-to-Many from this direction.

  The **items.reviews.body** values are associated to a single user, which has a single shipping address, leading to a One-to-One relationship in this opposite direction.

  Overall, the relationship between **users.shipping_address.street** and **items.reviews.body** is One-to-Many.

Proceed to next section