

---

CAT THINGS

---

# Database Design

---

Nicholas McHale, Andrew McCann, Susmita Awasthi,  
Manpreet Bahl, Austen Ruzicka, Luke Kazmierowicz,  
Hillman Chen

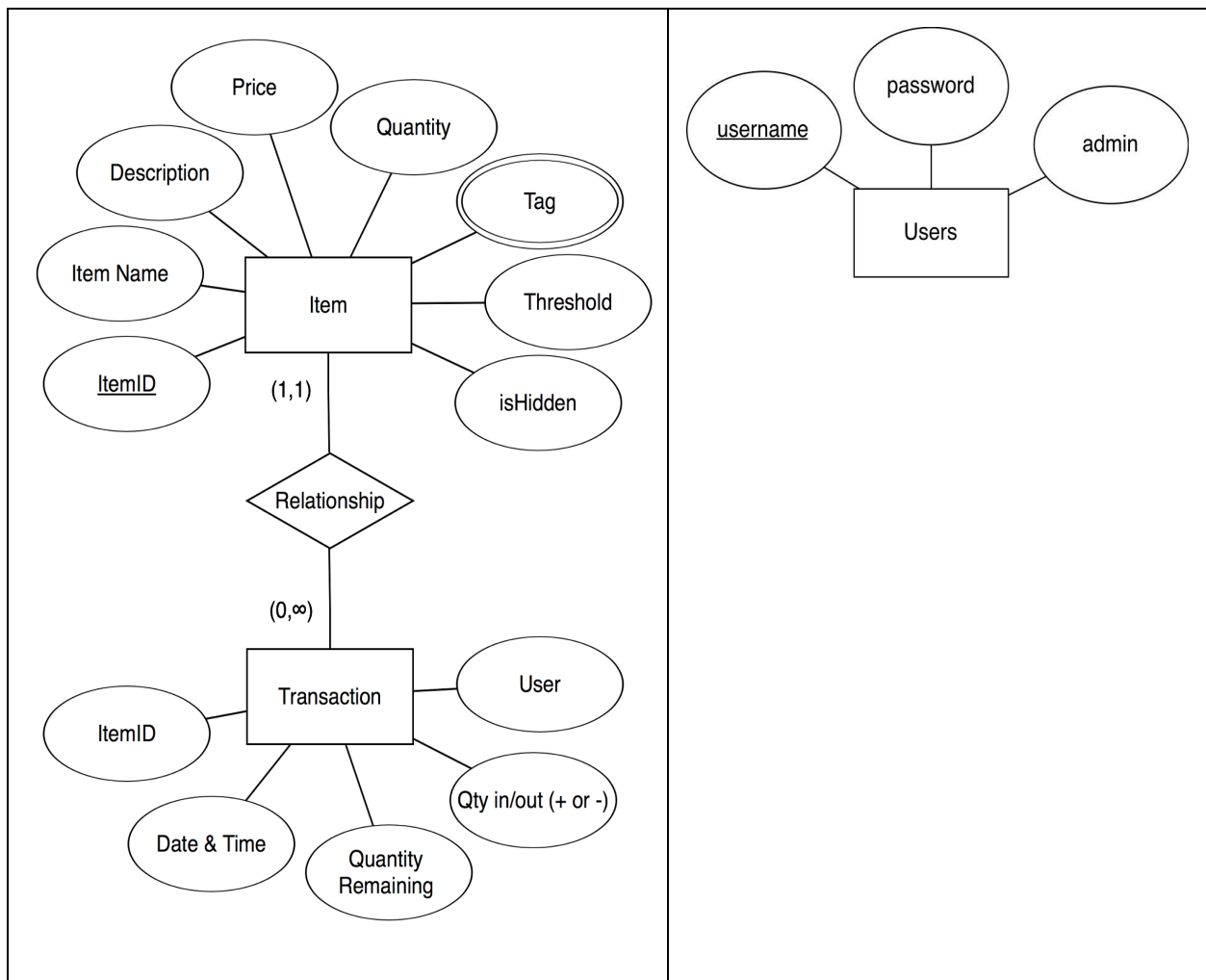
---

# Introduction

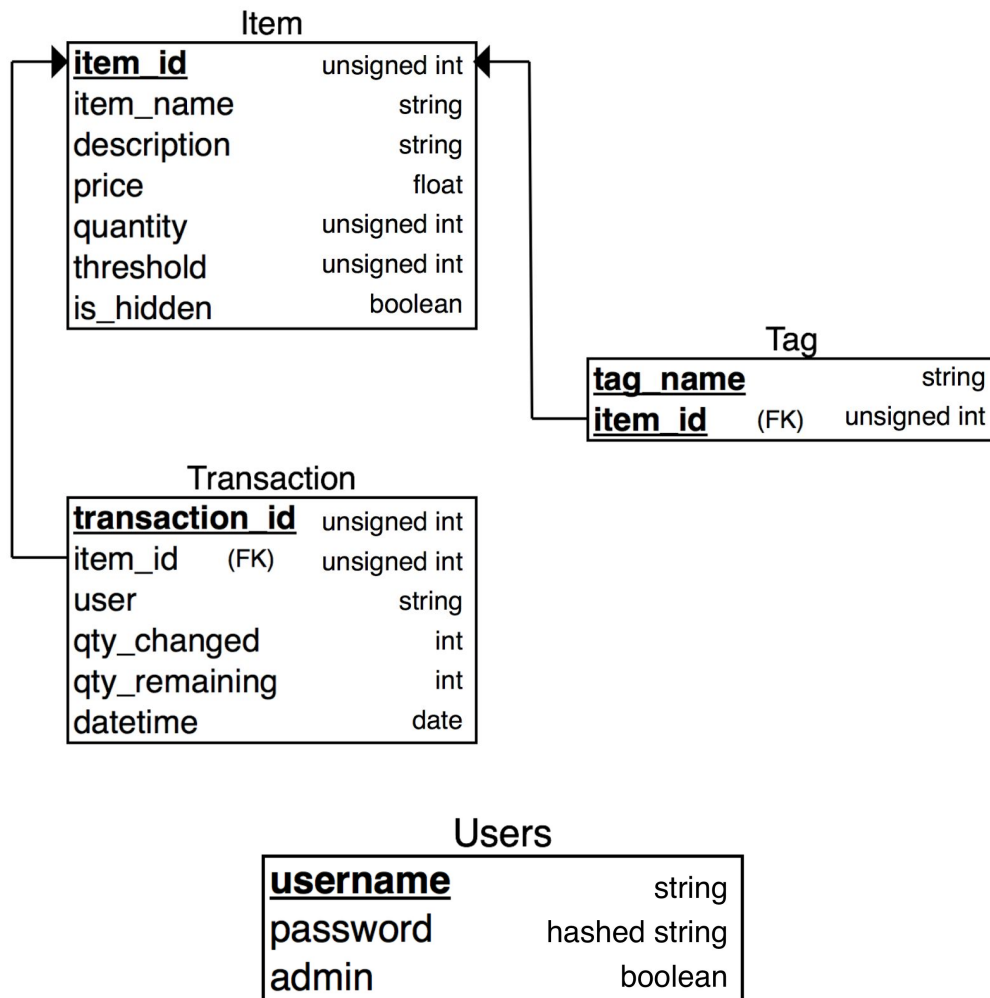
A database is needed to accurately track the entirety of the C.A.T.'s inventory. The data stored within will be item names, their quantities, their tags, etc. Updates may occur as frequently as several times a day, or less as relates to the C.A.T.'s needs. There will be no special requirements for security since the data is not something that needs to be kept confidential, except passwords will be provided so that a C.A.T. member might not accidentally compromise a regular password of theirs.

## Schema

### Entity Relational Diagram



## EER Diagram



## Table Descriptions

### Item Table

Attribute Name	Type	Key	Unique	Usage
item_id	unsigned int	YES	YES	Uniquely identifies each item in the database.
item_name	string	NO	NO	Name of the item. For user to identify each item.
description	string	NO	NO	A brief description of the item.
price	float	NO	NO	The price of the item when it was last purchased.
quantity	unsigned int	NO	NO	The number of such items currently in stock.

<b>threshold</b>	unsigned int	NO	NO	When the quantity drops below threshold, the admin is informed.
<b>is_hidden</b>	boolean	NO	NO	To mark whether to display the item to the user or not. Is used instead of deleting items.

## Transaction Table

Attribute Name	Type	Key	Unique	Usage
<b>transaction_id</b>	unsigned int	YES	YES	Uniquely identifies each item in the database.
<b>item_id</b>	unsigned int	NO	YES	Foreign key to Item table. Defines what item the transaction is dealing with.
<b>user</b>	string	NO	NO	The name of the user that checked in/out the item.
<b>qty_changed</b>	int	NO	NO	The number of items checked in or out of the database.
<b>qty_remaining</b>	unsigned int	NO	NO	The number of items in the database immediately following the transaction.
<b>is_hidden</b>	boolean	NO	NO	To mark whether to display the item to the user or not. Is used instead of deleting items.

## Tag Table

Attribute Name	Type	Key	Unique	Usage
<b>tag_name</b>	string	YES	YES	Uniquely identifies each tag in the database. Each item is related to one or more tags.
<b>item_id</b>	unsigned int	NO	YES	Foreign key to Item table. Defines what item the transaction is dealing with.

## User Table

Attribute Name	Type	Key	Unique	Usage
<b>username</b>	string	YES	YES	Uniquely identifies each user in the database.
<b>password</b>	unsigned int	NO	NO	A hashed char array to verify users are who they say they are.
<b>admin</b>	boolean	NO	NO	If true then user has admin privileges, otherwise they have only standard privileges.

In addition to the three tables described above, the database also has triggers and views that make it much easier to work with. When a row is inserted into the *transaction* table a function is triggered that does the following two things:

1. Adds the *qty\_changed* field of the new transaction to the *quantity* field of the items table. If the *qty\_changed* field is negative then the quantity in the items table is reduced by that amount.
2. The *qty\_remaining* field in the new transaction is set to the value of the updated *quantity* field of the item in the items table.

The database also uses two views to simplify retrieving data from it. The views act as the following tables:

1. An *items* view is created that returns every row from the *all\_items* table where the *is\_hidden* field is false. This way, when an item is marked as hidden it will never appear in any queries that reference the *items* view.
2. A *checkout\_per\_day* view is created that returns the *item\_name*, *item\_id* and the average quantity checked out for each day of the week averaged over the last 3 months. For days that no items were checked out it returns 0 for that column.

In case these are needed:

