

Outline:

For this final project I decided to design a database that contains Items with their content type and the company that it was made by. These Items are available in different stores that you can look at. This is like a normal search engine basically, and it be useful because technically it would really help a customer to be able to find out where he/she can buy these items from. Items could be added or removed from the store list.

Database Outline In Words:

We have this search engine that contains bunch of items that could be find in different stores.

Every Item has an id, name, and price.

Every item has exactly one type, with a Type Id and the type name. Example: Gaming, Appliances

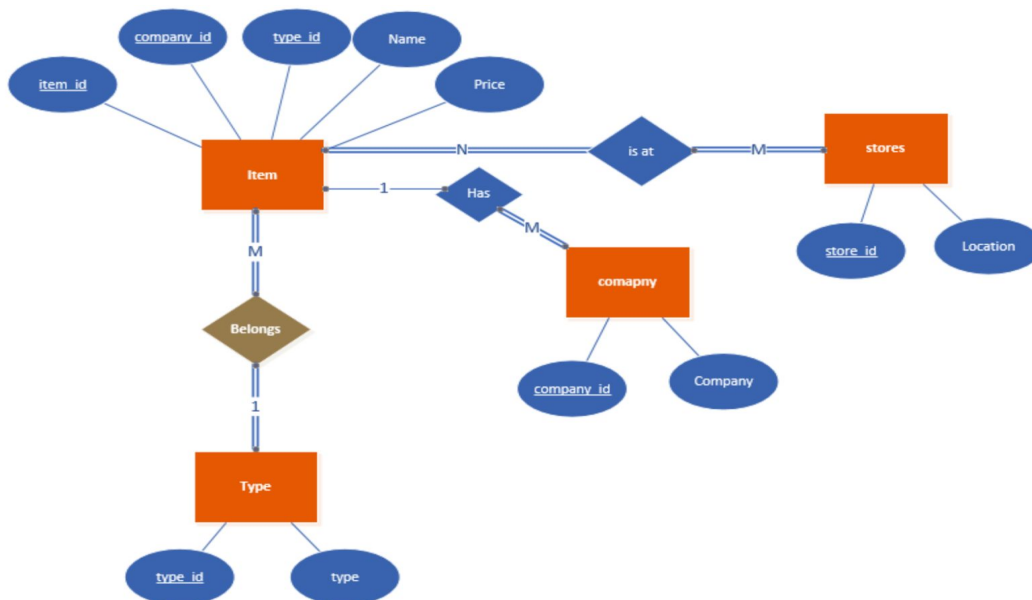
Every Item has exactly one company and every company has many items

Every Item also belongs to at least one store with a unique store Id, store location and a description for that store and every store has many items.

Different stores and types and company could be added to the database to be able to create more custom Items.

Every Item can be added to a different store with a unique description or removed from the store.

ER Diagram:



Database Schema:

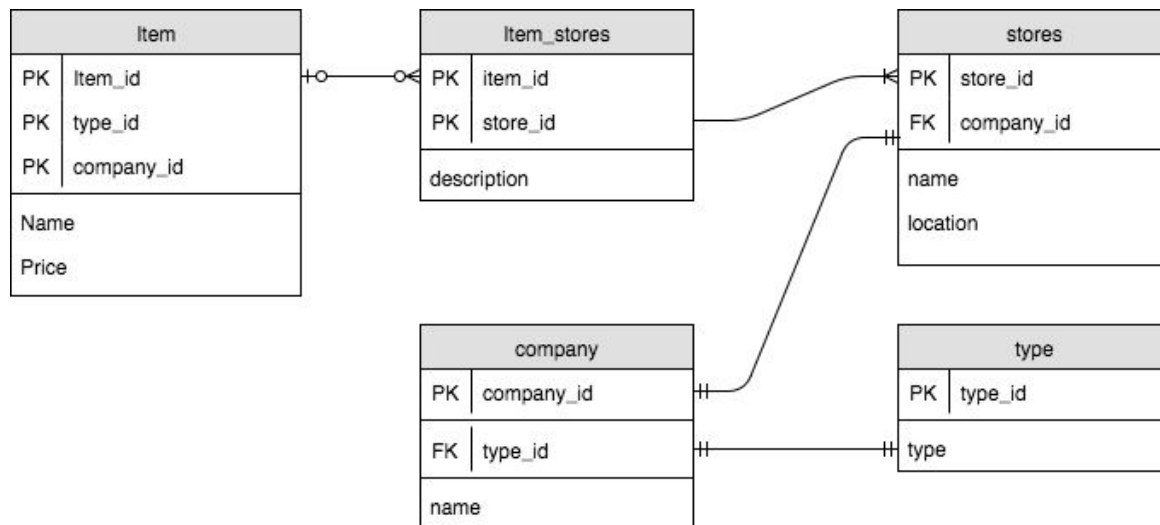


Table Creation Queries

```
-- Creating the Item table with 40 items in there
CREATE TABLE item (
  item_id INT AUTO_INCREMENT,
  type_id INT,
  company_id INT,
  price FLOAT,
  name VARCHAR(50) NOT NULL,
  CONSTRAINT item_unique_name UNIQUE (name),
  PRIMARY KEY(item_id, type_id, company_id)
)ENGINE=InnoDB;

-- Item Type table to distinguish between different types
CREATE TABLE type (
  type_id INT AUTO_INCREMENT PRIMARY KEY,
  type VARCHAR(50) NOT NULL,
  CONSTRAINT type_unique_name UNIQUE (type)
)ENGINE=InnoDB;

-- Company Table for every company that makes the Item
CREATE TABLE company (
  company_id INT AUTO_INCREMENT PRIMARY KEY,
  type_id INT,
  name VARCHAR(50) NOT NULL,
  CONSTRAINT company_unique_name UNIQUE (name),
  FOREIGN KEY(type_id) REFERENCES type(type_id)
)ENGINE=InnoDB;

-- Store table for stores that have those items
CREATE TABLE stores (
  store_id INT AUTO_INCREMENT PRIMARY KEY,
  company_id INT,
  name VARCHAR(50) NOT NULL,
  location VARCHAR(100) NOT NULL,
  CONSTRAINT stores_unique_name UNIQUE (name),
  FOREIGN KEY(company_id) REFERENCES company(company_id)
)ENGINE=InnoDB;

-- Includes the relationship between stores and the Items
CREATE TABLE item_stores (
  item_id INT,
  store_id INT,
  description VARCHAR(500),
  PRIMARY KEY(item_id, store_id)
)ENGINE=InnoDB;
```

General User Queries

```
INSERT INTO company(name) VALUES ([company]);
```

```
INSERT INTO item(type_id, company_id, price, name) VALUES ([type],[company],[price],[name]);
```

```
INSERT INTO stores(name, location) VALUES ([store],[location]);
```

```
INSERT INTO type(type) VALUES ([type]);
```

```
INSERT item_stores(item_id, store_id, description) VALUES ([item_id],[store_id],[description]);
```

```
DELETE FROM item_stores WHERE item_id = [item_id] AND store_id = [store_id];
```

```
SELECT company_id, name FROM company;
```

```
SELECT type_id, type FROM type;
```

```
SELECT item_id, name FROM item;
```

```
SELECT store_id, name FROM stores;
```

```
SELECT DISTINCT i.name, i.price, s.name, im.description FROM item i  
INNER JOIN item_stores im ON im.item_id = i.item_id  
INNER JOIN stores s ON s.store_id = im.store_id  
WHERE i.type_id = [item type_id];
```

```
SELECT i.name, i.price, s.name, im.description, c.name, t.type FROM item i  
INNER JOIN item_stores im ON im.item_id = i.item_id  
INNER JOIN stores s ON s.store_id = im.store_id  
INNER JOIN company c ON c.company_id = i.company_id  
INNER JOIN type t ON t.type_id = [type_id];
```