Supernatural Wardrobe

Team Members: Alexa Castro & Carolina Rodriguez
Group: 36
URL: http://flip2.engr.oregonstate.edu:49275/

Executive Summary:

The Supernatural Wardrobe started as a fun idea to now a fun working business idea. It has gone through many changes throughout the feedback from our peers and TA. The end result of our database has been modified for the client to search and shop through our inventory of costumes.

In the beginning, we presented our database outline with five entities and a M:M relationship between costumes and orders. An intersection table came into place to support the M:M relationship. The costume orders table provided an overview of how many orders of a specific costume would be purchased. Since we stated in our overview that we supply over 500 franchise companies, large bulk order data would be our priority.

Our peers gave us very constructive feedback from the very start. In the step 1 draft, we made major changes such as adding additional information to relationships in our outline. We made sure that unique constraints were removed from attributes that were listed as primary keys. This would interfere with the actual primary key that handles the unique constraint.

In the step 2 draft, we changed the relationship between inventory and costumes to 1:1. This would allow each inventory id to correspond to one costume id. For the most part, all entity relationships were established. Most of our changes were minor usability changes that would make our database run correctly. The tables were more defined with detailed attributes such as adding qty_ordered to costume_orders and company_name to companies. We were able to generate sample data of all entities through SQL queries.

In the step 3 draft, the website was live and our sql database tables generated the correct data. The UI implementation was available for each page. A table was created for each entity for future use of CRUD implementation. Additional queries were added to our DML file such as SELECT query for costume_orders, as our peers were not able to display an order.

As we entered the last two final drafts, the CRUD functionalities were implemented. Each page had a form to add or update data. There were pre filled values in the forms and a search function is available. Our peers were able to implement CRUD functions and their feedback let us know that our page is UI friendly. We were blocked in our intersection table to update the quantity which was linked to the orders table. Luckily, one of our peers suggested adding CRUD implementations to our intersection table and updating the orders table by summing the costume price x quantity.

All feedback and revisions helped us improve our overall functionality and create the UI user friendly. The final product is a database application that performs CRUD implementations for a costume warehouse. The application will provide data of costumes, orders, inventory, and companies.

Project Outline:

Supernatural Wardrobe is a distribution center for Halloween costumes. Consumers spend billions of dollars each year on Halloween costumes. We supply Halloween costumes to over 500 franchise companies with a large selection of costumes by allowing them to search through our inventory and purchase costumes in bulk. We make an estimated revenue of $1.2 billion in sales. On average each company orders 100 units that contain over 5,000 costumes. Each costume will be categorized by theme allowing companies to order multiple costumes in bulk. A database will allow us to record each order to keep track of our supply chain and order fulfillment. The database will also allow us to keep track of inventory and the companies we work with for a more streamlined process.

Supernatural Wardrobe will have the capability to allow a company to order in bulk and to choose their costume themes. The database will provide details on the costume orders, quantity ordered, and the classification of each order by costume and theme name.

Database Outline

costumes: records the details of the costumes in our system

- costume_id: int, auto_increment, not NULL, PK
- costume_name: varchar(150), not NULL
- price: decimal(18, 2), not NULL
- costume_description: varchar(750), not NULL
- theme_id: int, not NULL, FK
- Relationship:
    - M:N between costumes and orders with costume_orders acting as an intersection table.
    - M:1 between costume and themes with theme_id as a FK inside of costumes. It's mandatory for a costume to have a costume theme and for a costume theme to be associated with a costume.
    - 1:1 between costumes and inventory with inventory_id as a FK inside of costumes. It's possible for there to be 0 stock of a certain costume in inventory.

themes: list of themes that a costume can have

- theme_id: int, not NULL, PK
- theme_description: varchar(150), not NULL
- Relationship:
    - 1:M between costume and themes with theme_id as a FK inside of costumes. It's mandatory for a costume to have a costume theme and for a costume theme to be associated with a costume.

inventory: records the amount of costumes in our system

- inventory_id: int, auto_increment, not NULL, PK
- stock: int, DEFAULT NULL, 0/None is out of stock
- inventory_description: varchar(150), not NULL
- costume_id: int, not NULL, FK
- Relationship:
    - 1:1 between costumes and inventory with costume_id as a FK inside of inventory. It's possible for there to be 0 stock of a certain costume in inventory.

companies: records the information of companies that we provide costumes for

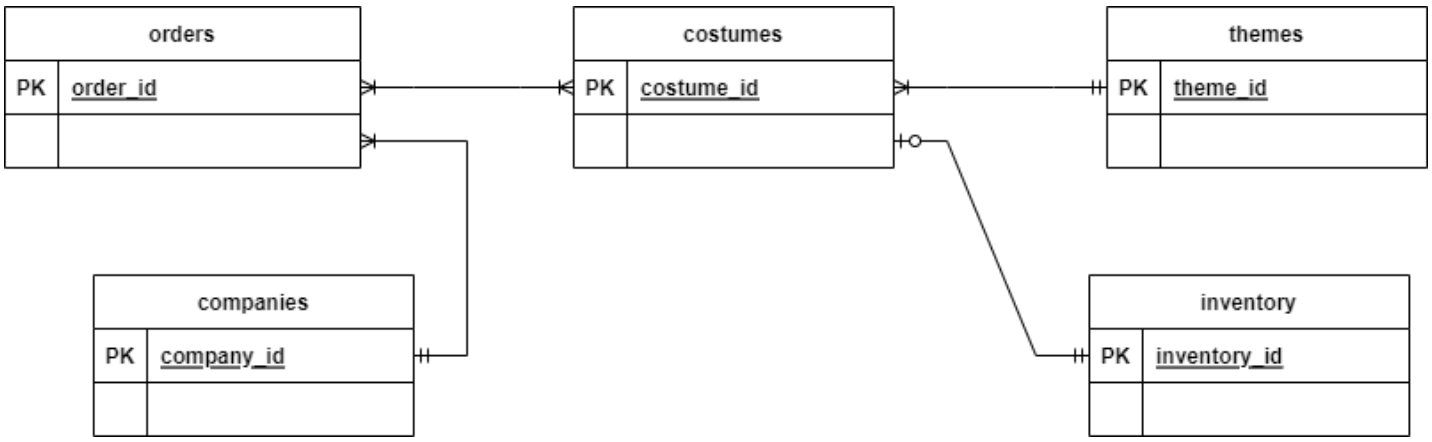
- company_id: int, auto_increment, not NULL, PK

- company_name: varchar(150), not NULL
- phone: varchar(20), not NULL
- email: varchar(50), not NULL
- Relationship:
    - M:1 between companies and orders with company_id as FK inside of orders. It's mandatory for a company to have 1 or more orders and it's mandatory for an order to be associated with one company.

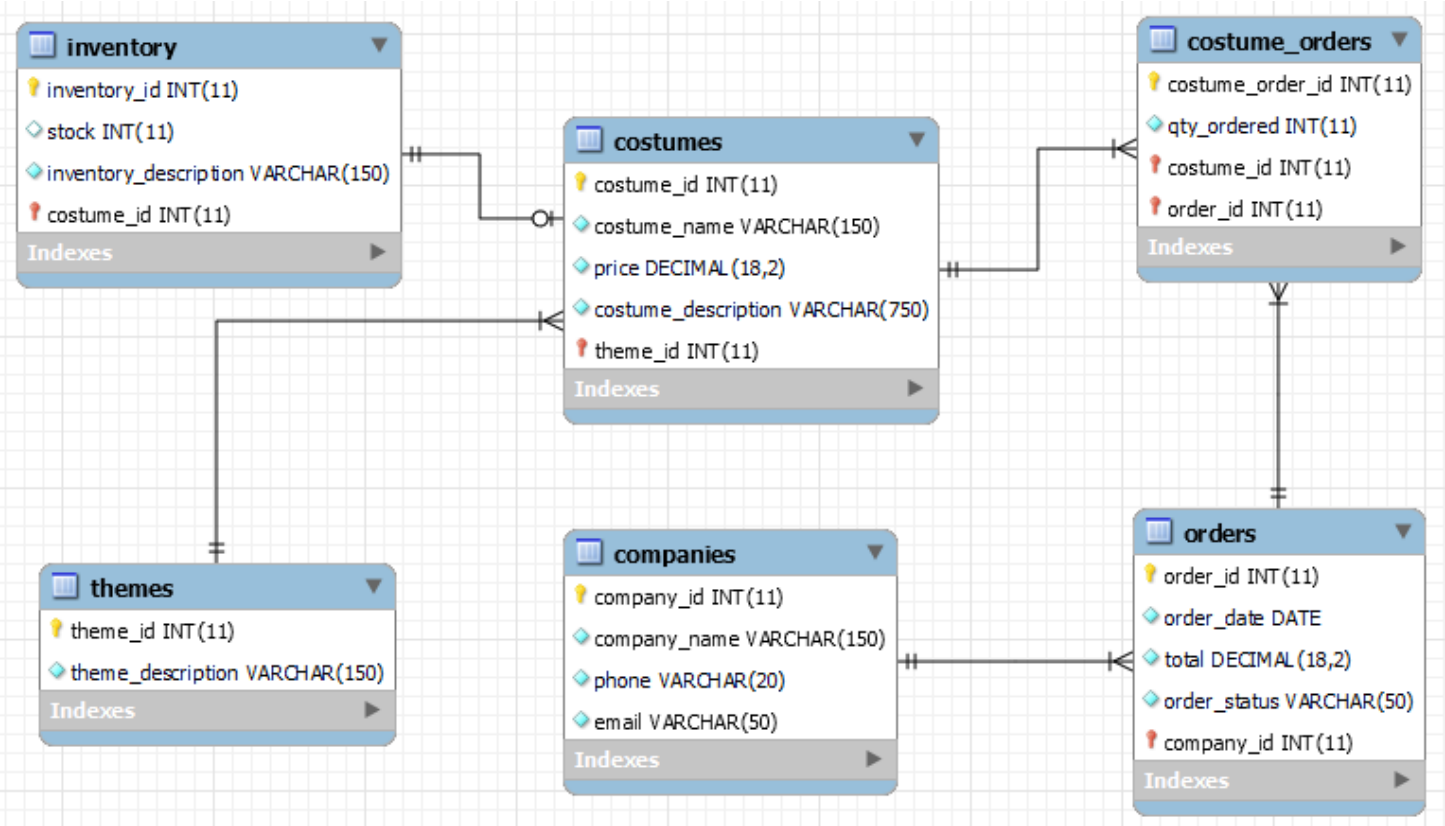
orders: records costume orders for companies
- order_id: int, auto_increment, not NULL, PK
- order_date: date, not NULL
- total: decimal(18,2), not NULL
- order_status: varchar(50), not NULL
- company_id: int, not NULL, FK
- Relationship:
    - M:N between costumes and orders with costume_orders acting as an intersection table.
    - 1:M between orders and companies with company_id as FK inside of orders. It's mandatory for a company to have 1 or more orders and it's mandatory for an order to be associated with one company.

costume_orders: intersection table between costumes and orders
- costume_order_id: int, not NULL, PK
- costume_id: int, not NULL, FK
- order_id: int, not NULL, FK
- qty_ordered : int, NOT NULL
- Relationship:
    - 1:M between costume_orders and costumes with costume_id as a FK inside of costume_orders.
    - 1:M between costume_orders and orders with order_id as a FK inside of costume_orders.

Entity-Relationship Diagram



Schema



Example Data

costumes:

costume_id	costume_name	price	costume_description	theme_id
1	Spooky Vampire	450.00	Thirsty for some blood? This adult vampire costume...	1
2	Dark Witch	500.00	This beautiful celestial witch costume will have you casting spells...	2
3	Dark Knight	600.00	Protect Gotham City from the forces of Evil and the deadly Joker...	3

- M:1 between costume and themes with theme_id as a FK inside of costumes. It's mandatory for a costume to have a costume theme and for a costume theme to be associated with a costume

themes:

theme_id	theme_description
1	Vampire
2	Witch
3	Superhero

- 1:M between costume and themes with theme_id as a FK inside of costumes. It's mandatory for a costume to have a costume theme and for a costume theme to be associated with a costume.

inventory:

inventory_id	stock	inventory_description	costume_id
1	500	aisle 1	1
2	1000	aisle 2	2
3	1200	aisle 3	3

- 1:1 between costumes and inventory with costume_id as a FK inside of inventory. It's possible for there to be 0 stock of a certain costume in inventory.

companies:

company_id	company_name	phone	email
1	Halloween 4 U	123-467-1224	halloweenforyou@gmail.com
2	Buy Costumes	123-789-4556	buycostumes@gmail.com
3	Halloween 2 Go	123-777-8889	halloween2gc@gmail.com

- M:1 between companies and orders with company_id as FK inside of orders. It's mandatory for a company to have 1 or more orders and it's mandatory for an order to be associated with one company.

orders:

order_id	order_date	total	order_status	company_id
1	2020-10-20	27500.00	delivered	1
2	2022-10-03	11250.00	in transit	2
3	2022-05-03	34200.00	in transit	1

- M:N between costumes and orders with costume_orders acting as an intersection table.
- 1:M between orders and companies with company_id as FK inside of orders

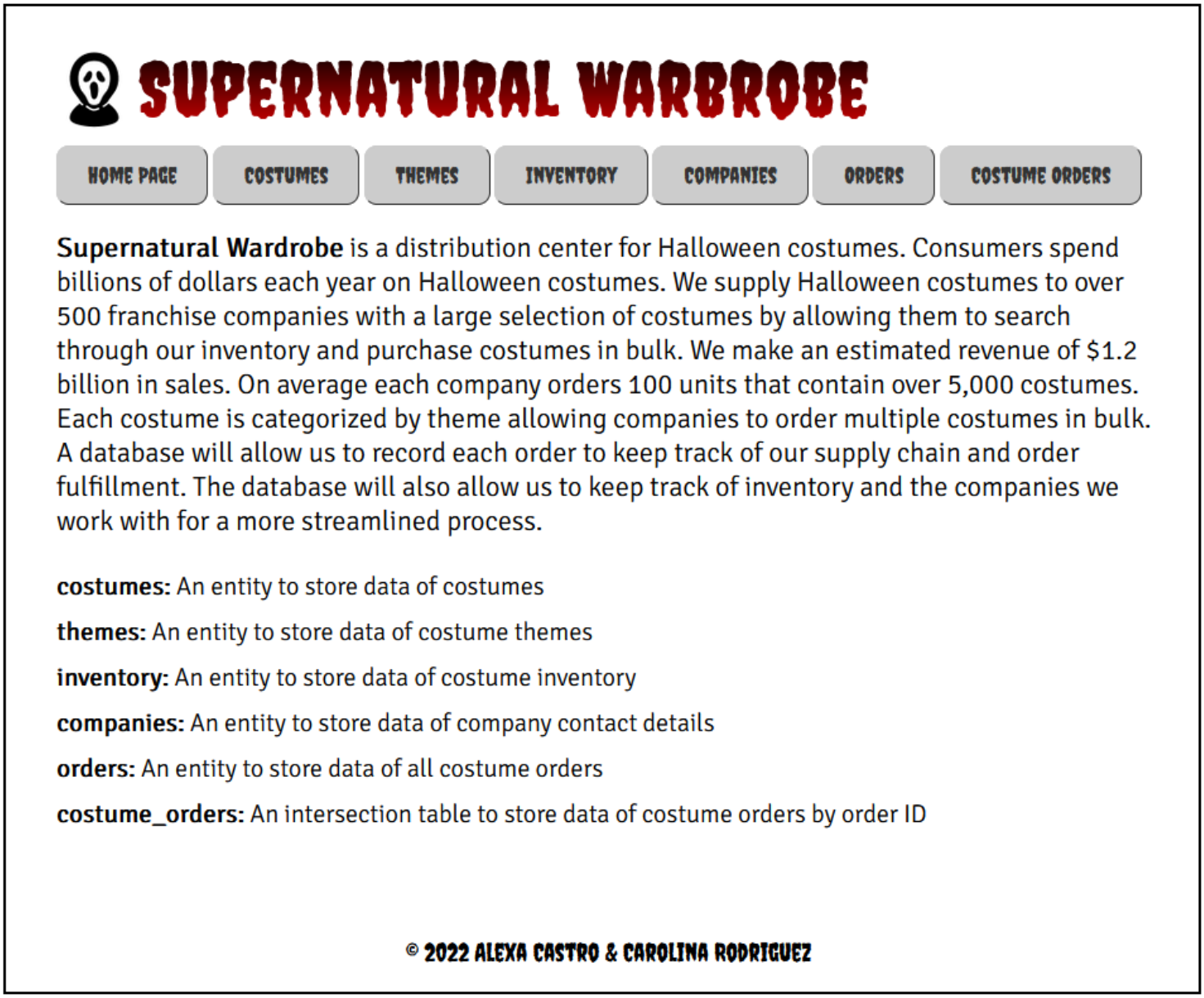
costume_orders (intersection table):

costume_order_id	order_id	costume_id	qty_ordered
1	2	1	25
2	3	1	36
3	1	2	55
4	3	3	30

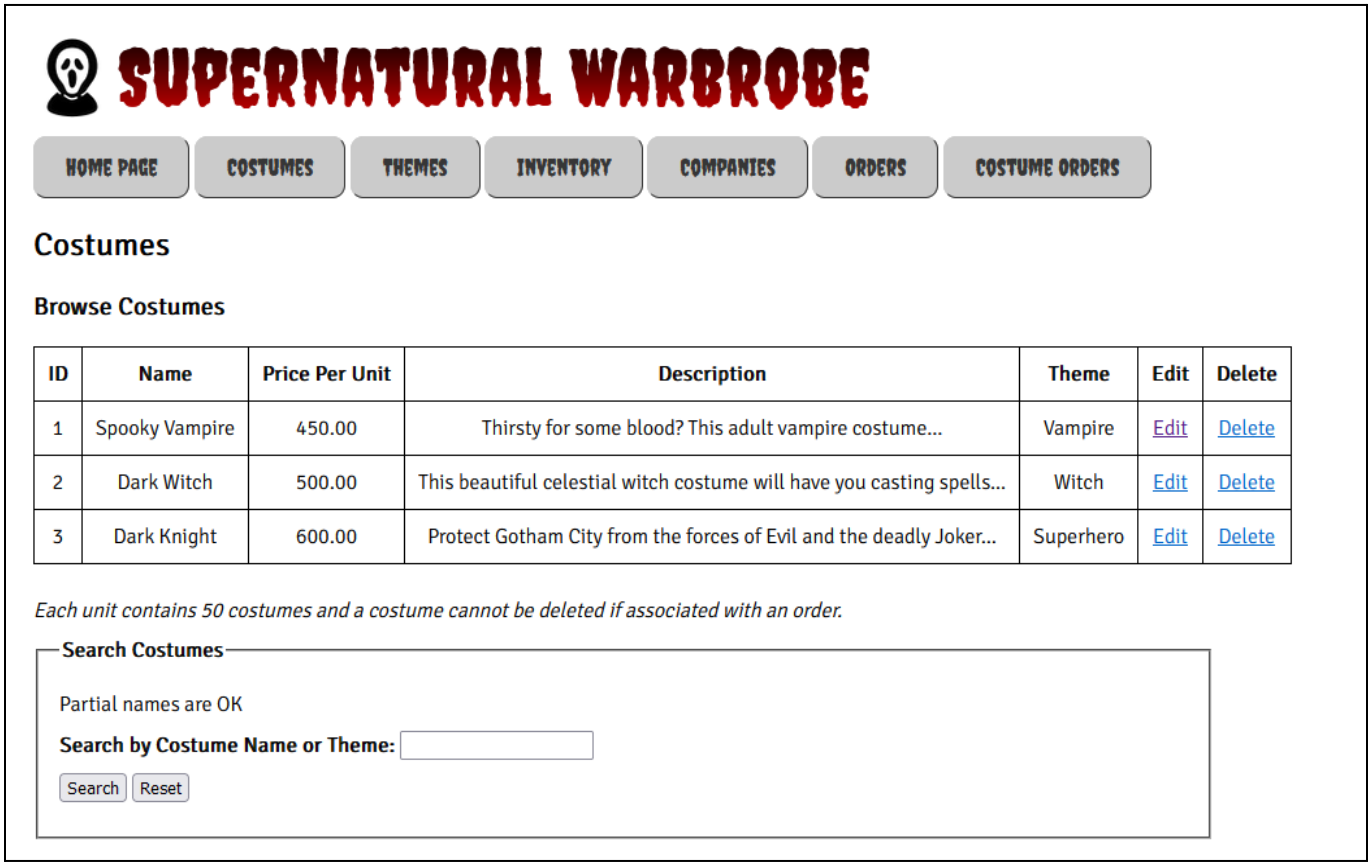
- 1:M between costume_orders and costumes with costume_id as a FK inside of costume_orders.
- 1:M between costume_orders and orders with order_id as a FK inside of costume_orders.

UI Screenshots:

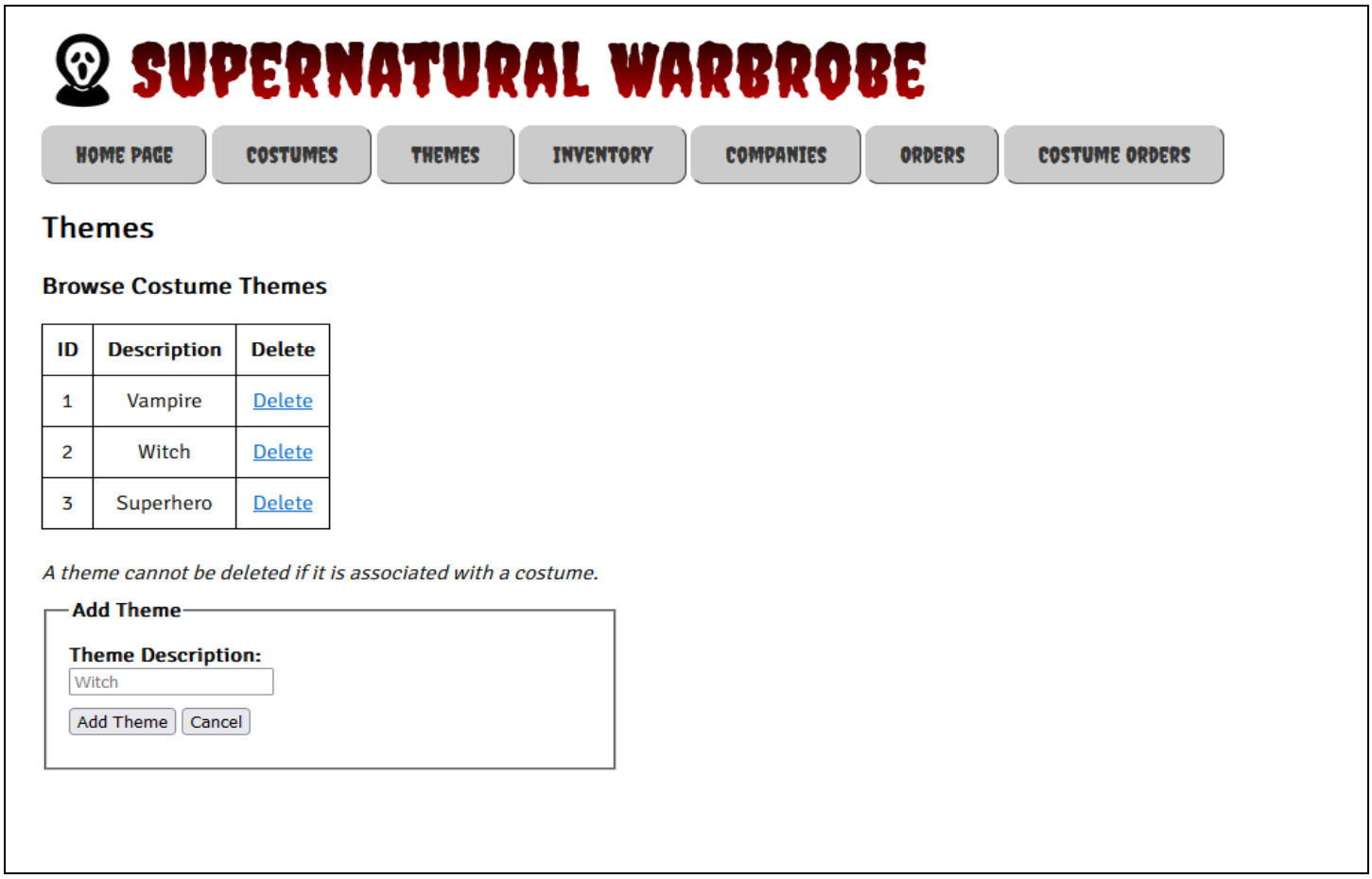
“Home Page”



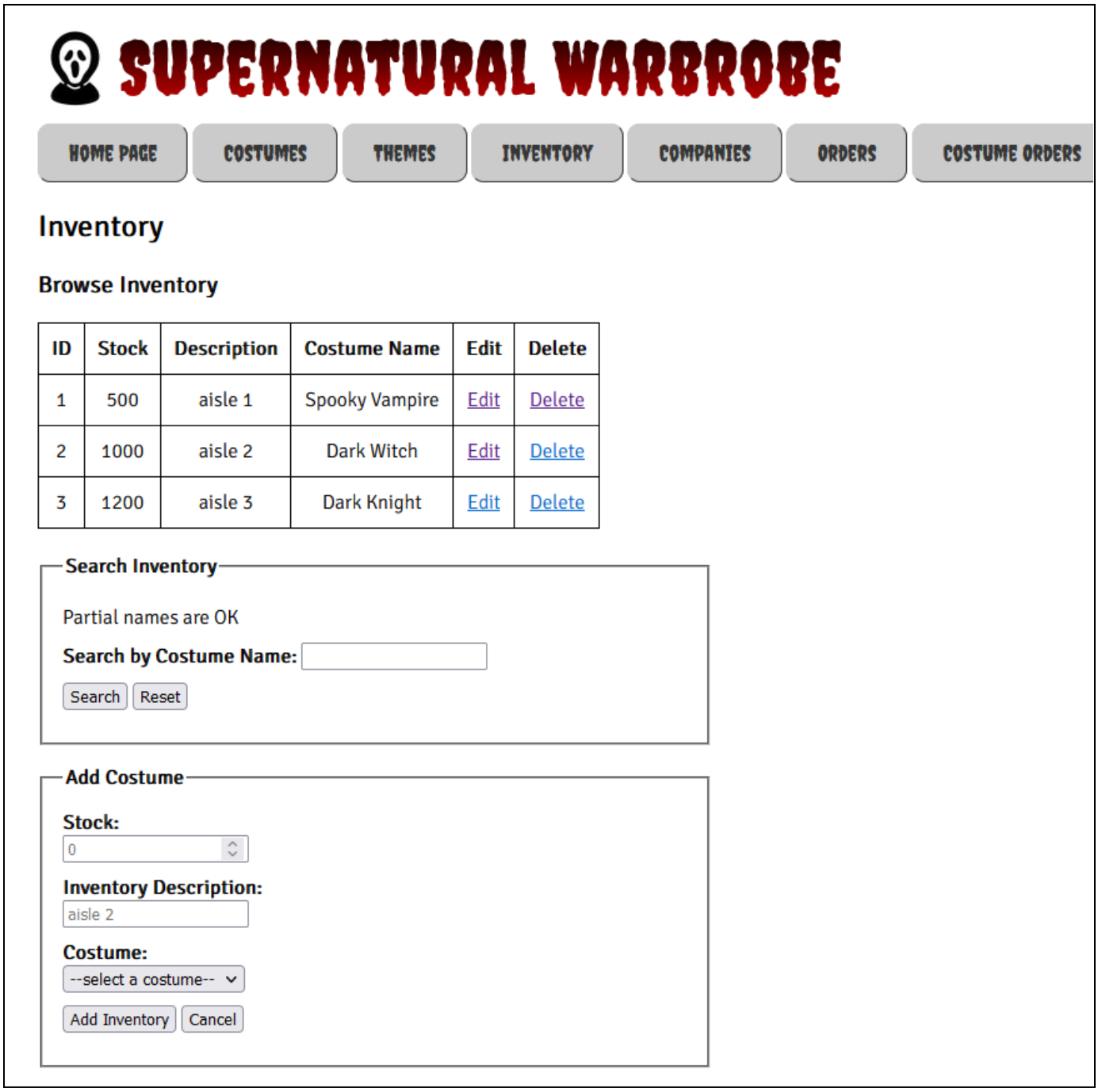
“CREATE/READ/UPDATE/DELETE & SEARCH Costumes Page”



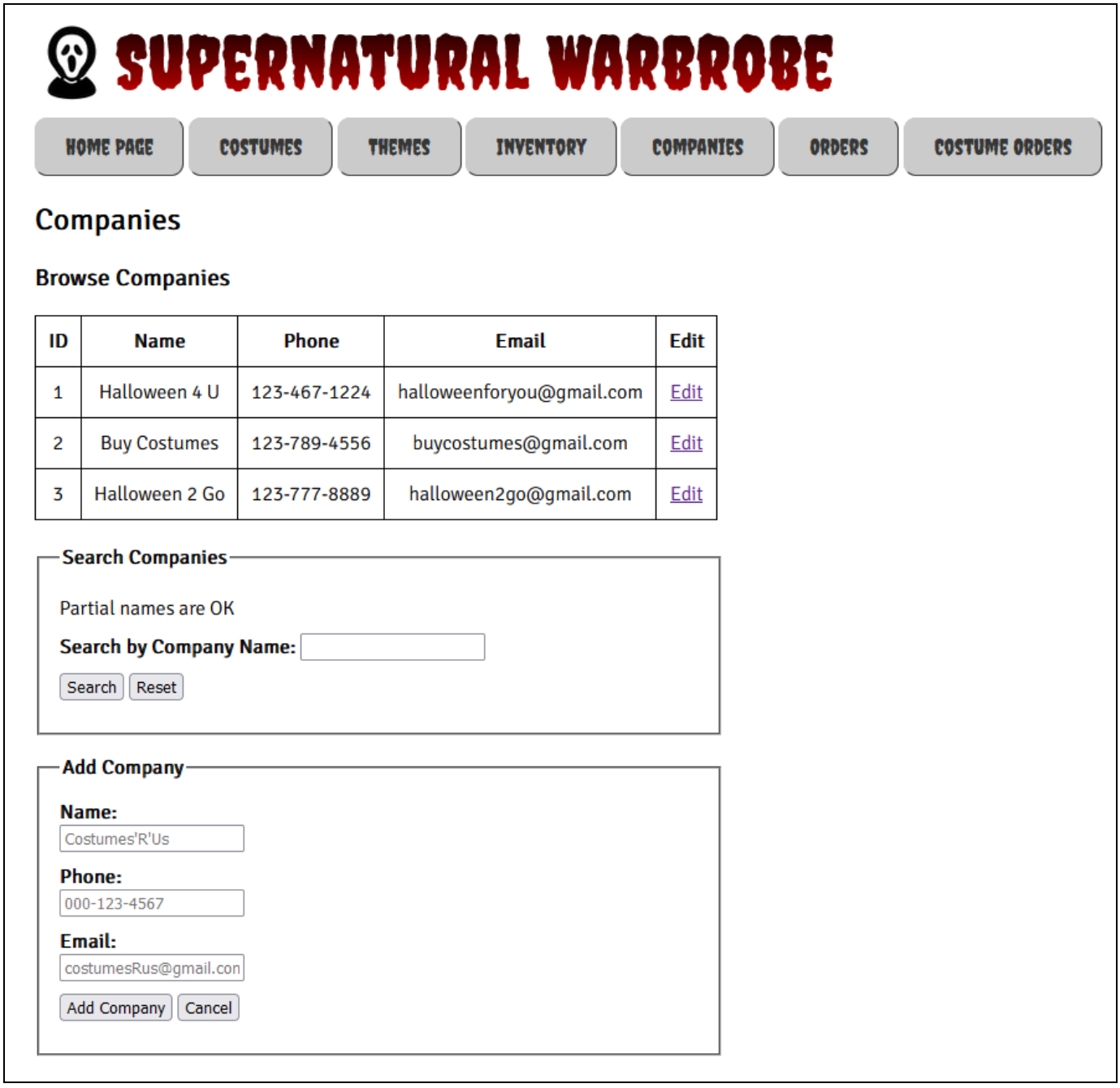
“CREATE/READ/DELETE Themes Page”



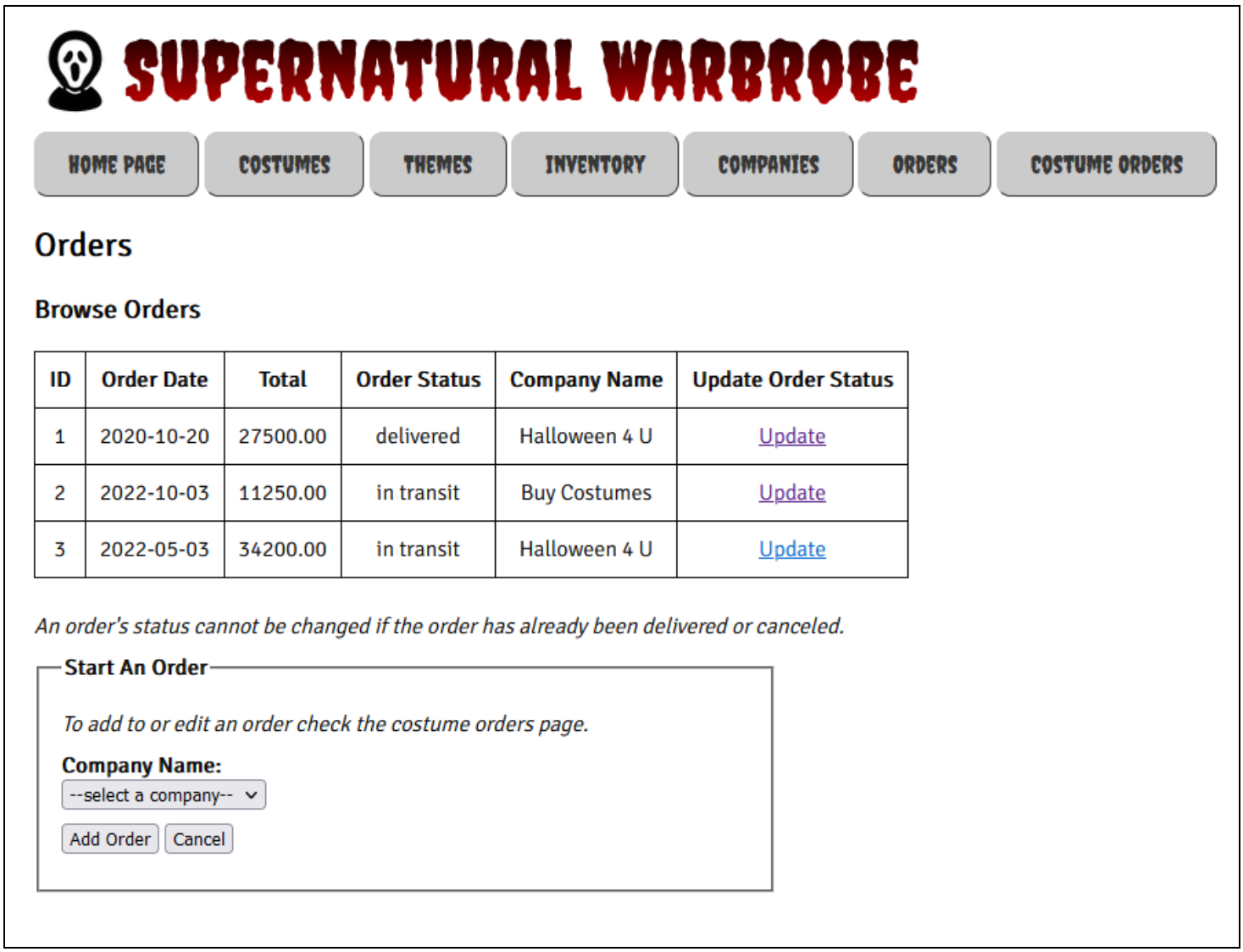
“CREATE/READ/UPDATE/DELETE & SEARCH Inventory Page”



“CREATE/READ/UPDATE & SEARCH Companies Page”



“CREATE/READ/UPDATE Orders Page”



“CREATE/READ/UPDATE/DELETE Costume Orders”