Harshita Singh 501042022
Dibbyo Saha 501069290
Eqan Ahmad 500910202

CPS 510 Assignment 1

Application Description: Finalize the application in consultation with the lab TA. Prepare a report to clearly describe about the application, its functions and information you expect from it, at the level of a technical report, and submit it to the TA (6 Marks)

Equipment Rental System

The proposed system for the assignment is an equipment rental system. The application will allow an end user to rent a tool from a company's physical store. The system will also let employees update information for products, view who has rented equipment and query, update and add relevant data to the system.

How will this system be used?

(a) Customer:
- A customer for our system will have the ability to search/query for products they are interested in renting.
- Customers will have the ability to also rent out a product.
- Interface for customer is primarily for searching for products and renting products

(b)Administrator:
- An administrator of the system who can create reports and view analytics based on the data
- Interface for admin is primarily for viewing reports, analyzing data, adding/removing products, updating price of products etc.

Core Functions:

Query:
- Search for a product by product name
- List all unrented equipment in a specific category
- List all equipment available in a specific store
- List all stores that offer the equipment rental service
- Return the price for a product
- Return whether a product is unavailable to be rented
- List which items are rented out and to which customer
- Admin users can search for most sold items

- Admin users can search for items unsold for a long period
- Admin users can categorize items on discounted sale

Our system is focused on an end user who is interested in renting out some equipment. The query functions are needed so that the end user can easily find all the relevant information they need. The customer has the option to do a broad search or find their product from categories. Furthermore, our query functions will let employees/admins determine which products are rented out and to which customer.

Update:

- Change a tool to be unavailable to be rented, if all are rented out
- Change the quantity available to be rented, when a tool is rented out
- Change the status of a tool to unavailable or available based on being rented or not.
- Change attributes of a user (email, phone)
- Update discounts and sales on products
- Update the prices of products when necessary
- Update changes in customer information

Update functions would primarily be for the customer to know the availability of an item. An administrator could utilize update functions as well to change the status of an item or report the quality of it.

Insert/Delete:

- Add or delete a new product
- Add a new record of a product being rented out
- Add or delete records of customers

Insert and delete functions would be utilized by administrators to add or remove products from the system. They will also be used when a customer rents out a tool, to create a historical record of the renters of a specific tool.

Units

unit_id	product_id	product_name	grade	availability	last_rened (mm/dd/yyyy)	price
H1B	H1	MasterCraft Stubby Hammer, 8-oz	B	Rented	09/12/2022	12

Products

product_id	product_name	product_location	product_category	owner_id	price_per_unit	quantity_total	quantity_rented
H1	MasterCraft Stubby Hammer, 8-oz	Brampton	Hammer	BMT005	12	5	2

Product Category

product_category_id	product_category_name
H	Hammer

Grade

grade_id	grade_name
A	Brand new

Units Leased

id	unit_id	product_id	renter_id	rent_from (mm/dd/yyyy)	rent_to (mm/dd/yyyy)	price_per_unit	total_price
1	H1B	H1	A123	09/12/2022	09/30/2022	12	50

Users

user_id	user_type_id	username	password	location_id	phone_num	email
A123	0	JohnSmith	pass123	1234	234-768-8180	john.smith@business.com

User Types

user_type_id	user_type_name
0	Commercial

User Location

user_id	location_id	country	city	postal_code
A123	1234	Canada	Toronto	M5B 2K3

High Level Database Constraints

- User_id should be unique for each USERS record (key constraint)
- Product_id should be unique for each PRODUCTS record (key constraint)
- Unit_id should be unique for each UNITS record (key constraint)
- Id should be unique for each UNITS_LEASED record (key constraint)
- Each USER record must relate to one USER_TYPES record (referential integrity constraint)
- Each USERS record is related to one USER LOCATION record (referential integrity constraint)

The Units table is a weak entity because it depends on the Products table. The relationship between them is a one-to-many relationship.

The Products table is an entity because it stores information about the products that are available for rent. The relationships between the Products table and the other tables are one-to-one relationships.

The Product Category table is an entity because it stores information about the different categories of products that are available. The relationship between the Product Category table and the Products table is a one-to-many relationship.

The Grade table is an entity because it stores information about the different grades of products that are available. The relationship between the Grade table and the Units table is a one-to-many relationship.

The Units Leased table is an entity because it stores information about the units that have been leased out. The relationships between the Units Leased table and the other tables are one-to-one relationships.

The Users table is an entity because it stores information about the users of the system. The relationship between the Users table and the User Location table is a one-to-one relationship.

The User Types table is an entity because it stores information about the different types of users that are able to use the system. The relationship between the User Types table and the Users table is a one-to-many relationship.

The User Location table is a weak entity because it depends on the Users table. The relationship between them is a one-to-many relationship.

The relationships in this system are as follows:

- One unit can be rented by many users (1:M)
- One user can rent many units (M:1)
- One product can have many units (1:M)
- One unit can belong to one product (1:1)
- One product category can have many products (1:M)
- One product can belong to one product category (1:1)
- One user can have one user type (1:1)
- One user type can have many users (1:M)
- One user can have one location (1:1)
- One location can have many users (1:M)

The participation constraints in this system are as follows:

- A unit must belong to a product (mandatory)
- A product can have many units (optional)
- A product category can have many products (optional)
- A product must belong to a product category (mandatory)
- A user can have many units leased (optional)
- A unit can be leased by many users (optional)
- A user can have one user type (mandatory)
- A user type can have many users (optional)
- A user can have one location (mandatory)
- A location can have many users (optional)

The primary keys in this system are as follows:

- The primary key for the Unit entity is unit_id.
- The primary key for the Product entity is product_id.
- The primary key for the Product Category entity is product_category_id.
- The primary key for the Grade entity is grade_id.
- The primary key for the Units Leased entity is id.
- The primary key for the User entity is user_id.
- The primary key for the User Type entity is user_type_id.
- The primary key for the User Location entity is user_id.

The foreign keys in this system are as follows:

- The foreign key in the Unit entity is product_id. This references the Product entity.
- The foreign key in the Product entity is owner_id. This references the User entity.
- The foreign key in the Product entity is product_category. This references the Product Category entity.
- The foreign key in the Units Leased entity is unit_id. This references the Unit entity.
- The foreign key in the Units Leased entity is renter_id. This references the User entity.