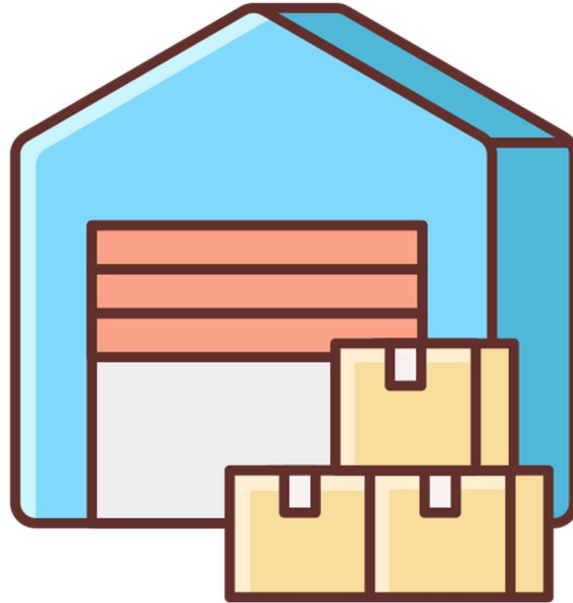




UNIVERSITÀ
DI PARMA

Warehouse



Progetto a cura di:

Tanzi Manuel – 307720

Professore:

Cagnoni Stefano

Sommario

Specifiche del progetto	2
Schemi Concettuali	3
Query.....	4

Specifiche del progetto

Il progetto deve possedere una schermata di login alla quale è possibile accedere con le credenziali fornite dall'amministratore del sistema poiché sarà proprio quest'ultime ad avere la capacità di amministrare gli utenti del sistema.

Tutti gli utenti possiedono un livello di operatività che gli permette di navigare all'interno del sito solamente nelle sezioni a lui dedicate. Nello specifico esistono 3 tipologie di livelli:

1. Admin
2. Special
3. User

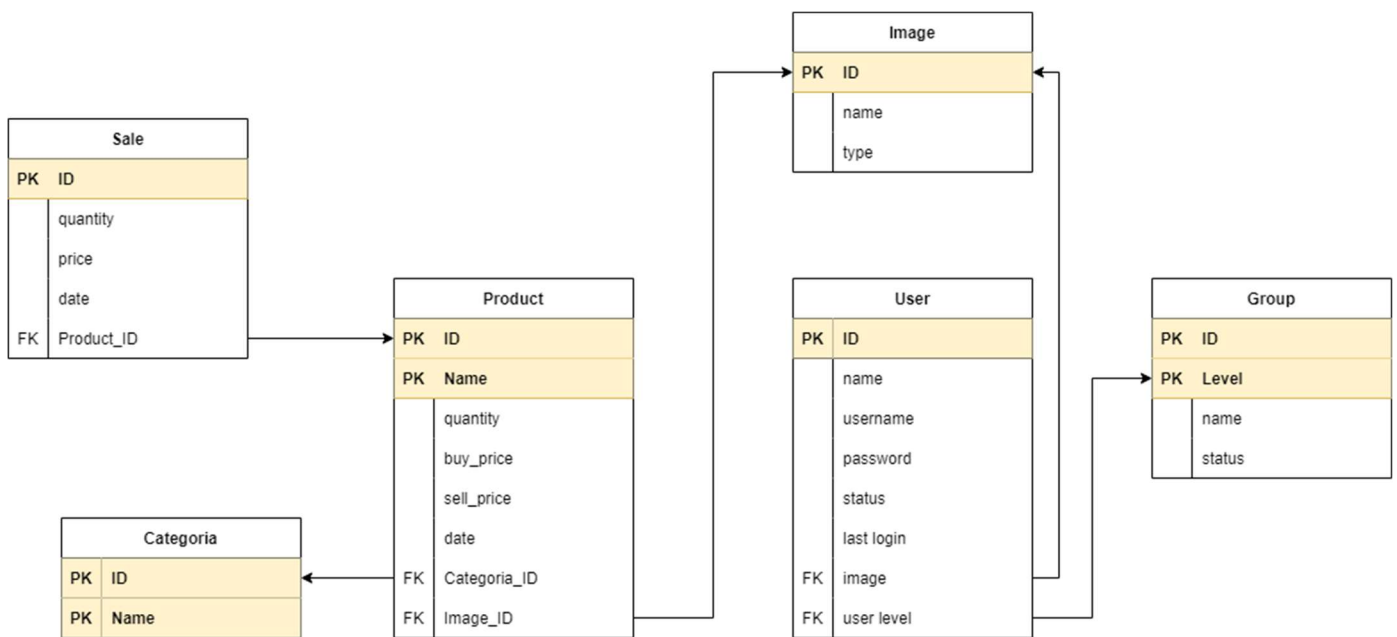
Ogni utente ha inoltre un parametro status che identifica la sua operabilità che può essere sempre gestita da un account con privilegio *"admin"*.

Una volta eseguito il login si presentano differenti possibilità di navigazione dipendenti dal tipo di account loggato.

Tipologie di schemate presenti nel sistema:

- **Dashboard** -> Presenta alcune card specifiche del sistema, questa pagina dinamica avrà una diversa visualizzazione per il tipo di account loggato, cosicché si è in grado di nascondere alcuni dati sensibili.
- **User Managment** -> In questa pagina si è in grado di amministrare tutto quello che riguarda gli utenti, i gruppi e i loro stati di attività.
- **Categories** -> E' possibile visionare tutte le categorie presenti nel sistema ed averne la possibilità di modifica, cancellazione e di aggiunta.
- **Products** -> Visualizzazione dei prodotti all'interno del magazzino con le relative informazioni. Sono inoltre possibili eseguire delle azioni come la cancellazione e la modifica dello stesso qualora si volesse correggere un errore o sistemare un prodotto.
- **Medias** -> Visualizzazione di tutti i media presenti e registrati all'interno del sito usati come immagine profilo o come immagine per rappresentare un determinato oggetto dello stock del magazzino. E' sempre possibile eseguire delle modifiche.
- **Sales** -> Pagina dedicata alle vendite. In questa pagina si possono visualizzare le vendite avvenute con le relative informazioni. Aggiungere una vendita passata o futura e avere la possibilità di eseguire modifiche su di essi qualora si avesse eseguito l'accesso con un account privilegiato.
- **Profile** -> Pagina dedicata alle impostazioni del profile potendo visualizzare qualche specifica tecnica e averne la possibilità di modificare le informazioni come *"username"*, *"password"*, *"immagine profilo"*.
- **Logout** -> Riporterà l'utente alla pagina di login.

Schemi Concettuali



GROUP(ID, LEVEL, name, status)

IMAGE (ID, name, type)

USER (ID, name, username, password, status, last_login, image, user_level)

CATEGORIA (ID, NAME)

PRODUCT (ID, NAME, quantity, buy_price, sell_price, date, Categoria_ID, Image_ID)

SALE (ID, quantity, price, date, Product_ID)

Query

Authenticate: SELECT id,username,password,user_level FROM users WHERE username = '%s' LIMIT 1", \$username

Find all users: SELECT u.id,u.name,u.username,u.user_level,u.status,u.last_login
g.group_name FROM users u LEFT JOIN user_groups g ON g.group_level=u.user_level
ORDER BY u.name ASC

Update last login: UPDATE users SET last_login='{ \$date}' WHERE id = '{ \$user_id}'

Find by group name: SELECT group_name FROM user_groups WHERE group_name = '{ \$db->escape(\$val)}'

Find by group level: SELECT group_level FROM user_groups WHERE group_level = '{ \$db->escape(\$level)}'

Find all: SELECT * FROM { \$db->escape(\$table)}

Find by ID: SELECT * FROM { \$db->escape(\$table)} WHERE id='{ \$db->escape(\$id)}'

Delete by ID: DELETE FROM { \$db->escape(\$table)} WHERE id={ \$db->escape(\$id)}

Count by ID: SELECT COUNT(id) AS total FROM { \$db->escape(\$table)}

Count by TypeUser: SELECT COUNT(id) AS total FROM { \$db->escape(\$table)} WHERE
user_level={ \$db->escape(\$type)}

Count price current month: SELECT COUNT(id) AS total FROM { \$db->escape(\$table)}
WHERE MONTH({ \$db->escape('date')}) = MONTH(CURRENT_DATE()) AND YEAR({ \$db->escape('date')}) = YEAR(CURRENT_DATE())

Count account active: SELECT COUNT(id) AS total FROM { \$db->escape("user_groups")}
WHERE group_status = 1

Sum price current Month: SELECT SUM(\$type) AS total FROM { \$db->escape(\$table)}
WHERE MONTH({ \$db->escape('date')}) = MONTH(CURRENT_DATE()) AND YEAR({ \$db->escape('date')}) = YEAR(CURRENT_DATE())

Sum price current Year: SELECT SUM(\$type) AS total FROM { \$db->escape(\$table)} WHERE
YEAR({ \$db->escape('date')}) = YEAR(CURRENT_DATE())

Product Info: SELECT p.id,p.name, p.quantity, p.buy_price, p.sale_price, p.media_id, p.date,
c.name categorie, m.file_name AS image FROM products p LEFT JOIN categories c ON c.id =
p.categorie_id LEFT JOIN media m ON m.id = p.media_id ORDER BY p.id ASC

Find Product Title: SELECT name FROM products WHERE name like '% \$p_name%'

Find all product by title: SELECT * FROM products WHERE name = '{ \$title}'

Update product quantity: UPDATE products SET quantity=quantity - '{ \$qty}' WHERE id =
'{ \$id}'

Recent product added: SELECT p.id,p.name,p.sale_price,p.media_id,c.name AS categorie, m.file_name AS image FROM products p LEFT JOIN categories c ON c.id = p.categorie_id LEFT JOIN media m ON m.id = p.media_id ORDER BY p.id DESC

Find Highest selling product: SELECT p.name, COUNT(s.product_id) AS totalSold, SUM(s.qty) AS totalQty FROM sales s LEFT JOIN products p ON p.id = s.product_id GROUP BY s.product_id ORDER BY SUM(s.qty) DESC

Find all sale: SELECT s.id,s.qty,s.price,s.date,p.name FROM sales s LEFT JOIN products p ON s.product_id = p.id BY s.date DESC

Find recent sale added: SELECT s.id,s.qty,s.price,s.date,p.name FROM sales s LEFT JOIN products p ON s.product_id = p.id ORDER BY s.date DESC

Find sale by date: SELECT s.date, p.name,p.sale_price,p.buy_price COUNT(s.product_id) AS total_records, (s.qty) AS total_sales, SUM(p.sale_price * s.qty) AS total_saleing_price, SUM(p.buy_price * s.qty) AS total_buying_price sales s JOIN products p ON s.product_id = p.id WHERE s.date BETWEEN '{\$start_date}' AND '{\$end_date}' GROUP BY DATE(s.date),p.name ORDER BY DATE(s.date) DESC

Daily sales: SELECT s.qty , DATE_FORMAT(s.date, '%Y-%m-%e') AS date,p.name, SUM(p.sale_price * s.qty) AS total_saleing_price FROM sales s LEFT JOIN products p ON s.product_id = p.id WHERE DATE_FORMAT(s.date, '%Y-%m') = '{\$year}-{\$month}' GROUP BY DATE_FORMAT(s.date, '%e'),s.product_id

Monthly sales: SELECT s.qty, DATE_FORMAT(s.date, '%Y-%m-%e') AS date,p.name, SUM(p.sale_price * s.qty) AS total_saleing_price FROM sales s LEFT JOIN products p ON s.product_id = p.id WHERE DATE_FORMAT(s.date, '%Y') = '{\$year}' GROUP BY DATE_FORMAT(s.date, '%c'),s.product_id ORDER BY date_format(s.date, '%c') ASC

Update user image: UPDATE users SET image='{\$db->escape(\$this->fileName)}'WHERE id='{\$db->escape(\$id)}'

Insert Media: INSERT INTO media (file_name,file_type) VALUES '{\$db->escape(\$this->fileName)}', '{\$db->escape(\$this->fileType)}')

Add categoria: INSERT INTO categories (name) VALUES ('{\$cat_name}')

Insert Group: INSERT INTO user_groups (group_name,group_level,group_status) VALUES ('{\$name}', '{\$level}','{\$status}')

Insert User: INSERT INTO users (name,username,password,user_level,status) VALUES ('{\$name}', '{\$username}', '{\$password}', '{\$user_level}','1')

Update group: UPDATE user_groups SET name='{\$name}', group_level='{\$level}', group_status='{\$status}' WHERE ID='{\$id}'

Update User: UPDATE users SET name = '{\$name}', username = '{\$username}', user_level='{\$level}', status='{\$status}' WHERE id='{\$id}'

Update password: UPDATE users SET password='{\$h_pass}' WHERE id='{\$db->escape(\$id)}'

Update Product: UPDATE products SET name = '{\$p_name}', quantity = '{\$p_qty}', buy_price = '{\$p_buy}', sale_price = '{\$p_sale}', categorie_id = '{\$p_cat}', media_id = '{\$media_id}' WHERE id = '{\$p_id}'

Add Product: INSERT INTO products (name,quantity,buy_price, sale_price,categorie_id, media_id,date) VALUES (" '{\$p_name}', '{\$p_qty}', '{\$p_buy}', '{\$p_sale}', '{\$p_cat}', '{\$media_id}', '{\$date}')) ON DUPLICATE KEY UPDATE name='{\$p_name}'

Update Sale: UPDATE sales SET product_id= '{\$p_id}',qty={\$s_qty}, price='{\$s_total}', date='{\$s_date}' WHERE id = '{\$s_id}'

Add Sale: INSERT INTO sales (product_id,qty,price,date) VALUES ('{\$p_id}','{\$s_qty}','{\$s_total}','{\$s_date}')