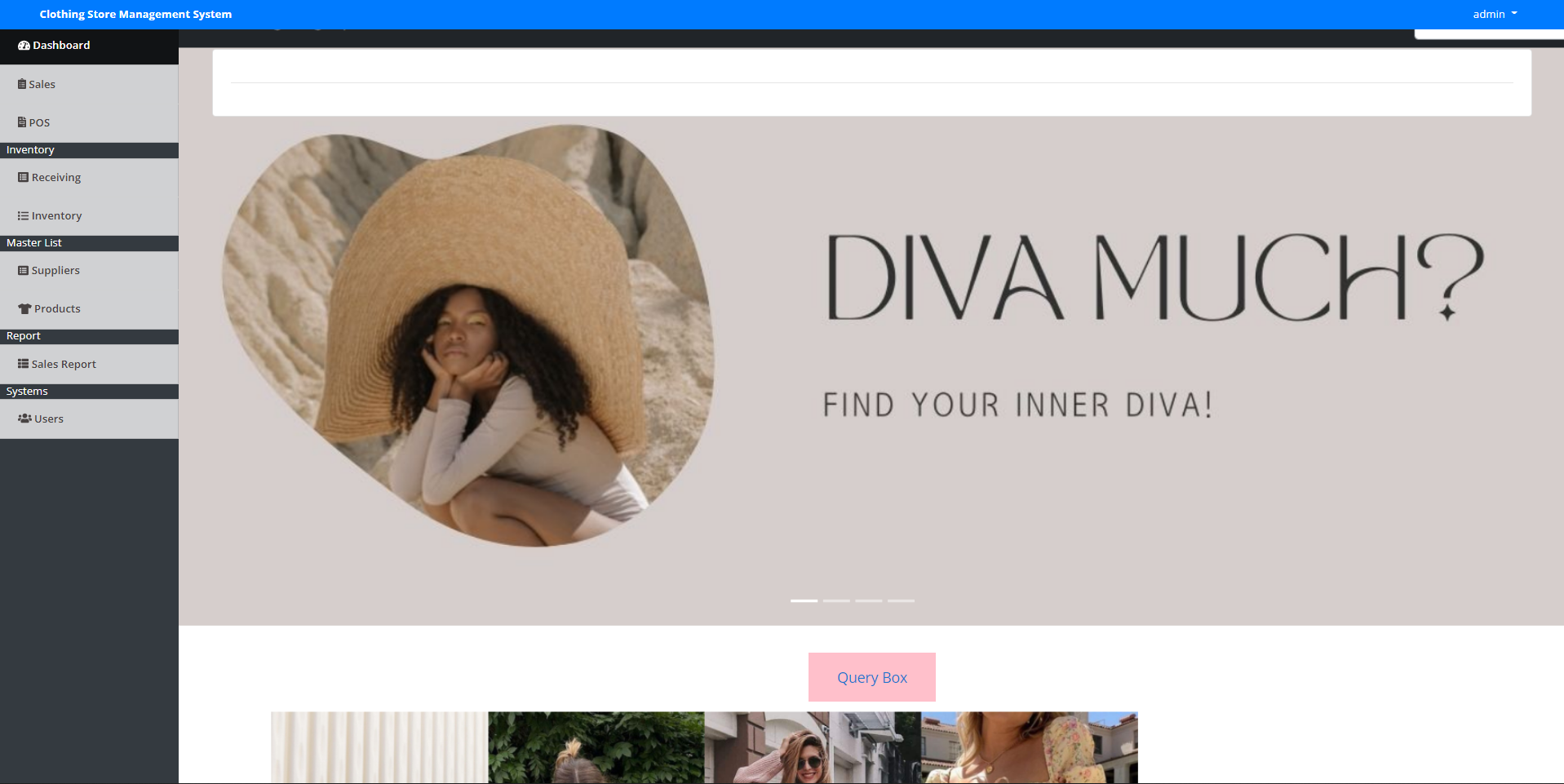
**DBMS - Mini Project**

**ECOMMERCE FOR CLOTHING MANAGEMENT SYSTEM**



**Submitted By:**

Name: Ananya Prabhakar

SRN: PES2UG20CS046

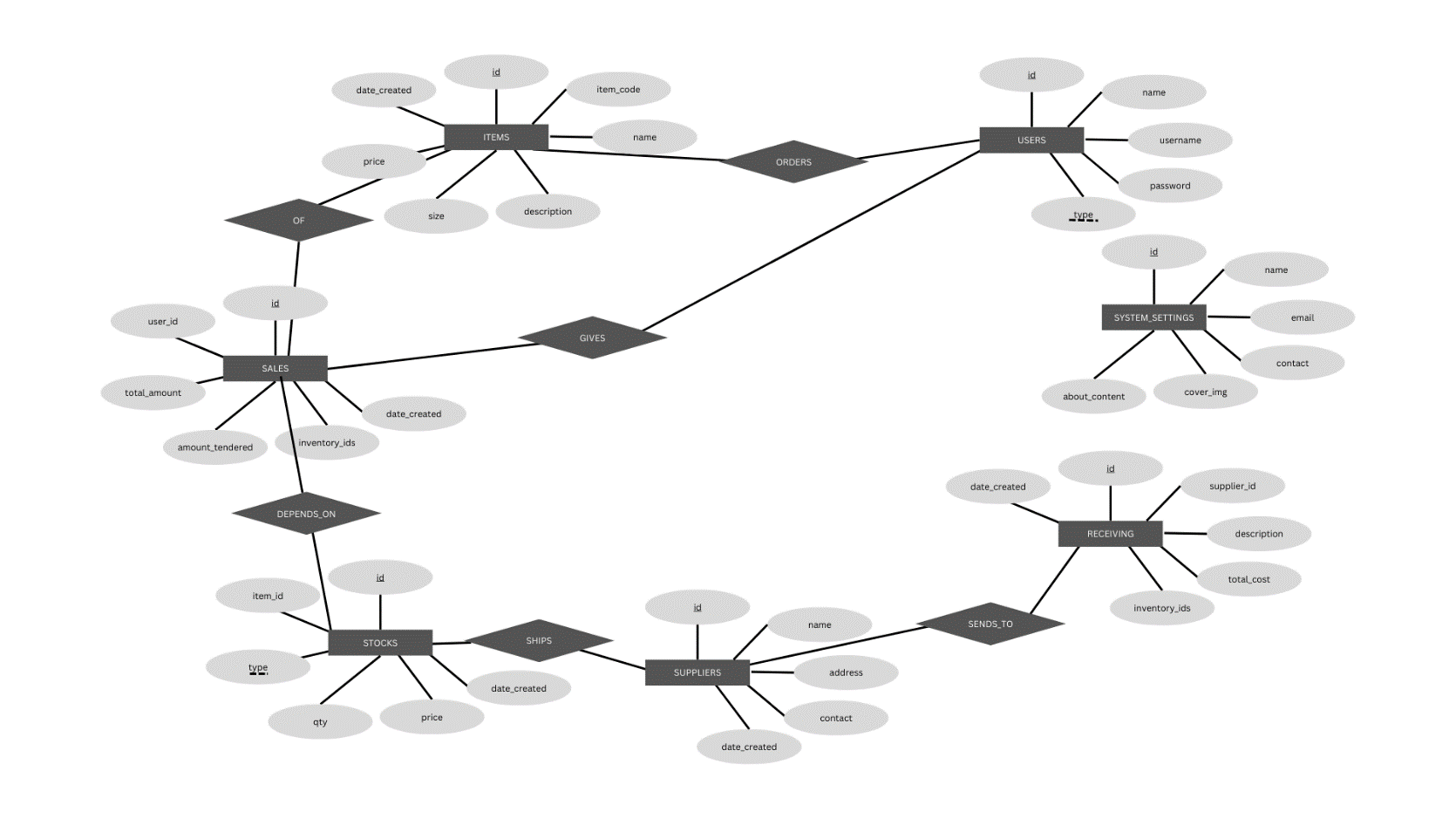
V Semester

Section: A

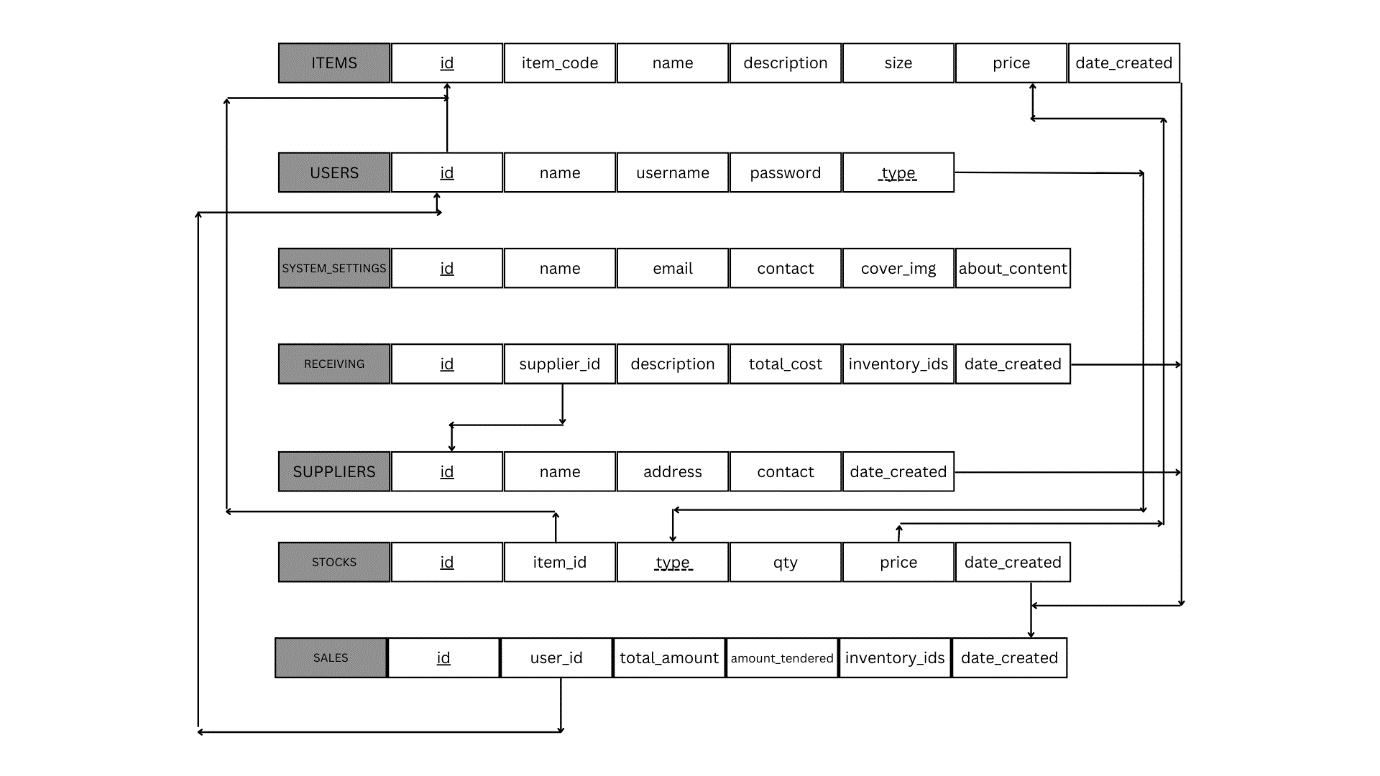
**Short Description and Scope of the Project**

The Clothing Store Management System is a simple PHP/MySQLi project that can help a clothing business or apparel businesses to manage their stocks and sales management. The system stores the list of products, suppliers, and the receiving transaction between the store and supplier. The clothing business can monitor their available stocks in each item or products that they are selling using the clothing store management system and also this system has a feature to manage their transaction with their customer which is the point of sale (POS) feature that stores the transaction to the database to allow the system calculate or update the available stock of each product of the clothing store. The POS feature of this system also generates a simple unofficial receipt of the transaction that has enough details of the transaction for the customer proof for any instances that may occur. The printable unofficial receipt has a responsive width feature so the receipt paper size won't be a problem when using this system.

**ER DIAGRAM**



**Relational Schema**

****

**DDL statements - Building the database and Populating database**

CREATE TABLE `items` (

`id` int(30) NOT NULL,

`item\_code` varchar(100) NOT NULL,

`name` varchar(200) NOT NULL,

`description` text NOT NULL,

`size` varchar(10) NOT NULL,

`price` float NOT NULL,

`date\_created` datetime NOT NULL DEFAULT current\_timestamp()

) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;

--

-- Dumping data for table `items`

--

INSERT INTO `items` (`id`, `item\_code`, `name`, `description`, `size`, `price`, `date\_created`) VALUES

(1, '309874256222', 'Sample Shirt', 'Sample only', 'M', 500, '2020-11-04 09:51:01'),

(2, '774420957142', 'sample 1', 'sample 1', 'XS', 250, '2020-11-04 10:58:59');

-- --------------------------------------------------------

--

-- Table structure for table `receiving`

--

CREATE TABLE `receiving` (

`id` int(30) NOT NULL,

`supplier\_id` int(30) NOT NULL,

`description` text NOT NULL,

`total\_cost` float NOT NULL,

`inventory\_ids` text NOT NULL,

`date\_created` datetime NOT NULL DEFAULT current\_timestamp()

) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;

--

-- Dumping data for table `receiving`

--

INSERT INTO `receiving` (`id`, `supplier\_id`, `description`, `total\_cost`, `inventory\_ids`, `date\_created`) VALUES

(1, 1, '', 215000, '7,9', '2020-11-04 14:56:18');

-- --------------------------------------------------------

--

-- Table structure for table `sales`

--

CREATE TABLE `sales` (

`id` int(30) NOT NULL,

`user\_id` int(30) NOT NULL,

`total\_amount` float NOT NULL,

`amount\_tendered` int(30) NOT NULL,

`inventory\_ids` text NOT NULL,

`date\_created` datetime NOT NULL DEFAULT current\_timestamp()

) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;

--

-- Dumping data for table `sales`

--

INSERT INTO `sales` (`id`, `user\_id`, `total\_amount`, `amount\_tendered`, `inventory\_ids`, `date\_created`) VALUES

(1, 1, 1500, 1500, '10,11', '2020-11-05 08:36:52'),

(2, 1, 750, 1000, '12', '2020-11-05 08:37:13');

-- --------------------------------------------------------

--

-- Table structure for table `stocks`

--

CREATE TABLE `stocks` (

`id` int(30) NOT NULL,

`item\_id` int(30) NOT NULL,

`type` tinyint(1) NOT NULL COMMENT '1= in,2=out',

`qty` int(30) NOT NULL,

`price` float NOT NULL,

`date\_created` datetime NOT NULL DEFAULT current\_timestamp()

) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;

--

-- Dumping data for table `stocks`

--

INSERT INTO `stocks` (`id`, `item\_id`, `type`, `qty`, `price`, `date\_created`) VALUES

(7, 2, 1, 500, 250, '2020-11-04 15:55:03'),

(9, 1, 1, 500, 180, '2020-11-04 15:55:29'),

(10, 2, 2, 2, 250, '2020-11-05 08:36:52'),

(11, 1, 2, 2, 500, '2020-11-05 08:36:52'),

(12, 2, 2, 3, 250, '2020-11-05 08:37:13');

-- --------------------------------------------------------

--

-- Table structure for table `suppliers`

--

CREATE TABLE `suppliers` (

`id` int(30) NOT NULL,

`name` varchar(200) NOT NULL,

`address` text NOT NULL,

`contact` varchar(50) NOT NULL,

`date\_created` datetime NOT NULL DEFAULT current\_timestamp()

) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;

--

-- Dumping data for table `suppliers`

--

INSERT INTO `suppliers` (`id`, `name`, `address`, `contact`, `date\_created`) VALUES

(1, 'ABC Apparel', 'CBD St., EFG City', '+6948 8542 623', '2020-11-04 09:33:26'),

(2, 'Men Apparel', 'Sample Address', '65524556', '2020-11-04 09:33:48'),

(3, 'Ladies Apparel', 'Company address', '65524556', '2020-11-04 09:34:15'),

(4, 'Trends Apparel', 'Sample Address', '8747808787', '2020-11-04 09:34:37');

-- --------------------------------------------------------

--

-- Table structure for table `system\_settings`

--

CREATE TABLE `system\_settings` (

`id` int(30) NOT NULL,

`name` text NOT NULL,

`email` varchar(200) NOT NULL,

`contact` varchar(20) NOT NULL,

`cover\_img` text NOT NULL,

`about\_content` text NOT NULL

) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;

--

-- Dumping data for table `system\_settings`

--

INSERT INTO `system\_settings` (`id`, `name`, `email`, `contact`, `cover\_img`, `about\_content`) VALUES

(1, 'Clothing Store Management System', '', '', '', '');

-- --------------------------------------------------------

--

-- Table structure for table `users`

--

CREATE TABLE `users` (

`id` int(30) NOT NULL,

`name` text NOT NULL,

`username` varchar(200) NOT NULL,

`password` text NOT NULL,

`type` tinyint(1) NOT NULL COMMENT '1=Admin,2=Staff'

) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;

--

-- Dumping data for table `users`

--

INSERT INTO `users` (`id`, `name`, `username`, `password`, `type`) VALUES

(1, 'admin', 'admin', '0192023a7bbd73250516f069df18b500', 1);

--

-- Indexes for dumped tables

--

--

-- Indexes for table `items`

--

ALTER TABLE `items`

ADD PRIMARY KEY (`id`);

--

-- Indexes for table `receiving`

--

ALTER TABLE `receiving`

ADD PRIMARY KEY (`id`);

--

-- Indexes for table `sales`

--

ALTER TABLE `sales`

ADD PRIMARY KEY (`id`);

--

-- Indexes for table `stocks`

--

ALTER TABLE `stocks`

ADD PRIMARY KEY (`id`);

--

-- Indexes for table `suppliers`

--

ALTER TABLE `suppliers`

ADD PRIMARY KEY (`id`);

--

-- Indexes for table `system\_settings`

--

ALTER TABLE `system\_settings`

ADD PRIMARY KEY (`id`);

--

-- Indexes for table `users`

--

ALTER TABLE `users`

ADD PRIMARY KEY (`id`);

--

-- AUTO\_INCREMENT for dumped tables

--

--

-- AUTO\_INCREMENT for table `items`

--

ALTER TABLE `items`

MODIFY `id` int(30) NOT NULL AUTO\_INCREMENT, AUTO\_INCREMENT=3;

--

-- AUTO\_INCREMENT for table `receiving`

--

ALTER TABLE `receiving`

MODIFY `id` int(30) NOT NULL AUTO\_INCREMENT, AUTO\_INCREMENT=2;

--

-- AUTO\_INCREMENT for table `sales`

--

ALTER TABLE `sales`

MODIFY `id` int(30) NOT NULL AUTO\_INCREMENT, AUTO\_INCREMENT=3;

--

-- AUTO\_INCREMENT for table `stocks`

--

ALTER TABLE `stocks`

MODIFY `id` int(30) NOT NULL AUTO\_INCREMENT, AUTO\_INCREMENT=13;

--

-- AUTO\_INCREMENT for table `suppliers`

--

ALTER TABLE `suppliers`

MODIFY `id` int(30) NOT NULL AUTO\_INCREMENT, AUTO\_INCREMENT=5;

--

-- AUTO\_INCREMENT for table `system\_settings`

--

ALTER TABLE `system\_settings`

MODIFY `id` int(30) NOT NULL AUTO\_INCREMENT, AUTO\_INCREMENT=2;

--

-- AUTO\_INCREMENT for table `users`

--

ALTER TABLE `users`

MODIFY `id` int(30) NOT NULL AUTO\_INCREMENT, AUTO\_INCREMENT=2;

COMMIT;

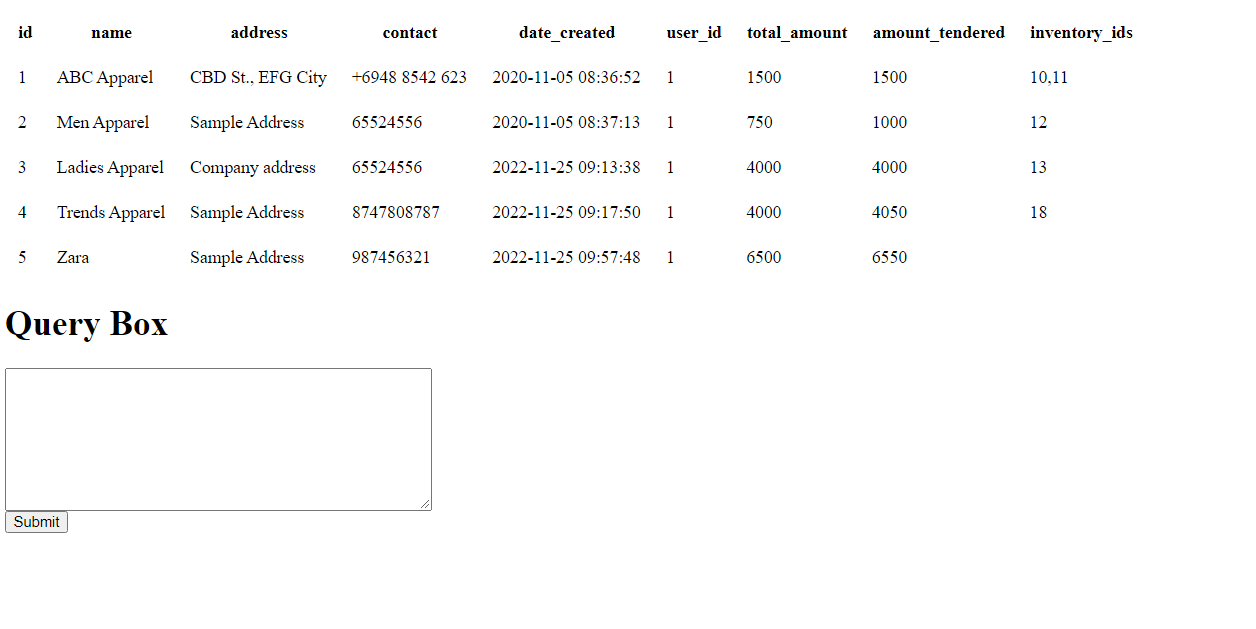
/\*!40101 SET CHARACTER\_SET\_CLIENT=@OLD\_CHARACTER\_SET\_CLIENT \*/;

/\*!40101 SET CHARACTER\_SET\_RESULTS=@OLD\_CHARACTER\_SET\_RESULTS \*/;

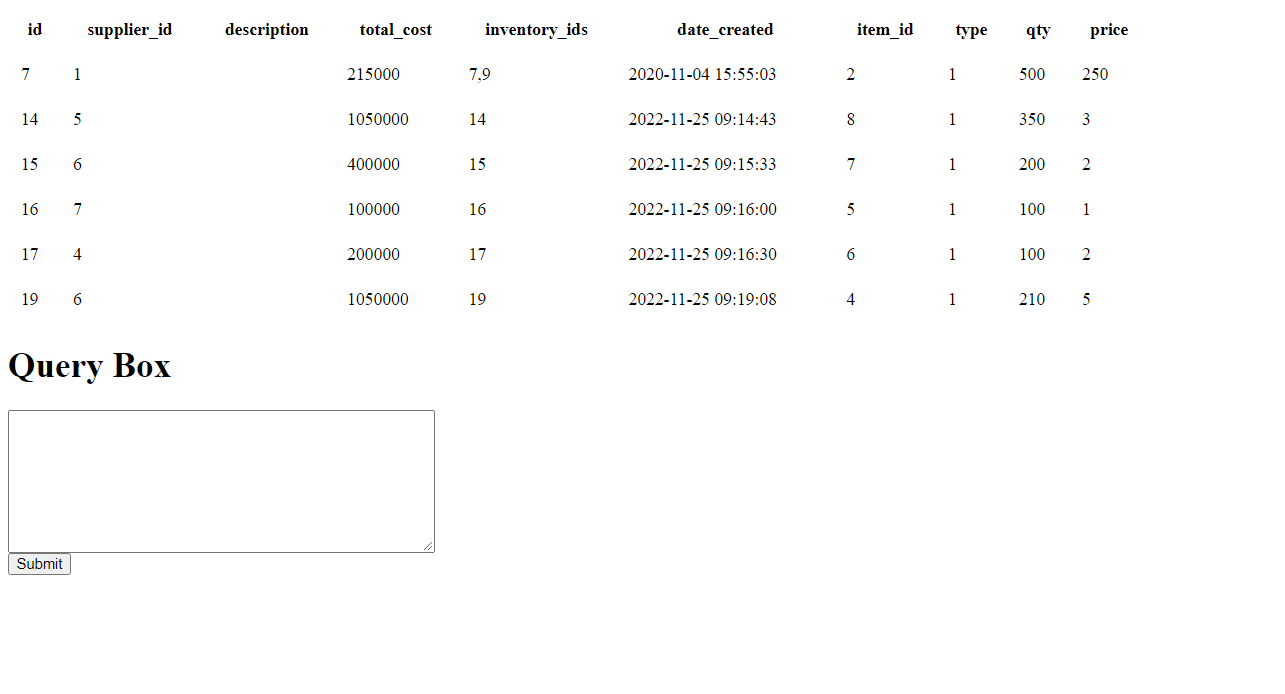
/\*!40101 SET COLLATION\_CONNECTION=@OLD\_COLLATION\_CONNECTION \*/;

**JOIN QUERIES**

* select \* from suppliers join sales on sales.id = suppliers.id;



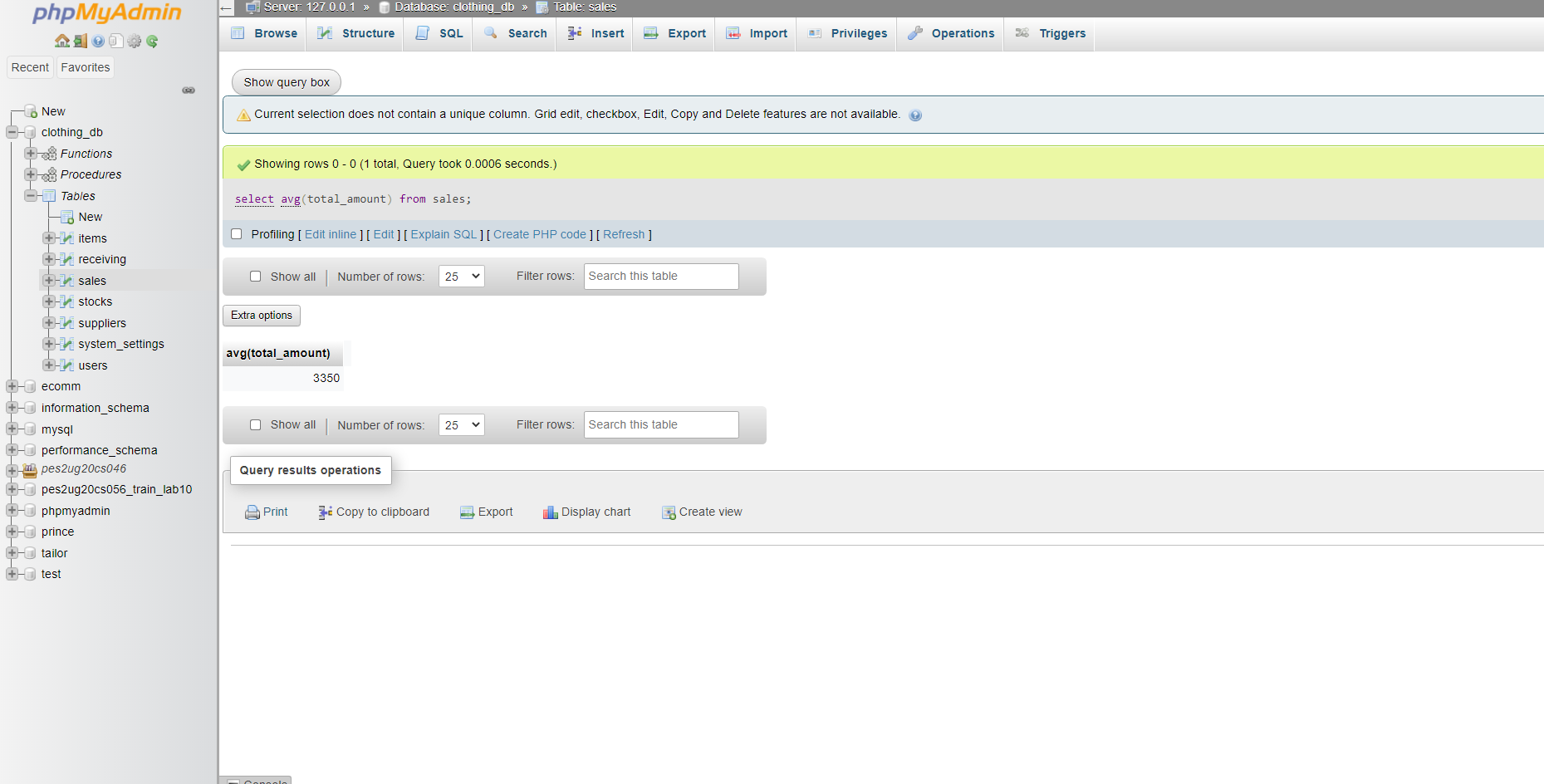
* select \* from receiving join stocks on stocks.id = receiving.inventory\_ids;

****

**AGGREGATE FUNCTIONS**

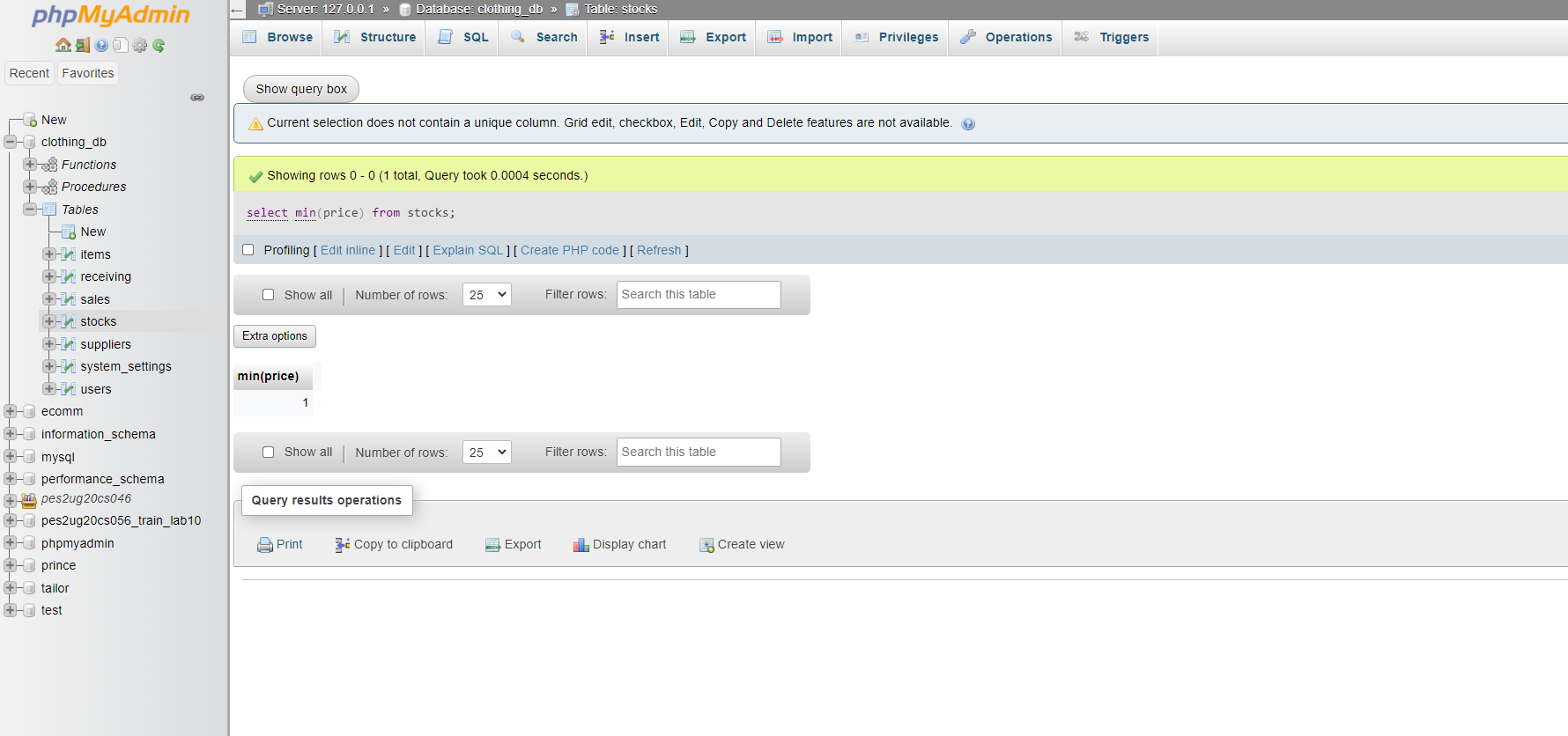
* Average of the total amount from the sales table.

select avg(total\_amount) from sales;



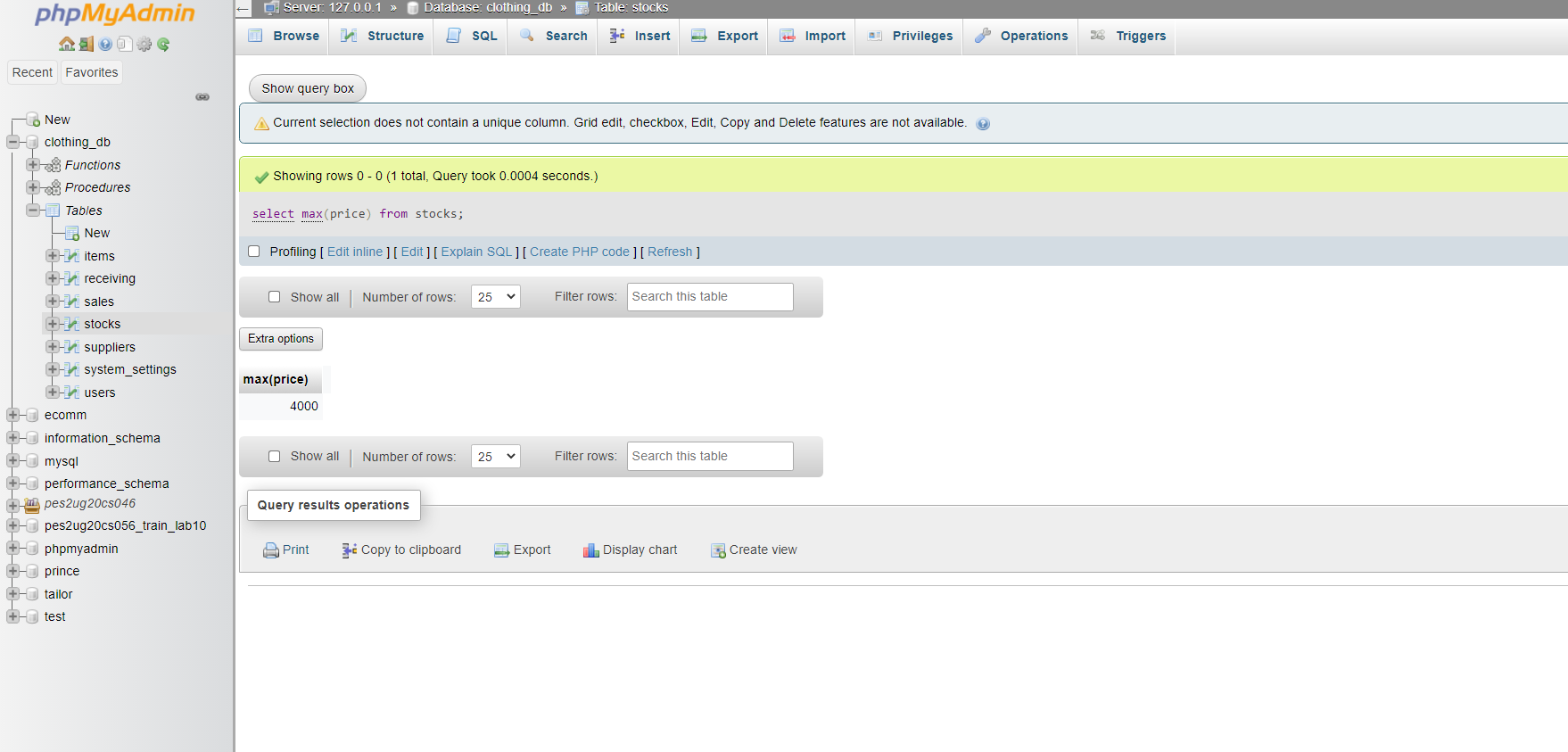
* Minimum price from Stocks

select min(price) from stocks;



* Maximum price from Stocks

select max(price) from stocks;

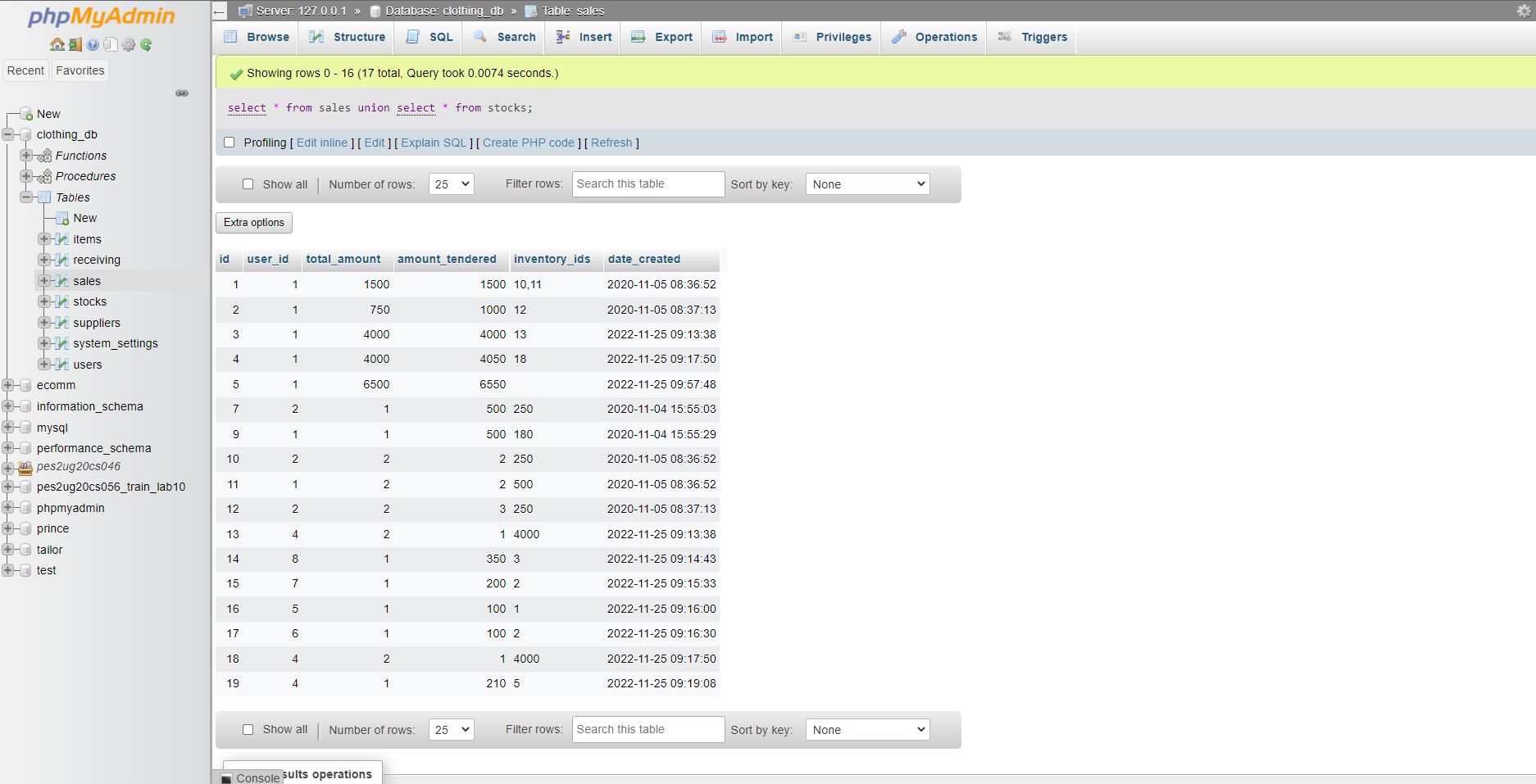


**SET OPERATIONS**

* select \* from sales

union

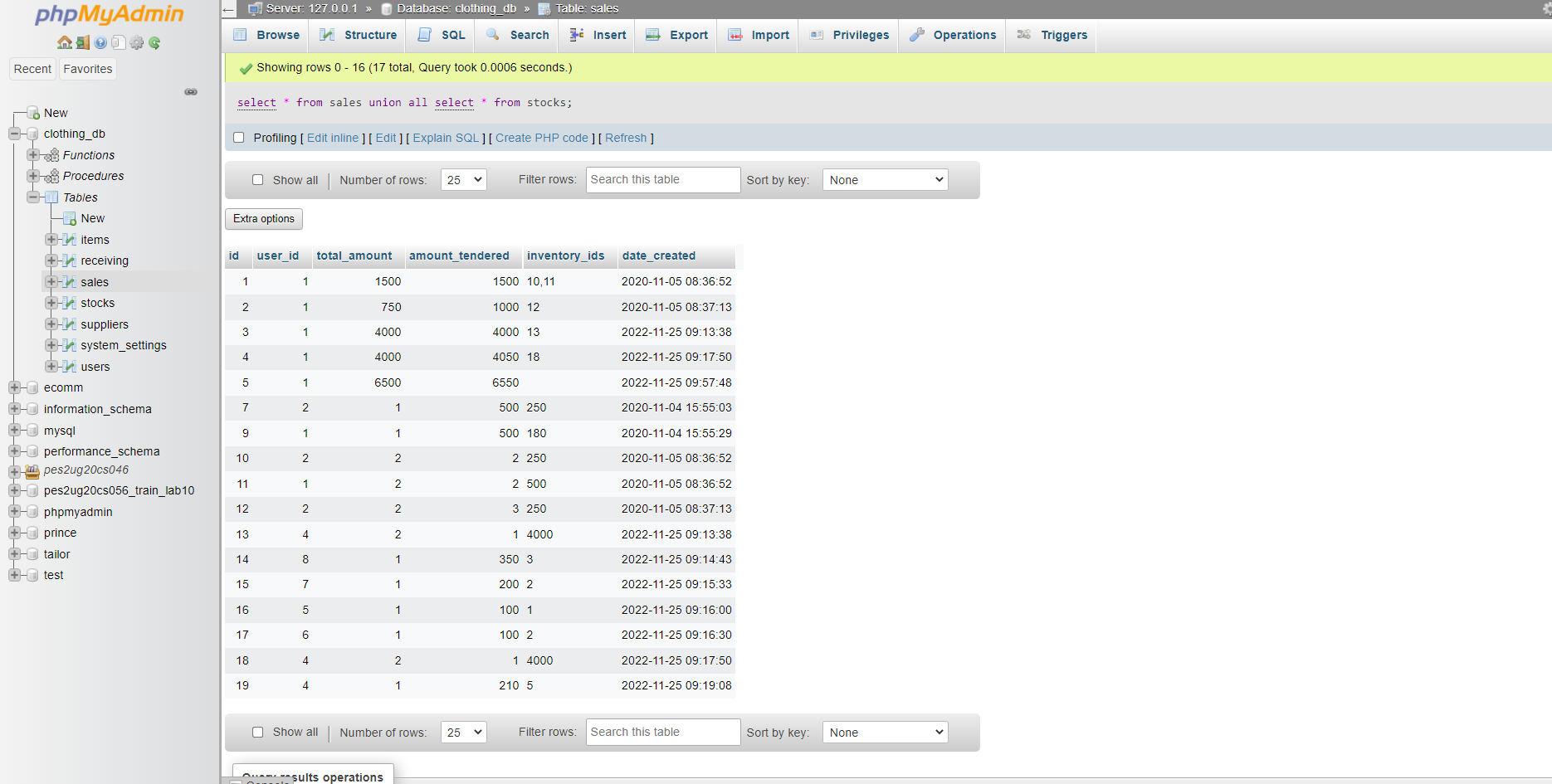
select \* from stocks;



* select \* from sales

union all

select \* from stocks;



**FUNCTIONS AND PROCEDURES**

Create a Function and Procedure. State the objective of the function / Procedure. Run and display the results

**FUNCTIONS**

BEGIN

DECLARE change\_status varchar(10);

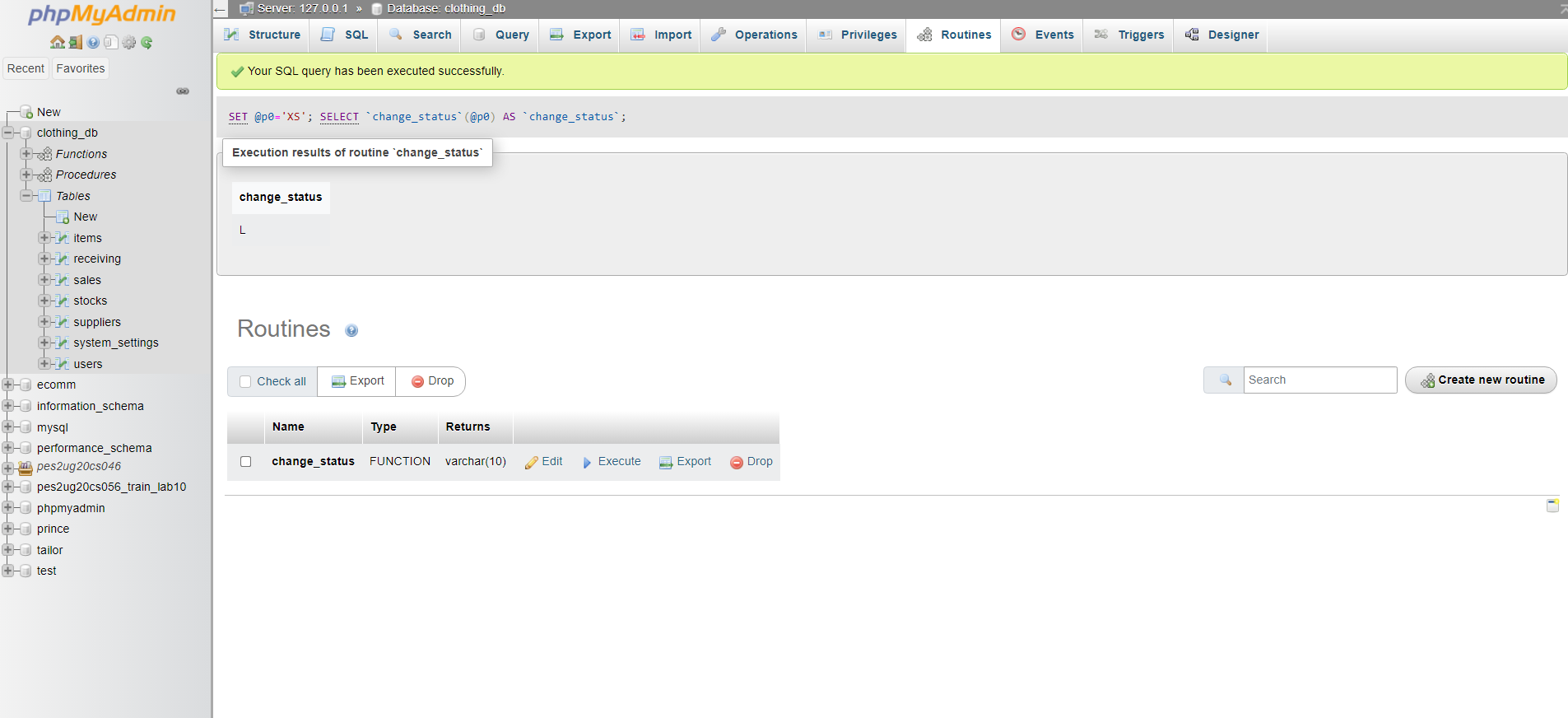
if size ='XS' THEN

SET change\_status ='L';

END IF;

RETURN (change\_status);

END

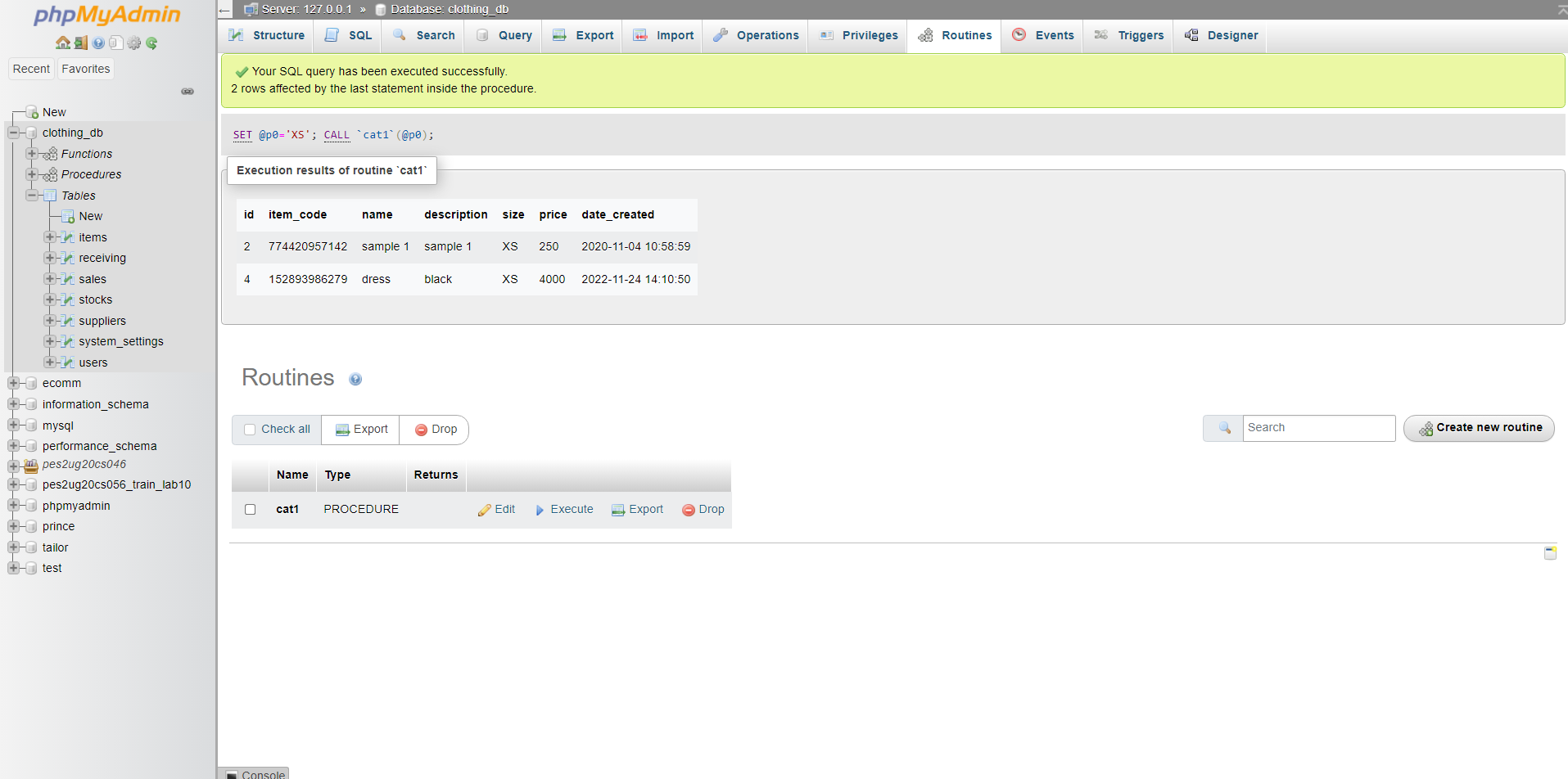
****

**PROCEDURES:**

BEGIN

SELECT \* FROM items as I WHERE I.size=var;

END



**TRIGGERS**

* When we try to add a stock into the “Stocks” table whose quantity is more than 500, there should be an error as the clothing store cannot manage stocks that large yet.

BEGIN

DECLARE errormsg varchar(255);

SET errormsg = CONCAT( 'Stock quantity can not be greater than 500');

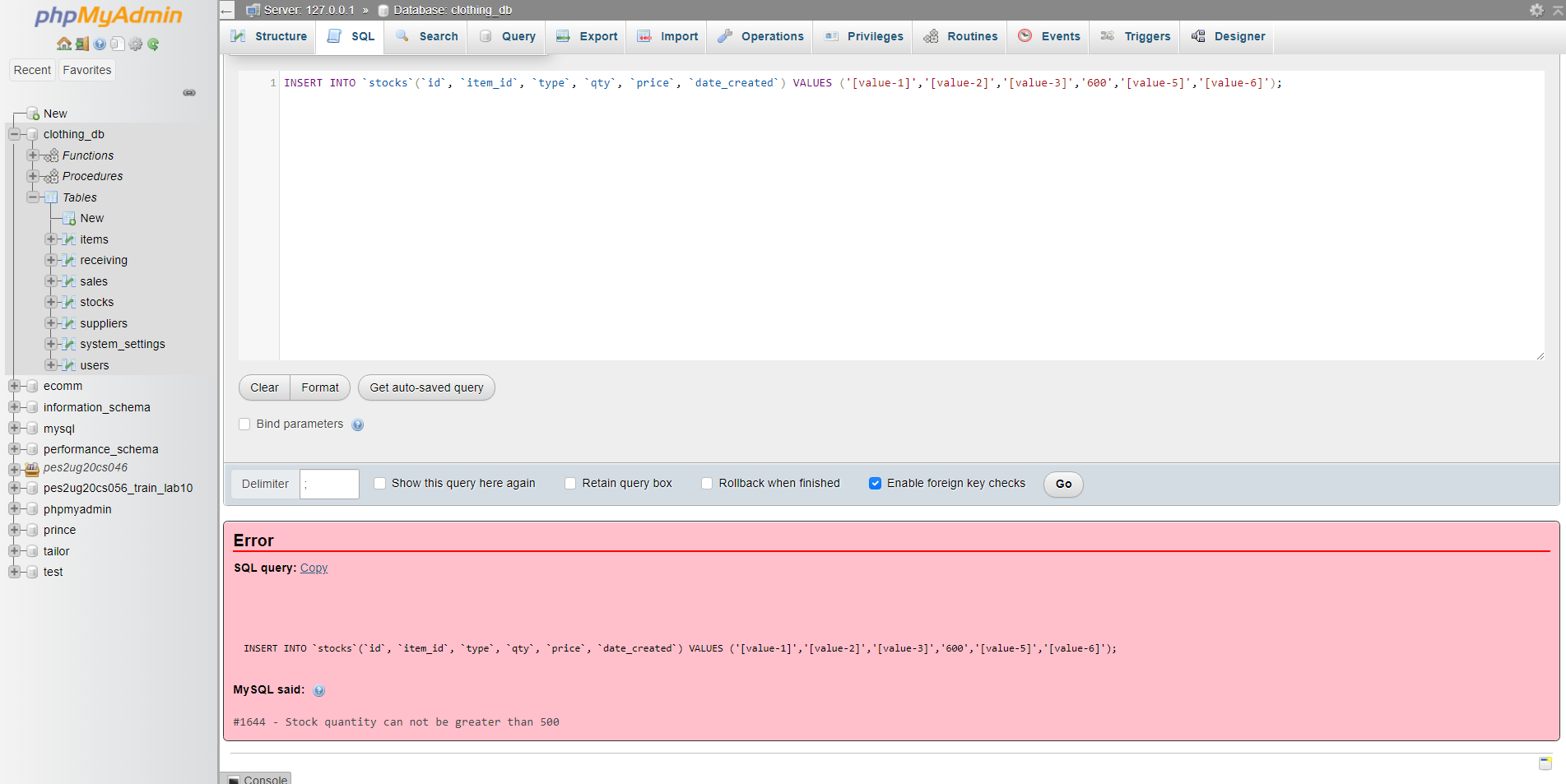
IF new.qty > 500 THEN

SIGNAL SQLSTATE '45000'

SET MESSAGE\_TEXT= errormsg;

END IF;

END



Developing a Frontend

The frontend should support

1. Addition, Modification and Deletion of records from any chosen table

2. There should be a window to accept and run any SQL statement and display the result.

