**Storing and Retrieving Data Project**

Shoes Your Weapon

Presentation: <https://prezi.com/view/VdsSVr5Xx78VKfAzT3yn/>

Text to speech AI: <https://ttsmaker.com/>

(0)

In this video, we will present the database created for the company “Shoes Your Weapon”.

(1)

This project will be focused on a fictitious portuguese online shop, Shoes Your Weapon. The company was created in 2022 by three young entrepeneurs. This business focuses on acquiring and selling limited edition shoes.

**(2)**

These products are acquired through several high-end suppliers and then sold to clients across Europe, since the store provides a delivery service.

There is a pre-established price for each product (the price already includes VAT), but there may be a promotion applied to the sale.

After the purchase, the store asks the customer to rate the whole service and the individual items bought.

An invoice is generated for each sale.

(3)

Our database, SYW, was created on the relational database management software MySQL and is composed of 11 tables to store information. They are: log, order supplier, delivery, item, order line, carrier, order, promo, supplier, customer and country.

**(4)**

In this image, it’s possible to observe the ERD for our database, which we will explore in more detail.

**(5)**

Order: this table provides information regarding the order. It provides a date for when the order was placed, who was the customer, if any promotion was applied and the total price (including VAT and delivery taxes). In this table it is also possible to have information about the delivery. Finally, there’s the rating of the order.

**(6)**

Customer: table with general information about the clients, namely contacts and address.

**(7)**

Country: table with countries from Europe; this is because Shoes Your Weapon only delivers orders to Europe and it is in this continent that the suppliers have their headquarters.

**(8)**

Supplier: table with general information about the suppliers, namely contacts and address.

**(9)**

Order\_supplier: this table provides information regarding the orders that are placed from Shoes Your Weapon to each supplier. The suppliers are specialised in specific brands and due to the nature of the business, only one type of item is ordered each time. The price attribute includes VAT and delivery costs (so it is the total cost of the order, and not necessarily the price of the shoes).

**(10)**

Delivery: this table serves both for customers delivery and suppliers delivery. It has information on the orders and the carrier making the delivery.

**(11)**

Carrier: table with information regarding the several carriers that Shoes Your Weapon uses to deliver orders to clients or receive orders made to suppliers.

**(12)**

Item: provides basic information about the items that the store sells, including who is the supplier for that specific item and the mean of the ratings (round up) given by customers.

**(13)**

Order\_line: this is a linking table between orders and items, so an order can have more than one item associated, thus respecting the first three normal forms. For each item is associated the quantity and price for the specified quantity.

**(14)**

Promo: table with promotions, with the name of the promotion and the discount given.

**(15)**

Log: This table brings together relevant information for business analysis. Since this is an international business, one of the most useful tools for us is to make a customer analysis based on countries, in order to improve marketing campaigns.

**(16)**

Now let’s analyse our database regarding connections between tables:

* Item is related to order supplier in a one to many cardinality, because one item can exist in several orders to suppliers, but in each order we only request to the supplier one specific product at a time, since this is a niche business in which we look for limited editions.
* Supplier and order supplier are connected in a one to many relationship, mandatory on both sides, because every order needs to have a supplier associated, and we only have suppliers in our database if we have made a transaction with them.
* Country is related with supplier in a one to many cardinality, in which a supplier has necessarily a country associated in the address, but not every country exists in suppliers’ addresses. Country is also related in the same way to customers. This is because our business works inside Europe: we have suppliers from Europe and we only ship to customers in Europe. Therefore, our country table has all the European countries listed, in case we have new customers or suppliers added to our database.
* Customer and order are connected in a one to many cardinality because each order is made by a client, and a client can make several orders.
* Promo is a table with several pre-studied promotions that make sense on our business and can be regularly applied. An order can have a promotion being applied, but it’s not mandatory and the same promotion can be applied to different orders. However, a promotion can exist in our database and not be used - for example, we have a specific weekend promotion, but no client makes an order on that weekend. Therefore, both ends are optional.
* Each order has mandatorily one delivery service associated, since we are an online shop that delivers directly to customers. However, our delivery table has an optional relationship with the order table. This is because the delivery table serves to keep records of deliveries from the store to the customers, but also from suppliers to our store. Therefore, each delivery is either associated with an order from the supplier or an order from a client. Thus, the relationship between order supplier and delivery is the same.
* Carrier and delivery have a one to many relationship, mandatory on both sides. The same carrier can perform several deliveries, but each delivery is made by only one carrier.
* Order line is a linking table to establish a many to many relationship between order and item, to comply with the three normal forms. This way, we can have several products being purchased on the same order. Consequently, order has a mandatory one to many cardinality with order line and item also has a mandatory relationship of one to many with order line.
* The log table has no connections to other tables. As the name says, it’s a log to keep records of transaction informations relevant to the business.

**(17)**

For this project, we had to create data to populate our database, in order to make some queries that will be further analysed.

**(18)**

The first query asks to list all the customer’s names, dates, and products bought by these customers in a range of two dates.

**(19)**

For the first query, we selected all sales made between the 5th of january of 2022 and the 4th of november of 2023.

**(20.)**

The second query asks for a list of the best three customers/products/services/ places. Since we were free to define the criteria for what means “best”, we chose to return the best customers as the ones that spend the most on our shop.

**(21)**

Our three best customers are Sophie Dubois, Luca Ricci and Mateo Ortega.

**(22.)**

On the third query, we should get the average amount of sales/bookings/ rents/deliveries for a period that involves 2 or more years, returning only one line. Once again, we chose to get the average amount of sales.

**(23)**

We selected the period between the 5th of January of 2022 and the 4th of November of 2023. Here, we can see the total sales, yearly average and monthly average. All these values are in euros.

**(24.)**

For the fourth query, we had to get the total sales/bookings/rents/deliveries by geographical location (city or country). We opted to retrieve the total sales for location.

**(25)**

We can see here the total sales by location. We can also know which country each location belongs to.

**(26.)**

The last query asks to list all the locations where products/services were sold, and the product has customer’s ratings. For this, we listed the locations where products were sold and also the average of the ratings given to orders by location.

**(27)**

The result of the last query only retrieves the countries where we sold products and whose order has already been rated, since the clients only rate the order after receiving it.

**(28)**

There are three triggers implemented on this database, regarding stock management and information in our log table.

**(29)**

The first trigger serves to update the item stock after we place an order to the supplier.

**(30)**

This trigger fires after a row is inserted on the order supplier table. These orders will always increase our stock of a certain product, as we can see on the right: after the order was placed, the stock of item 5 increased by 3 units.

**(31)**

The second trigger is very similar to the first one, in the sense that it updates the item stock, but in this case is after a client makes an order.

**(32)**

For every customer order, we have to insert a row on the order table and one or more rows on the order line, depending on how many different products the client buys. In this example, we can see the stock of the item decreasing after an order is placed.

**(33)**

The last trigger inserts on the log table information that we find relevant for business analysis. Therefore, after every customer order, a row with information regarding that order is inserted on the log.

**(34)**

This trigger is actually divided in two triggers: the first part inserts a row on the log table, but leaves the quantity of items as the default null. This has to do with the flow of our code: first we create an order and only after that can we insert the corresponding row on the order line, since the order line needs the order ID from the order table. So, after that row is inserted on the order line, we have the second part of this trigger, and update the quantity on the log table.

**(35)**

Finally, since this is a store, there’s the emission of invoices for the purchases. For this, we generate two invoices: one simplified, only with the details of the purchase and another with more information.

**(36)**

The simplified invoice includes only the details for the order, showing, for the order chosen, the model and item of the products bought, along with the unit price, quantity of each item and the total price of each item.

In this case, we have a depiction of order number 2.

**(37)**

The complete invoice has all the details regarding the order from the client. It presents information about the client and the company. It is also possible to know the tax rate and how much it comprehends, along with the sub-total. After the promotion is applied, we have the final price of the order. This invoice is also referent to order number 2.

**(38)**

Here we have a quick demonstration of our database creation on mysql, the results of our queries and the triggers working.

**(39)**

We hope you have enjoyed our business presentation. If you have any doubts or want further information, feel free to contact us!

In this video, we will present the database created for the company “Shoes Your Weapon”.

This project will be focused on a fictitious portuguese online shop, Shoes Your Weapon. The company was created in 2022 by three young entrepeneurs. This business focuses on acquiring and selling limited edition shoes.

These products are acquired through several high-end suppliers and then sold to clients across Europe, since the store provides a delivery service.

There is a pre-established price for each product (the price already includes VAT), but there may be a promotion applied to the sale.

After the purchase, the store asks the customer to rate the whole service and the individual items bought.

An invoice is generated for each sale.

Our database, SYW, was created on the relational database management software MySQL and is composed of 11 tables to store information. They are: log, order supplier, delivery, item, order line, carrier, order, promo, supplier, customer and country.

In this image, it’s possible to observe the ERD for our database, which we will explore in more detail.

Order: this table provides information regarding the order. It provides a date for when the order was placed, who was the customer, if any promotion was applied and the total price (including VAT and delivery taxes). In this table it is also possible to have information about the delivery. Finally, there’s the rating of the order.

Customer: table with general information about the clients, namely contacts and address.

Country: table with countries from Europe; this is because Shoes Your Weapon only delivers orders to Europe and it is in this continent that the suppliers have their headquarters.

Supplier: table with general information about the suppliers, namely contacts and address.

Order\_supplier: this table provides information regarding the orders that are placed from Shoes Your Weapon to each supplier. The suppliers are specialised in specific brands and due to the nature of the business, only one type of item is ordered each time. The price attribute includes VAT and delivery costs (so it is the total cost of the order, and not necessarily the price of the shoes).

Delivery: this table serves both for customers delivery and suppliers delivery. It has information on the orders and the carrier making the delivery.

Carrier: table with information regarding the several carriers that Shoes Your Weapon uses to deliver orders to clients or receive orders made to suppliers.

Item: provides basic information about the items that the store sells, including who is the supplier for that specific item and the mean of the ratings (round up) given by customers.

Order\_line: this is a linking table between orders and items, so an order can have more than one item associated, thus respecting the first three normal forms. For each item is associated the quantity and price for the specified quantity.

Promo: table with promotions, with the name of the promotion and the discount given.

Log: This table brings together relevant information for business analysis. Since this is an international business, one of the most useful tools for us is to make a customer analysis based on countries, in order to improve marketing campaigns.

Now let’s analyse our database regarding connections between tables:

Item is related to order supplier in a one to many cardinality, because one item can exist in several orders to suppliers, but in each order we only request to the supplier one specific product at a time, since this is a niche business in which we look for limited editions.

Supplier and order supplier are connected in a one to many relationship, mandatory on both sides, because every order needs to have a supplier associated, and we only have suppliers in our database if we have made a transaction with them.

Country is related with supplier in a one to many cardinality, in which a supplier has necessarily a country associated in the address, but not every country exists in suppliers’ addresses. Country is also related in the same way to customers. This is because our business works inside Europe: we have suppliers from Europe and we only ship to customers in Europe. Therefore, our country table has all the European countries listed, in case we have new customers or suppliers added to our database.

Customer and order are connected in a one to many cardinality because each order is made by a client, and a client can make several orders.

Promo is a table with several pre-studied promotions that make sense on our business and can be regularly applied. An order can have a promotion being applied, but it’s not mandatory and the same promotion can be applied to different orders. However, a promotion can exist in our database and not be used - for example, we have a specific weekend promotion, but no client makes an order on that weekend. Therefore, both ends are optional.

Each order has mandatorily one delivery service associated, since we are an online shop that delivers directly to customers. However, our delivery table has an optional relationship with the order table. This is because the delivery table serves to keep records of deliveries from the store to the customers, but also from suppliers to our store. Therefore, each delivery is either associated with an order from the supplier or an order from a client. Thus, the relationship between order supplier and delivery is the same.

Carrier and delivery have a one to many relationship, mandatory on both sides. The same carrier can perform several deliveries, but each delivery is made by only one carrier.

Order line is a linking table to establish a many to many relationship between order and item, to comply with the three normal forms. This way, we can have several products being purchased on the same order. Consequently, order has a mandatory one to many cardinality with order line and item also has a mandatory relationship of one to many with order line.

The log table has no connections to other tables. As the name says, it’s a log to keep records of transaction informations relevant to the business.

For this project, we had to create data to populate our database, in order to make some queries that will be further analysed.

The first query asks to list all the customer’s names, dates, and products bought by these customers in a range of two dates.

For the first query, we selected all sales made between the 5th of january of 2022 and the 4th of november of 2023.

The second query asks for a list of the best three customers/products/services/ places. Since we were free to define the criteria for what means “best”, we chose to return the best customers as the ones that spend the most on our shop.

Our three best customers are Sophie Dubois, Luca Ricci and Mateo Ortega.

On the third query, we should get the average amount of sales/bookings/ rents/deliveries for a period that involves 2 or more years, returning only one line. Once again, we chose to get the average amount of sales.

We selected the period between the 5th of January of 2022 and the 4th of November of 2023. Here, we can see the total sales, yearly average and monthly average. All these values are in euros.

For the fourth query, we had to get the total sales/bookings/rents/deliveries by geographical location (city or country). We opted to retrieve the total sales for location.

We can see here the total sales by location. We can also know which country each location belongs to.

The last query asks to list all the locations where products/services were sold, and the product has customer’s ratings. For this, we listed the locations where products were sold and also the average of the ratings given to orders by location.

The result of the last query only retrieves the countries where we sold products and whose order has already been rated, since the clients only rate the order after receiving it.

There are three triggers implemented on this database, regarding stock management and information in our log table.

The first trigger serves to update the item stock after we place an order to the supplier.

This trigger fires after a row is inserted on the order supplier table. These orders will always increase our stock of a certain product, as we can see on the right: after the order was placed, the stock of item 5 increased by 3 units.

The second trigger is very similar to the first one, in the sense that it updates the item stock, but in this case is after a client makes an order.

For every customer order, we have to insert a row on the order table and one or more rows on the order line, depending on how many different products the client buys. In this example, we can see the stock of the item decreasing after an order is placed.

The last trigger inserts on the log table information that we find relevant for business analysis. Therefore, after every customer order, a row with information regarding that order is inserted on the log.

This trigger is actually divided in two triggers: the first part inserts a row on the log table, but leaves the quantity of items as the default null. This has to do with the flow of our code: first we create an order and only after that can we insert the corresponding row on the order line, since the order line needs the order ID from the order table. So, after that row is inserted on the order line, we have the second part of this trigger, and update the quantity on the log table.

Finally, since this is a store, there’s the emission of invoices for the purchases. For this, we generate two invoices: one simplified, only with the details of the purchase and another with more information.

The simplified invoice includes only the details for the order, showing, for the order chosen, the model and item of the products bought, along with the unit price, quantity of each item and the total price of each item.

In this case, we have a depiction of order number 2.

The complete invoice has all the details regarding the order from the client. It presents information about the client and the company. It is also possible to know the tax rate and how much it comprehends, along with the sub-total. After the promotion is applied, we have the final price of the order. This invoice is also referent to order number 2.

Here we have a quick demonstration of our database creation on mysql, the results of our queries and the triggers working.

We hope you have enjoyed our business presentation. If you have any doubts or want further information, feel free to contact us!