

URL: <https://cs340-gilillat.pages.dev/>

Fixes based on Feedback from Step 1 and 2:

Feedback by the TAs and peer reviewers:

Step 2:

William Orona:

- Does the schema present a physical model that follows the database outline and the ER logical diagram exactly?

If yes, summarize what you see. If not, what changes would you suggest be made?

Yes, I see that we have a database that is modeled to track inventory in a warehouse and orders placed that will be shipped from the warehouse.

- Is there consistency in a) naming between overview, outline, ER and schema entity/attributes b) entities plural, attributes singular c) use of capitalization for naming?

The attribute named MSRP in the Products table is in all caps, this is not consistent with other attributes.

The inventory_line_id in the Inventory table can simply be inventory_id to be consistent with other PKs.

- Is the schema easy to read (e.g. diagram is clear and readable with relationship lines not crossed)?

Yes the schema is easy to read because there are no crossed relationship lines

- Are intersection tables properly formed (e.g. two FKs and facilitate a M:N relationship)?

Yes the Order_Details table (the intersection table) joins the M:N relationship between the Orders and Products table with an FK from each table.

- Does the sample data suggest any non-normalized issues, e.g. partial dependencies or transitive dependencies?

If yes, what are the issues, and what can be done to improve upon them? If not, summarize how the sample data shows normalization.

No, I did not see any issues that would cause any anomalies because of transitive or partial dependencies.

- Is the SQL file syntactically correct? This can be easily verified by using phpMyAdmin and your CS 340 database (do not forget to take backup of your own database before you do this!)

Yes, the SQL file is syntactically correct, I was able to use it as a source in mySQL and there were no errors. However I noticed it is not using CREATE OR REPLACE table or similar.

- In the SQL, are the data types appropriate considering the description of the attribute in the database outline?

A phone number will not need to be VARCHAR(45) consider reducing the size.

The other datatypes seem appropriate for the named attributes.

- In the SQL, are the primary and foreign keys correctly defined when compared to the Schema? Are appropriate CASCADE operations declared?

If yes, summarize how they are correctly defined, and where CASCADE operations are declared. If not, what changes should be made to the key definitions and/or what CASCADE operations would you suggest declaring?

Yes, CASCADE operations are appropriately used in tables with FKs. There is a delete operation for every foreign key except the sales_rep_id in the Orders table. This makes sense because we would like to keep all of our sales_reps in their table on a deletion in the Orders table. This is the same with the ProductID in the Order_Details table. Update operations are not used which is fine because cascading updates are typically avoided.

- In the SQL, are relationship tables present when compared to the ERD/Schema?

Yes, there is an intersection table called Order_Details present in the SQL file.

- In the SQL, is all sample data shown in the PDF INSERTED?

If yes, summarize which tables in the SQL have the sample data from the PDF inserted. If not, what sample data is missing or inconsistent?

Yes I see the same data in the SQL as the PDF. These are data from the Customers, Sales_Reps, Products, Inventory, Orders, and Order_Details table.

- Is the SQL well-structured and commented (e.g. hand authored) or not (e.g. exported from MySQL)?

Yes the SQL file is easy to read with comments before each action. However the group information is missing from the file and it does look like many of the comments were exported.

Robert Stober:

- Does the schema present a physical model that follows the database outline and the ER logical diagram exactly?

If yes, summarize what you see. If not, what changes would you suggest be made?

Yes, the schema models the outlined warehouse database. However, there is no ER diagram so that's something they need to add.

- Is there consistency in a) naming between overview, outline, ER and schema entity/attributes b) entities plural, attributes singular c) use of capitalization for naming?

If yes, summarize what the group uses for each of the above. If not, what changes would you suggest be made for improved consistency?

Yes. each attribute is lowercase with underscores as spaces. This is consistent throughout the schema and the outline. There is no ERD so I can't speak to that.

- Is the schema easy to read (e.g. diagram is clear and readable with relationship lines not crossed)?

If yes, summarize what makes the schema clear and easy to read. If not, what changes would you suggest to improve the schema's readability?

Yes, the scheme is laid out in an understandable manner. No lines are going under tables and all the relationships are properly represented.

- Are intersection tables properly formed (e.g. two FKs and facilitate a M:N relationship)?

If yes, summarize the components you see making up the properly formed intersection tables/M:N relationship. If not, what changes would you suggest to improve them?

Yes, the two intersection tables are properly represented with two foreign keys in each of them and facilitate M:N relationships.

- Does the sample data suggest any non-normalized issues, e.g. partial dependencies or transitive dependencies?

If yes, what are the issues, and what can be done to improve upon them? If not, summarize how the sample data shows normalization.

There are no repeating values where it's possible to minimize. There are no attributes that are dependent on other attributes(besides primary keys) and each entity represents its own idea. The schema is in 3NF.

- Is the SQL file syntactically correct? This can be easily verified by using phpMyAdmin and your CS 340 database (do not forget to take backup of your own database before you do this!)

If yes, summarize what you see upon importing (e.g. any status message, which tables were imported AND populated with their corresponding sample data, etc.). If not, what happened with the import/what appears to be the issue(s) with the file that are causing this?

The tables Order_Details, Orders, Inventory, Products, Sales_Reps, and Customers are all imported and populated with the sample data.

- In the SQL, are the data types appropriate considering the description of the attribute in the database outline?

If yes, summarize why they seem fitting. If not, what changes would you suggest based on the attribute descriptions?

Yes, the numbers that need precision are decimal, and the numbers that need uniqueness(ids) are integers. Varchars are given the appropriate amount of bytes for their field.

- In the SQL, are the primary and foreign keys correctly defined when compared to the Schema? Are appropriate CASCADE operations declared?

If yes, summarize how they are correctly defined, and where CASCADE operations are declared. If not, what changes should be made to the key definitions and/or what CASCADE operations would you suggest declaring?

Yes, primary keys are all listed per table and foreign keys are listed in the appropriate tables as well. Each of these foreign keys is set to CASCADE on delete but not update. This makes sense as foreign keys shouldn't be updated in tables.

- In the SQL, are relationship tables present when compared to the ERD/Schema?

If yes, summarize the relationship tables present in both the SQL and the ERD/Schema. If not, which relationship tables aren't present?

Yes, there are two M:N relationships(Sales_rep->customers and orders->products). The remaining relationship is 1:M(products->inventory)

- In the SQL, is all sample data shown in the PDF INSERTED?

If yes, summarize which tables in the SQL have the sample data from the PDF inserted. If not, what sample data is missing or inconsistent?

Yes, the data of the table Order_Details, Orders, Inventory, Products, Sales_Reps, and Customers are all Inserted.

- Is the SQL well-structured and commented (e.g. hand authored) or not (e.g. exported from MySQL)?

If yes, summarize the structure and comments you see. If not, what changes would you suggest to improve the SQL file?

The table appears to be exported from MySQL, but is very nicely cleaned up. I see no changes to be made.

Angela Wu:

- Does the schema present a physical model that follows the database outline and the ER logical diagram exactly?

If yes, summarize what you see. If not, what changes would you suggest be made?

The schema matches the database outline the entity of Customers, Sales_Reps, Orders, Order_Details, Products, and Inventory and the attributes themselves matches with each other as well. The only issue is there is no ERD available. The ERD is similar to Schema but it normally show just the pk and could be with fk with relationships and attributes are not necessarily needed. This can be created in draw.io or lucidchart or other preferences.

- Is there consistency in a) naming between overview, outline, ER and schema entity/attributes b) entities plural, attributes singular c) use of capitalization for naming?

If yes, summarize what the group uses for each of the above. If not, what changes would you suggest be made for improved consistency?

Yes the entity tables are in PASCAL CASE and the attributes are in snake_case like first_name and Sales_Reps. The entities are also plural and the attributes are singular. They are also consistent with the naming between database outline, ERD, and Schema.

- Is the schema easy to read (e.g. diagram is clear and readable with relationship lines not crossed)?

If yes, summarize what makes the schema clear and easy to read. If not, what changes would you suggest to improve the schema's readability?

Yes, the Schema is easy to read, the lines are not crossed and enough space is provided. The relationships are also presented and follows the same explanation in the database outline.

- Are intersection tables properly formed (e.g. two FKs and facilitate a M:N relationship)?

If yes, summarize the components you see making up the properly formed intersection tables/M:N relationship. If not, what changes would you suggest to improve them?

Yes, the Order_Details is the intersection of the M:N relationship between Products and Orders. It contains the PK of the intersection table and the FK of the Orders id and Products is with its associated attributes that are required as well.

- Does the sample data suggest any non-normalized issues, e.g. partial dependencies or transitive dependencies?

If yes, what are the issues, and what can be done to improve upon them? If not, summarize how the sample data shows normalization.

The sample data follows the normalization of 3NF where there is no repeating values and attributes are not dependent on other attributes. The tables themselves also holds unique information that does not overlap with other tables.

- Is the SQL file syntactically correct? This can be easily verified by using phpMyAdmin and your CS 340 database (do not forget to take backup of your own database before you do this!)

If yes, summarize what you see upon importing (e.g. any status message, which tables were imported AND populated with their corresponding sample data, etc.). If not, what happened with the import/what appears to be the issue(s) with the file that are causing this?

Yes, the SQL files are syntactically correct. In the beginning and end the SET FOREIGN_KEY was set to the correct value as well as SET AUTOCOMMIT. Before tables are created it drops the tables to clear out the tables. Then the tables were created and the following information was inserted. Terminal also shows that things were inserted as well as created table.

- In the SQL, are the data types appropriate considering the description of the attribute in the database outline?

If yes, summarize why they seem fitting. If not, what changes would you suggest based on the attribute descriptions?

Yes, the appropriate data types were utilized like varchar(45), and Decimal(10,2) for MSRP wholesale_cost, unit_price, line_total.

- In the SQL, are the primary and foreign keys correctly defined when compared to the Schema? Are appropriate CASCADE operations declared?

If yes, summarize how they are correctly defined, and where CASCADE operations are declared. If not, what changes should be made to the key definitions and/or what CASCADE operations would you suggest declaring?

Yes the CASCADE operations are declared. All table has its own primary key including the intersection table for the M:N and set to CASCADE to delete and no update. It also has the necessary foreign keys that relates to the relationship.

- In the SQL, are relationship tables present when compared to the ERD/Schema?

If yes, summarize the relationship tables present in both the SQL and the ERD/Schema. If not, which relationship tables aren't present?

Yes, the tables and the attributes show the same as the Scheme after the SQL was created. The relationships are all the same. The ERD does not exist so can not determine if it is the same as the ERD.

- In the SQL, is all sample data shown in the PDF INSERTED?

If yes, summarize which tables in the SQL have the sample data from the PDF inserted. If not, what sample data is missing or inconsistent?

Yes, the data shown are the same as the PDF INSERTED. Customers, Sales_Reps, Products, Inventory, Orders, and Order_Details insert information align with the insert from the .sql file.

- Is the SQL well-structured and commented (e.g. hand authored) or not (e.g. exported from MySQL)?

If yes, summarize the structure and comments you see. If not, what changes would you suggest to improve the SQL file?

Yes, comment of insert and creation of table are commented and also for dropping and disabling foreign key checks and commits are also commented.

OVERALL SUMMARY

- Besides the missing ERD the group submission shows good understanding of the normalization process, consistent naming conventions, creation of schema and .sql to perform create, table, checks, and drop, and commits. Overall shows understanding of the information learned in class so far.

Kai Black:

Does the schema present a physical model that follows the database outline and the ER logical diagram exactly?

If yes, summarize what you see. If not, what changes would you suggest be made?

- Yes, the schema accurately reflects the database outline and the ER diagram. The tables (Customers, Sales_Reps, Orders, Products, Order_Details, Inventory) and their respective attributes match the entities and relationships as described. The relationships, including 1:M and M:N, are properly represented.

Is there consistency in a) naming between overview, outline, ER and schema entity/attributes b) entities plural, attributes singular c) use of capitalization for naming?

If yes, summarize what the group uses for each of the above. If not, what changes would you suggest be made for improved consistency?

- Yes, the schema is consistent in naming conventions. Entity names are plural (e.g., Customers, Sales_Reps, Orders), while attributes are singular (e.g., customer_id, email). The capitalization is also consistent, with SQL keywords in uppercase and table names in PascalCase.

Is the schema easy to read (e.g. diagram is clear and readable with relationship lines not crossed)?

If yes, summarize what makes the schema clear and easy to read. If not, what changes would you suggest to improve the schema's readability?

- Yes, the schema is clear and easy to read. The diagram is well-structured, with no crossing relationship lines, making the flow of information easily understandable. Tables are logically grouped, and the relationships between them are clearly defined.

Are intersection tables properly formed (e.g. two FKs and facilitate a M:N relationship)?

If yes, summarize the components you see making up the properly formed intersection tables/M:N relationship. If not, what changes would you suggest to improve them?

- Yes, the intersection table Order_Details is correctly formed. It includes two foreign keys, order_id and product_id, which establish the M:N relationship between the Orders and Products tables. Additional attributes like quantity, unit_price, and line_total are included to store necessary transaction details.

Does the sample data suggest any non-normalized issues, e.g. partial dependencies or transitive dependencies?

If yes, what are the issues, and what can be done to improve upon them? If not, summarize how the sample data shows normalization.

- No, the sample data does not exhibit any non-normalized issues. There are no signs of partial or transitive dependencies. Each table contains only relevant attributes, and the relationships are properly maintained through foreign keys.

Is the SQL file syntactically correct? This can be easily verified by using phpMyAdmin and your CS 340 database (do not forget to take backup of your own database before you do this!)

If yes, summarize what you see upon importing (e.g. any status message, which tables were imported AND populated with their corresponding sample data, etc.). If not, what happened with the import/what appears to be the issue(s) with the file that are causing this?

- Yes, the SQL file is syntactically correct. Tables are created with appropriate data types, constraints, and indexes. Foreign key relationships are accurately defined with ON DELETE CASCADE and ON UPDATE NO ACTION where applicable.

In the SQL, are the data types appropriate considering the description of the attribute in the database outline?

If yes, summarize why they seem fitting. If not, what changes would you suggest based on the attribute descriptions?

- Yes, the data types are appropriate for the given attributes. The schema uses INT for IDs, VARCHAR for textual data, DECIMAL for monetary values, and DATETIME for dates. The VARCHAR lengths seem reasonable for the data they are intended to store.

In the SQL, are the primary and foreign keys correctly defined when compared to the Schema? Are appropriate CASCADE operations declared?

If yes, summarize how they are correctly defined, and where CASCADE operations are declared. If not, what changes should be made to the key definitions and/or what CASCADE operations would you suggest declaring?

- Yes, the primary and foreign keys are correctly defined. Primary keys are assigned as INT with AUTO_INCREMENT, which ensures unique identification of records. Foreign keys are properly referenced, and appropriate CASCADE operations (e.g., ON DELETE CASCADE for Orders and Inventory) are defined to maintain data integrity and enable proper handling of related records upon deletion.

In the SQL, are relationship tables present when compared to the ERD/Schema?

If yes, summarize the relationship tables present in both the SQL and the ERD/Schema. If not, which relationship tables aren't present?

- Yes, the relationship tables are present and correctly implemented. The Order_Details table serves as the intersection table for the M:N relationship between Orders and Products. All relationships outlined in the ER diagram are faithfully represented in the schema.

In the SQL, is all sample data shown in the PDF INSERTED?

If yes, summarize which tables in the SQL have the sample data from the PDF inserted. If not, what sample data is missing or inconsistent?

- Yes, all sample data from the PDF is successfully inserted into the corresponding tables. Data has been added to Customers, Sales_Reps, Products, Inventory, Orders, and Order_Details, and it matches the sample data provided in the database outline.

Is the SQL well-structured and commented (e.g. hand authored) or not (e.g. exported from MySQL)?

If yes, summarize the structure and comments you see. If not, what changes would you suggest to improve the SQL file?

- Yes, the SQL file is well-structured and includes enough comments. The file is divided into clear sections, such as table creation and data insertion, with comments explaining each section's purpose. For example, comments like "-- Insert sample Customer data" help clarify the intent behind each query.

Step 1:

TA Matthew Harangody:

This is great work!

Please be sure to include the feedback from the TA and if you did or did not take any suggestions based off of it.

Please be sure to address all feedback from reviewers (solid lines from Callum).

Let me know if you have any questions.

Thanks!

Anjelica Cucchiara:

"Hello Group 141.

Your draft describes the problem as the need for a comprehensive tool to manage orders, inventory, and customer information. It shows inefficiencies in the current system and aims to centralize sales data and streamline operations. The proposed system addresses the need for employees to efficiently view customer orders, inventory, and sales data. It also aims to manage thousands of transaction records and provide aggregate sales insights on a dashboard.

Your draft includes five entities: Customers, Orders, Order_Details (bridge table), Products, and Inventory. Relationships are shown, such as the 1:M relationship between Customers and Orders, and the M:M relationship between Products and Orders via the Order_Details table. The naming conventions are consistent, with entities pluralized (e.g., Orders, Customers) and attributes singular (e.g., customer_id, product_id).

The project draft is organized nicely and covers all the essential aspects of a database-driven website. It lays out a clear problem statement, has a logical structure for the entities, and defines the relationships in a thoughtful way."

Callum Pickard:

"Greetings group 141,

Your proposed database design is clear and simple and seems to manage relationships really well. I was secretly hoping to find a problematic attribute or relationship definition in there at least, so I could more easily provide critical feedback and some constructive criticism, but it seems you have covered all your bases really well. In this case, you are getting nothing but praise.

Nice work using some big numbers to give a sense of the scale. It really set the tone and made me see the structure, and division of data within the database as necessary. I'm pretty sure with the way you have separated your concerns across tables that this kind of traffic would be manageable.

I'm not 100% sure which is your nullable relationship as you haven't pointed this out explicitly, and all your foreign keys are specifically NOT_NULL.

Since you are using MySQL workbench, I'll also suggest that you be wary of using solid lines to connect your tables, since they represent identifying relationships in MySQL workbench. This is true for your order_details table, where your foreign keys are part of its primary key, but not for the others. This is obviously not vital at the moment, but good to be aware of.

And I'm presuming you chose to use your own styling as opposed to the style guide suggestions, which are totally just suggestions, but just be wary of using the same case for the tables and the attributes. For example, customers (table) and address (attribute), use the same naming convention. I know for myself when using programming

languages, I'm often saved the fact that classes and other elements being capitalized cause me to recognize their uniqueness when compared to variables.

All the best, and once again, great job,
Callum Pickard”

Havel Nimith Konda:

“Great work on laying out the plan for the Westside Auto Parts Sales Management System. The overall concept of managing orders, inventory, and customer info through the proposed system seems well thought out and would definitely improve efficiency for employees. I really like how you’ve structured the database outline and relationships between the entities, which will allow for a smooth flow of information across the system.

For the Customers table, the fields look good. I see that you're using customer_id as the primary key, which is standard and appropriate for identifying each customer. The foreign key relationship with Orders makes sense, ensuring that each order can be traced back to a customer.

The Orders table seems to cover all the necessary details, and the relationship to Customers is clear. The fact that you've set up a Order_Details bridge table for the many-to-many relationship between Orders and Products is spot on. This setup will allow for flexibility in tracking multiple products per order and storing detailed info about quantities and line totals.

The Products table also looks solid. Using a product_id as the primary key and linking it to both Orders and Inventory makes sense. It's good that you're tracking both MSRP and wholesale cost, which should be helpful for pricing and profit analysis down the line. For the Inventory table, I like how you're capturing the stock across multiple warehouses. It's great that you're keeping track of the warehouse location, which will allow employees to quickly see where products are stored.

One thing to consider is whether any additional constraints (like CHECK constraints) might be useful for data validation. For example, you could add a constraint on Inventory to make sure the quantity doesn't go below 0 when managing stock.”

Chloe Rice:

- Does the overview describe what problem is to be solved by a website with DB back end? If yes, summarize. If not, what changes would better support describing the problem to be solved?

The overview describes how the database will help to make the management of Westside Auto Parts more streamlined and organized by storing sorting thousands of transaction records.

- Does the overview list specific facts? If yes, summarize what the facts illustrate about the proposed DB solution. If not, what facts would better support illustrating the scope and scale of the proposed DB solution?

Yes the overview lists specific facts. The overview describes how the database can help employees to manage orders, inventory, and customer information with greater flexibility

and ease. The system will serve to streamline management of the company by organizing transactions

- Are at least four entities described, and does each one represent a single idea to be stored as a list? If yes, summarize. If not, based on the course material, what changes can you suggest to improve?

Yes, a total of five entities are described in this database proposal plan. The customers entity keeps track of the customers at Westside Auto Parts, including their contact details. The orders entity records the details of orders the company receives, including the data and time in which it was received and the customer who ordered it. The order details list the products within an order, and other details like the quantity of each product. The products entity records the details of each product the company sells, and the inventory tracks how much of each product is in a particular warehouse.

- Does the outline of entity details describe the purpose of each, list attribute datatypes and constraints, and describe relationships between entities? If yes, summarize. If not, based on the course material, what changes can you suggest to improve?

The outline of entity details the purpose of each and lists all of the datatypes and constraints, and also describes the relationships between entities and the types of tables and keys they use if necessary.

- Are 1:M relationships correctly formulated? Is there at least one M:M relationship? Does the ERD present a logical view of the database? If yes, summarize. If not, based on the course material, what changes can you suggest to improve?

Yes, the relationships are correctly formulated and there is at least one to M:M relationship which is the relationship between orders and products. The ERD is very well structured and represents a logical view of the database, with the order details making a logical bridge between orders and products.

- Is there consistency in a) naming between overview and entity/attributes b) entities plural, attributes singular c) use of capitalization for naming? If yes, summarize. If not, based on the course material, what changes can you suggest to improve?

Yes, there is consistency between the overview and the entity names. Entities are plural and attributes are singular. The name of each entity and attribute is either fully lowercase or completely capitalized (if it's an acronym), which is a simple way to keep the names consistent and clean.

Overall, the project is very nicely organized and gets straight to the point without focusing on unnecessary details that might overcomplicate or clutter the project.”

TA Matthew Harangody:

This is a really great start! I just have a few minor suggestions:

Make sure to include your name and group information in the initial post on ED, not just in the pdf link.

You can either add a composite primary key (of the two foreign keys) in the order_details table or add its own unique PK, but its best practice in design to have a primary key in an intersection table.

Your quantity attribute in the Inventory table is int in the outline and varchar in the ER diagram, make sure to match them up.

This is great work, I look forward to seeing your project progress!

Actions based on the feedback:

While Anjelica Chloe didn't provide any needed updates, Callum pointed out that we were missing the 1:N relationship and Havel suggested a check constraint on the quantity attribute of the Inventory table. We took this into account and incorporated an additional table, Sales_Reps, to satisfy this requirement by providing a nullable foreign key from this table to the Orders table. We added a primary key to the order_details table and corrected the discrepancy with the Inventory table being int in the outline and varchar in the ERD. Also updated missing issues with not all the feedback being on the document.

Upgrades to the Draft version:

Step 1:

We implemented the new Sales_Reps table and we also introduced a check on the quantity attribute of the Inventory table. We also added a unit_price and a PK, order_details_id, to the Order_Details table.

Step 2:

Added in comments about the team in the SQL files and added in an ERD to support the Schema.

Project Outline and Database Outline - Updated Version:

Trevor Gililland
Alyshia Benson
Group 141
CS340 - Winter 2025

Westside Auto Parts Sales Management System

Overview

Westside Auto Parts is a small online retailer generating approximately \$750,000 in annual sales. The company relies on a third-party service for shipping. The proposed

website will utilize a database and will serve as a comprehensive tool for employees, enabling them to manage orders, inventory, and customer information with ease. Employees can view customer orders, inventory levels by warehouse, and detailed product and customer data, ensuring they know what products are in stock and where they are located. They will do this by using the Customers table to look at any customer related information, the Orders table and the Order_Details table to look at information regarding orders. The Sales_Rep table will record which employees made what sales. The Inventory table allows employees to check what is in stock and at which warehouse locations. Employees can also use the Products table to get more information about a specific product.

This sales management system will also streamline sales management by storing and organizing the need for up to 50,000 transaction records, making them readily available for review. This functionality allows employees to access a customer's order history, aiding in reorder decisions and facilitating returns. Additionally, the website will provide a dashboard on the main landing page that displays aggregate sales data for the day, month, and year, offering valuable insights at a glance.

Database Outline

- Customers: This entity records the details of the customers we do business with.
 - customer_id: int, auto_increment, unique, not NULL, PK
 - email: varchar, not NULL
 - first_name: varchar, not NULL
 - last_name: varchar, not NULL
 - address: varchar, not NULL
 - phone: varchar
 - Relationship: a 1:M relationship between Customers and Orders with customer_id as a FK inside of Orders
- Sales_Reps: This entity records the sales representative's information that took a customer's order.
 - sales_rep_id: int, auto_increment, unique, not NULL, PK
 - email: varchar, not NULL
 - first_name: varchar, not NULL
 - last_name: varchar, not NULL
 - Relationship: a 1:M relationship Sales_Reps and Orders with sales_rep_id as a FK inside of Orders
- Orders: This entity records the basic information about the orders we receive, such as date, total cost, customer and sales representative.
 - order_id: int, auto_increment, unique, not NULL, PK
 - order_date: datetime, not NULL
 - order_total: decimal, not NULL
 - customer_id: int, not NULL, FK
 - sales_rep_id: int, FK

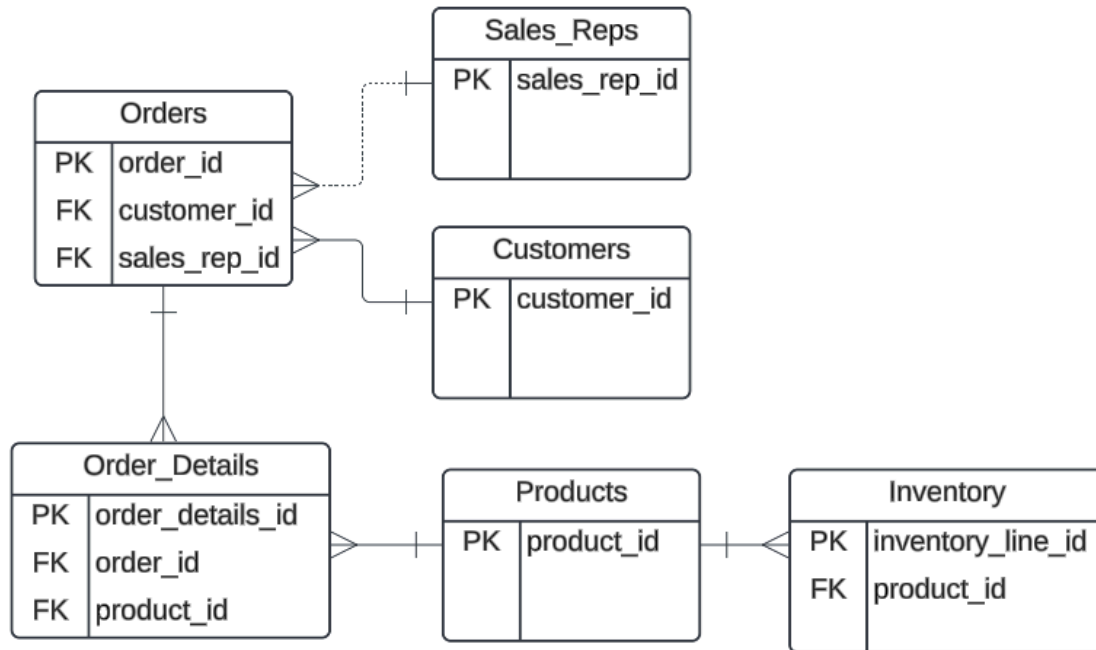
- Relationship: a M:1 relationship between Customers and Orders with customer_id as a FK inside of Orders, a M:1 relationship between Sales_Reps and Orders with sales_rep_id as a FK inside of Orders, and a M:N relationship between Products and Orders with an intersection table, Order Details, between containing both the order_id and product_id as FKs

- Order_Details: This entity is an intersection table to map various products to various orders, this also shows the exact product that was ordered, as well as information about the quantity and price paid.
 - order_details_id: int, auto_increment, unique, not NULL, PK
 - order_id: int, not NULL, FK
 - product_id: int, not NULL, FK
 - quantity: int, not NULL
 - unit_price: decimal, not NULL
 - line_total: decimal, not NULL
 - Relationship: Intersection table between Orders and Products due to the M:N relationship, containing both the order_id and product_id as FK

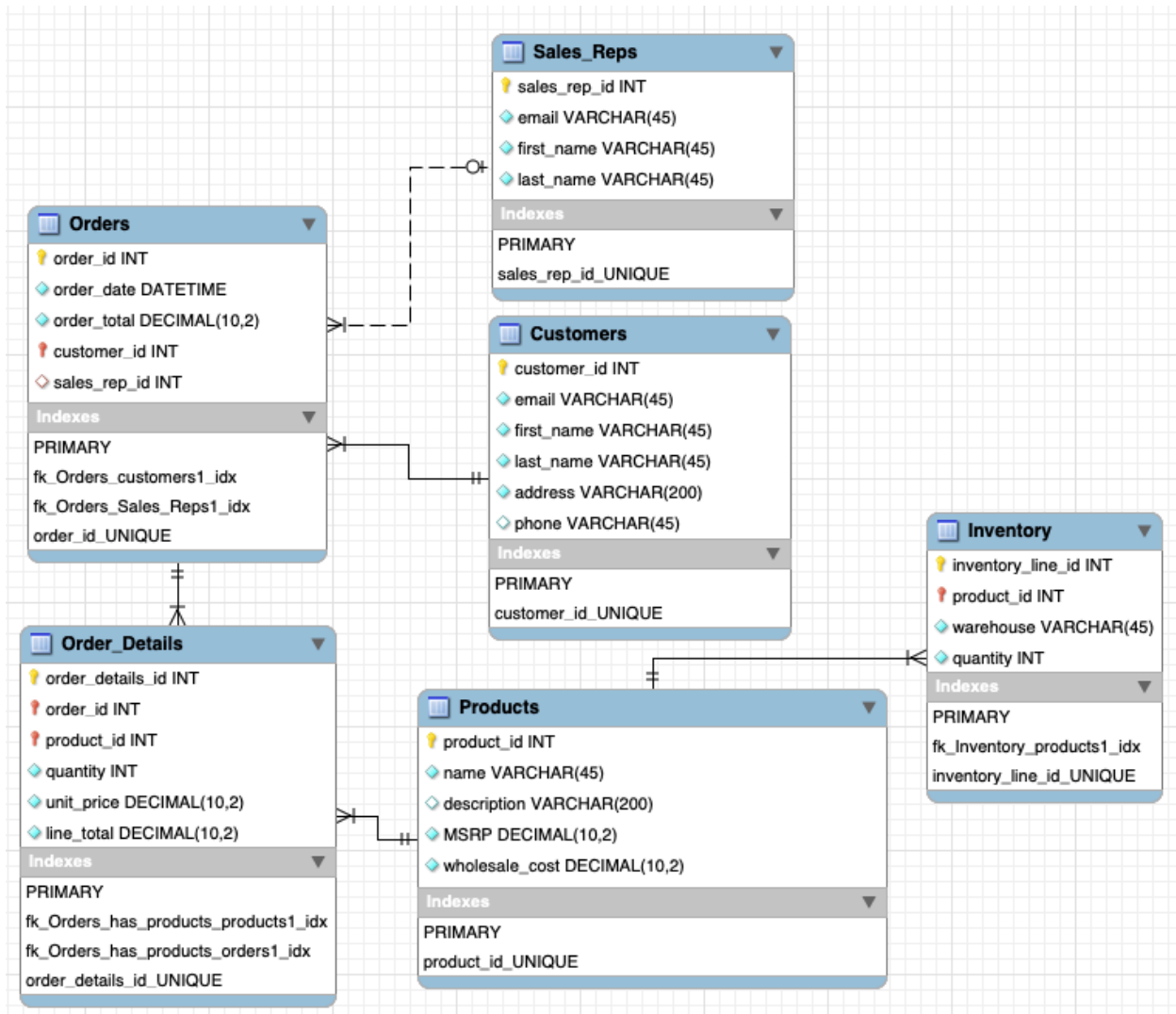
- Products: This entity records the details of each of the products we sell.
 - product_id: int, auto_increment, unique, not NULL, PK
 - name: varchar, not NULL
 - description: varchar
 - MSRP: decimal, not NULL
 - wholesale_cost: decimal, not NULL
 - Relationship: relationship between Products and Orders with an intersection table, Order Details, between containing both the order_id and product_id as a FK, 1:M relationship between Products and Inventory with product_id as a FK inside of Inventory

- Inventory: This entity records the inventory of products across both warehouses and allows you to see which warehouse a specific product is in.
 - inventory_line_id: int, auto_increment, unique, not NULL, PK
 - product_id: int, not NULL, FK
 - warehouse: varchar, not NULL
 - quantity: int, CHECK(quantity >= 0)
 - Relationship: M:1 relationship between Products and Inventory with product_id as a FK inside of Inventory

Entity-Relationship Diagram:



Schema:



Example Data:

Customers Table (customer_id, email, first_name, last_name, address, phone):

(1, 'jane@gmail.com', 'Jane', 'Smith', '1443 Main St Los Angeles CA 94534', '310-225-5878'),
(2, 'john@gmail.com', 'John', 'Robertson', '4642 1st St Corvallis OR 97331', '541-547-5568'),
(3, 'mary@hotmail.com', 'Mary', 'Jackson', '3867 Hope St Seattle WA 98309', '206-884-2345')

Sales_Reps Table (sales_rep_id, email, first_name, last_name):

(1, 'mark@autosales.com', 'Mark', 'Robinson'),
(2, 'sally@autosales.com', 'Sally', 'Davis'),
(3, 'steve@autosales.com', 'Steve', 'Jefferson')

Products Table (product_id, name, description, MSRP, wholesale_cost):

(1, 'Goodyear Tire', 'Single P225 70R16 Goodyear tire', 149.99, 89.99),
(2, 'Magnaflow Catalytic Converter', 'Magnaflow catalytic converter for Toyota Camry vehicles', 1299.99, 599.99),
(3, 'Brembo Brake Caliper Set', 'Two front Red Brembo brake calipers for Porsche 911 vehicles', 3699.99, 2199.99)

Inventory Table (inventory_line_id, product_id, warehouse, quantity):

(1, 1, 'AW4', 100),
(2, 2, 'AW4', 500),
(3, 3, 'AW4', 100)

Orders Table (order_id, order_date, order_total, customer_id, sales_rep_id):

(1, 20240506, 1899.95, 1, 1),
(2, 20240607, 1299.99, 2, NULL),
(3, 20250107, 3699.99, 3, 2)

Order_Details Table (order_details_id, order_id, product_id, quantity, unit_price, line_total):

(1, 1, 1, 4, 149.99, 599.96),
(2, 1, 2, 1, 1299.99, 1299.99),
(3, 2, 2, 1, 1299.99, 1299.99),
(4, 3, 3, 1, 3699.99, 3699.99)