Project Part 2:
Database Implementation

The database application you are creating will be used by Pizzas-R-Us to track the day-to-day operations of their pizzeria. For this assignment you will use the descriptions and ERD below to create and populate the database needed by Pizzas-R-Us.

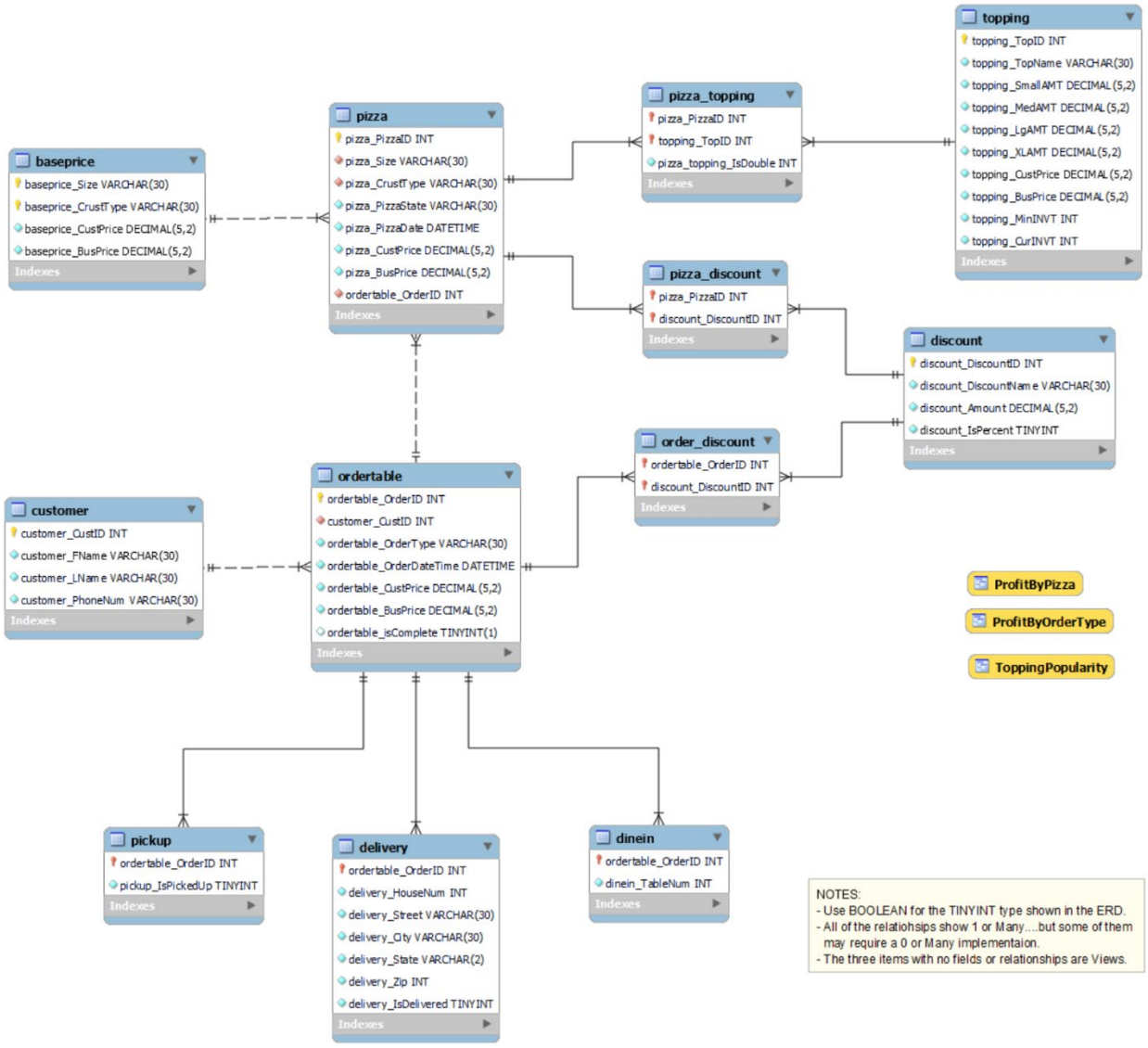
Pizzas-R-Us:
The most obvious thing that needs to be tracked as part of this database system is the information about each pizza. A pizza has a crust type (thin, original, pan, gluten free) and a size (small, medium, large, x-large). A pizza has an associated price and cost to the company, both of which are determined by the size of the pizza and the toppings on the pizza. A pizza can be in two states: completed by the kitchen or still being processed by the kitchen. Each pizza can have multiple toppings. Each topping has a name, a price to the customer, a price to the business, an amount used for each pizza size, a minimum inventory level, and a current inventory level (which is updated whenever a pizza is ordered). The same topping can be on many pizzas (i.e., several pizzas can have pepperoni on them). A customer can request extra of any topping, which is always a double amount. Cheese counts as a topping (there is no free cheese in this organization).

Pizzas belong to orders. An order can be for dine in, pickup, or delivery. An order can have multiple pizzas on it. An order can be marked as complete once all its pizzas are complete. Each order has a total cost to the business, which is calculated by adding up the costs of each pizza. Each order should have a timestamp for when the food was ordered (so the kitchen can prioritize orders). Each order also has a total price to the customer, which is calculated by adding the prices of each pizza. If an order is for a dine in customer, then we need to know the table number. If an order is for pickup, then it needs to have a pickup customer associated with it. That customer must have a name and a phone number. If an order is for delivery, then it must have a delivery customer associated with it and include a name, phone number and address. A customer can have many orders, since the information is saved for the next time they order pizza. A customer could have some pickup orders, and some delivery orders. While other pizza places might allow a customer to save multiple addresses, Pizzas-R-Us only allows a customer to have one address. Note, we don't need any customer information for a dine in customer. When designing your database, the type of order should be used as your discriminator.

Furthermore, Pizzas-R-Us offers discounts. Discounts can be applied to individual pizzas or an entire order; although you can't apply to same discount to both a pizza and an order. Discounts have a name and either a dollar amount off or a percentage off. A pizza or order can have multiple discounts applied to it, and a discount can be applied to many pizzas or orders. Order discounts are applied to the entire order after all the pizza discounts have been applied.

The pizzeria also needs to track the base prices for their pizzas. Each pizza needs a base price (to the customer) and a base cost (to the business) based on the crust type and pizza size. To compute the price of a pizza, you would look at the size and crust of the pizza and find the corresponding base price. To that you would add the price for each topping on the pizza (accounting for double topping quantities). Finally, you would apply any discounts to the pizza. To find the total for the order, you would add up the price for each pizza, then apply any discounts that apply to the order. While the base prices and topping prices will change over time, those changes should not be reflected in past orders. So, a pizza's price should be calculated once and saved. To find the cost of a pizza to the business, the same process is used, with base cost instead of base price. Discounts do not lower the cost of the order to a business.

Based on the information above, a database design team has created the following ERD:



The pizzeria is under new management and will be very closely monitoring profitability. To make this easier to do, you will need to implement three views. Management would like reports (aka views) on:

- Popular Toppings: rank order of all the toppings (accounting for extra toppings) from most popular to least popular
- Profit by Pizza: a summary of the profit by pizza size and crust type over a selected time period ordered by profit from most profitable to least profitable
- Profit by Order Type: a summary of the profit for each of the three types of orders by month with a grand total over all the orders at the pizzeria

The views will be used in Part 3 of the project.

Requirements:

1. Create an SQL script file that has the statements necessary to build the tables and relationships shown in the ERD. You must match the table names, fields and types EXACTLY. This is the database you will use in Part 3. Remember to include any constraints that you need for the database. The script should include the SQL to create a database schema called PizzaDB. **Name this file "CreateTables.sql"**
2. Create an SQL script file that maps your tables onto a set of standard views. You must create these views with the following view names and column names.
    - ToppingPopularity: rank order of all the toppings (accounting for extra toppings) by ToppingCount-from most popular to least popular-and then but Topping alphabetically.

Topping	ToppingCount
Pepperoni	10
Regular Cheese	10
Four Cheese Blend	7
Chicken	3
Mushrooms	3
Banana Peppers	2
Black Olives	2
Green Pepper	2
Ham	2
Onion	2
Pineapple	2
Roma Tomato	2
Sausage	2
Bacon	1
Feta Cheese	1
Goat Cheese	1
Jalapenos	0

    - ProfitByPizza: a summary of the profit by pizza size and crust type each month and ordered by profit from most profitable to least profitable.

Size	Crust	Profit	OrderMonth
Small	Original	5.53	4/2025
Medium	Pan	9.62	4/2025
XLarge	Gluten-Free	20.86	3/2025
Large	Thin	31.25	4/2025
Large	Thin	38.24	3/2025
XLarge	Original	65.20	4/2025
Large	Original	69.48	3/2025

    - ProfitByOrderType: a summary of the profit for each of the three types of orders by month with a grand total over all the orders at the pizzeria ordered by customer type and profit.

customerType	OrderMonth	TotalOrderPrice	TotalOrderCost	Profit
dinein	3/2025	19.75	3.68	16.07
dinein	4/2025	19.78	4.63	15.15
pickup	3/2025	117.98	27.64	90.34
delivery	4/2025	100.61	26.99	73.62
delivery	3/2025	25.81	3.64	22.17
	Grand Total	283.93	66.58	217.35

DO NOT Change the view names or the column names shown above. **Name this file "CreateViews.sql".** If you've populated your tables correctly, your view output will match the output shown!

3. IF you plan to use stored procedures, triggers or functions to help populate the data in your tables, you must put this code along with the appropriate DELIMETER statements into a separate file named "**CreateSPs.sql**". The code in **CreateSPs** will execute after the **CreateTables** and **CreateViews** scripts are executed, but before the **PopulateData** script This ensures that your stored procedures and triggers are available to use in the **PopulateData** script file to initialize your tables with data. **CreateSPs.sql** should be empty (but present) if you are not using any stored SQL code.

   SPECIAL NOTE: Graduate and Honors students, you must define AND use:
   - ***2 Stored Procedures***
   - ***2 Stored Functions***
   - ***2 Update Triggers (on 2 different tables)***
   - ***2 Insert Triggers (on 2 different tables)***

   This code must be active AND used to receive full credit. REMEMBER, your stored procedures and triggers will execute while the PopulateData script is running, so they should do something useful related to entering data!

4. Create an SQL script that has the statements necessary to populate your database with the starter data provided below. **Name this file "PopulateData.sql"**
5. Create an SQL script that will DROP each table and view in your database. This will be helpful when there are errors in your create or populate scripts and you need to reset the entire database. **Name this file "DropTables.sql"**

The autograder will execute your SQL in order: CreateTables, then CreateViews, then CreateSPs, then PopulateData, and finally DropTables. Be aware that the autograder will be running on an Ubuntu virtual machine, meaning that the database names ARE case sensitive! Which also means the case of the file names MUST be exactly as shown above.

Notes:

As discussed in class, MySQL has an autoincrement feature; it will be needed for select Primary Key fields where there is not a natural primary key in the data. Remember, any autoincrement fields should start at 1 and increment by 1. When you do this, you will not know what the key value is and won't be able to reference it in later insert statements in a foreign key field. There are ways around this such as selecting the key data from the primary key table as part of your insert into the FK. Make sure that all of your script files contain the appropriate code to select the correct schema.

Help:

Do not wait until the last day to ask questions or get started!

Submission

You will submit your assignment via Gradescope. You must zip the 5 files (**CreateTables.sql, CreateViews.sql , CreateSPs, PopulateData.sql, and DropTables.sql)** together and upload them via Gradescope. When submitting to Gradescope you must indicate who your partner is (if you have one). Only 1 partner needs to make the submission. **At the top of each .sql file should be a comment with the name(s) of the individual(s) that developed the script**. If you do not use the specified file names, the autograder will not run and your submission will receive a 0! MAKE SURE YOUR SQL RUNS before submitting it!

CHEATING:

Don't do it! The only way to learn the material is to do your own work…so DO IT! You are expected to do your own work, possibly in collaboration with ONE partner. Plagiarism detection will be run on the submitted solutions.

Groups:

You may work as an individual, or work with **one partner** for this project. I encourage you to work with a partner. **Please note that once you have selected a partner, you are only allowed to work with that partner for the rest of the semester long project.** If you work with someone and decide you do not want to work with them on a later stage, you each have to complete the project on your own. So, pick your partner carefully. If you work with a partner, **actually work with them**.

You may only work with one partner. Any larger groups would be violating the academic integrity rules for this class. Any groups that work together would also violate the academic integrity rules. For this assignment, there is not much that can be discussed without violating academic integrity, so it is best not to discuss it with anyone other than your partner. If working with a partner, make sure select your partner when submitting to Gradescope and both partner's names must be included in all the SQL files.

Starter Data:

Initialize your database with the following data:

Toppings:

Toppings					Units Used			
Name	Price per unit	Cost per unit	Inventory	Minimum	Small	Medium	Large	XLarge
Pepperoni	1.25	0.2	100	50	2	2.75	3.5	4.5
Sausage	1.25	0.15	100	50	2.5	3	3.5	4.25
Ham	1.5	0.15	78	25	2	2.5	3.25	4
Chicken	1.75	0.25	56	25	1.5	2	2.25	3
Green Pepper	0.5	0.02	79	25	1	1.5	2	2.5
Onion	0.5	0.02	85	25	1	1.5	2	2.75
Roma Tomato	0.75	0.03	86	10	2	3	3.5	4.5
Mushrooms	0.75	0.1	52	50	1.5	2	2.5	3
Black Olives	0.6	0.1	39	25	0.75	1	1.5	2
Pineapple	1	0.25	15	0	1	1.25	1.75	2
Jalapenos	0.5	0.05	64	0	0.5	0.75	1.25	1.75
Banana Peppers	0.5	0.05	36	0	0.6	1	1.3	1.75
Regular Cheese	0.5	0.12	250	50	2	3.5	5	7
Four Cheese Blend	1	0.15	150	25	2	3.5	5	7
Feta Cheese	1.5	0.18	75	0	1.75	3	4	5.5
Goat Cheese	1.5	0.2	54	0	1.6	2.75	4	5.5
Bacon	1.5	0.25	89	0	1	1.5	2	3

Discounts:

Name	% off	$ off
Employee	15%	
Lunch Special Medium		$1.00
Lunch Special Large		$2.00
Specialty Pizza		$1.50
Happy Hour	10%	
Gameday Special	20%	

Base Prices

Size	Crust	Price	Cost
Small	Thin	3	0.5
Small	Original	3	0.75
Small	Pan	3.5	1
Small	Gluten-Free	4	2
Medium	Thin	5	1
Medium	Original	5	1.5
Medium	Pan	6	2.25
Medium	Gluten-Free	6.25	3
Large	Thin	8	1.25
Large	Original	8	2
Large	Pan	9	3
Large	Gluten-Free	9.5	4
XLarge	Thin	10	2
XLarge	Original	10	3
XLarge	Pan	11.5	4.5
XLarge	Gluten-Free	12.5	6

Calculating prices:
If a discount is applied to a pizza that discount reduces the cost to the customer but does not impact the cost to the business. Once all the pizza prices are calculated and discounts applied, the pizza prices can be rolled up to calculate the order price and costs and any order discounts are then applied to the entire order. When multiple discounts are applied, make sure you apply the $ discounts before applying the % discounts. The same discount can be applied to multiple pizzas and likewise the same discounts can also be applied to the order!

Orders:
Enter the following orders into the database. You do not need to deduct from the inventory for these orders (it has already been taken into account). The customer price and business cost of each pizza (after discounts) is included for reference. All orders should be marked as completed, delivery orders should be marked as delivered, and pickup orders should be marked as picked up. **All dates are for the current calendar year.**

#1. On March 5th at 12:03 pm there was a dine-in order (at table 21) for a large thin crust pizza with Regular Cheese (extra), Pepperoni, and Sausage (Price: $19.75, Cost: $3.68). They used the "Lunch Special Large" discount for the pizza.

#2. On April 3rd at 12:05 pm there was a dine-in order (at table 4). They ordered a medium pan pizza with Feta Cheese, Black Olives, Roma Tomatoes, Mushrooms and Banana Peppers (Price: $12.85, Cost: $3.23). They used the "Lunch Special Medium" and the "Specialty Pizza" discount for the pizza. They also ordered a small original crust pizza with Regular Cheese, Chicken and Banana Peppers (Price: $6.93, Cost: $1.40), no discounts on this pizza.

#3. On March 3rd at 9:30 pm Andrew Wilkes-Krier placed an order for pickup of 6 large original crust pizzas with Regular Cheese and Pepperoni (Price: $14.88, Cost: $3.30 each). Andrew's phone number is 864-254-5861.

#4. On April 20th at 7:11 pm there was a delivery order made by Andrew Wilkes-Krier for 1 xlarge pepperoni and Sausage pizza (Price: $27.94, Cost: $9.19), one xlarge pizza with Ham (extra) and Pineapple (extra) pizza (Price: $31.50, Cost: $6.25), and one xlarge Chicken and Bacon pizza (Price: $26.75, Cost: $5.55). All the pizzas have the Four Cheese Blend on it and are original crust. The order has the "Gameday Special" discount applied to it, and the ham and pineapple pizza has the "Specialty Pizza" discount applied to it. The pizzas were delivered to 115 Party Blvd, Anderson SC 29621. His phone number is the same as before.

#5. On March 2nd at 5:30 pm Matt Engers placed an order for pickup for an xlarge pizza with Green Pepper, Onion, Roma Tomatoes, Mushrooms, and Black Olives on it. He wants the Goat Cheese on it, and a Gluten Free Crust (Price: $28.70, Cost: $7.84). The "Specialty Pizza" discount is applied to the pizza. Matt's phone number is 864-474-9953.

#6. On March 2nd at 6:17 pm Frank Turner places an order for delivery of one large pizza with Chicken, Green Peppers, Onions, and Mushrooms. He wants the Four Cheese Blend (extra) and thin crust (Price: $25.81, Cost: $3.64). The pizza was delivered to 6745 Wessex St Anderson SC 29621. Frank's phone number is 864-232-8944.

#7. On April 13th at 8:32 pm Milo Auckerman ordered two large thin crust pizzas. One had the Four Cheese Blend on it (extra) (Price: $18.00, Cost: $2.75), the other was Regular Cheese and Pepperoni (extra) (Price: $19.25, Cost: $3.25). He used the "Employee" discount on his order. He had them delivered to 8879 Suburban Home, Anderson, SC 29621. His phone number is 864-878-5679.