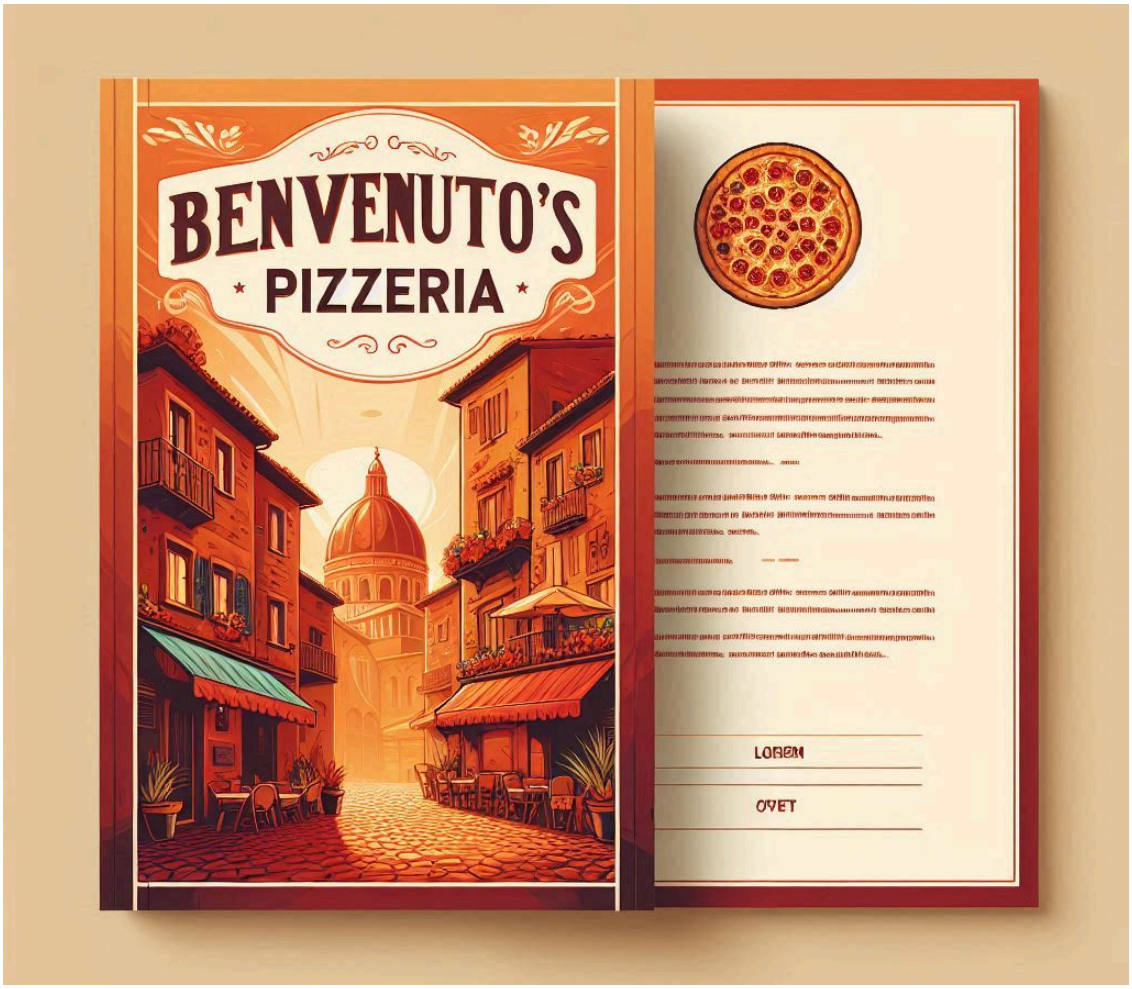
Benvenuto's Pizzeria Database Management Project



Matthew Pizzitola

mattmp.04@gmail.com

+1 (647) 984-0504

6 August 2024

Table of Contents

Introduction

This project focuses on creating a comprehensive SQL database management system to enhance my Information Technology portfolio and further develop my IT skills. The project is centred around designing and implementing a relational database and crafting custom SQL queries, which are critical components for managing data in real-world scenarios. Specifically, the database will handle essential functions such as customer orders, inventory management, and staff scheduling for a hypothetical pizzeria.

The key objectives of this project are:

1. **Design and Build a SQL Database:** Develop a custom relational database to store and organize crucial business data, covering areas like customer orders, stock levels, and staff management.

2. **Write Custom SQL Queries:** Hone my skills in data extraction, manipulation, and analysis through the creation of custom SQL queries, thereby enhancing my problem-solving abilities and proficiency in data handling.

This project is strategically structured to demonstrate essential skills required for a data analyst role. By applying these techniques in a practical context, I aim to showcase my technical expertise and readiness to address real-world data challenges.

PART 1 - Designing a Relational Database

1. Understanding the Business Requirements

Benvenuto's Pizzeria operates as a dine-in restaurant, and the primary objective of the database is to efficiently store and manage data to monitor and enhance business performance. The key focus areas are:

- **Orders:** Tracking all sales transactions.
- **Stock Control:** Managing inventory of ingredients and supplies.
- **Staff:** Managing employee information.

The initial step in designing the database involves identifying and organizing the necessary data fields into tables. This process includes normalizing the data to minimize redundancy and defining relationships between tables to ensure efficient data management.

2. Defining the Database Schema

The schema will be designed based on the following entities and attributes:

Order Data Requirements:

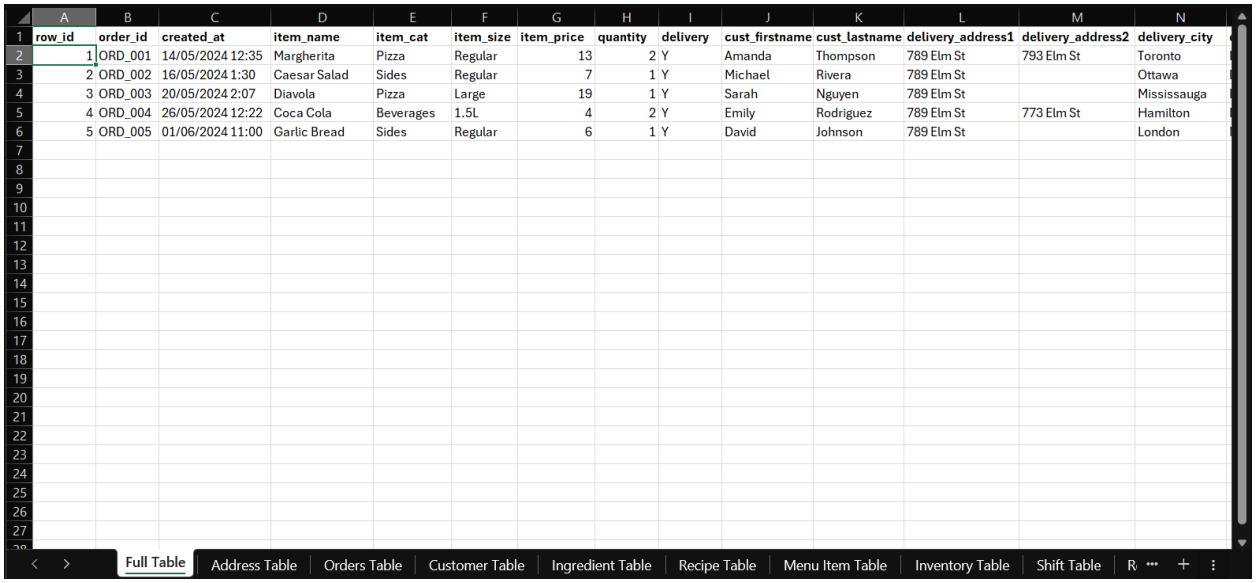
Ben, the owner, has provided a list of the data he wants to collect for each order:

- Item name
- Item price
- Quantity
- Customer name
- Delivery address ← (to be split into separate components)

This initial data will be expanded with additional fields to ensure comprehensive data collection.

Our Additional Data Fields:

- Row ID
- Order ID
- Item name
- Item category
- Item size
- Item price
- Quantity
- Customer first name
- Customer last name
- Delivery address 1
- Delivery address 2
- Delivery city
- Delivery zip code



row_id	order_id	created_at	item_name	item_cat	item_size	item_price	quantity	delivery	cust_firstname	cust_lastname	delivery_address1	delivery_address2	delivery_city
1	ORD_001	14/05/2024 12:35	Margherita	Pizza	Regular	13	2	Y	Amanda	Thompson	789 Elm St	793 Elm St	Toronto
2	ORD_002	16/05/2024 1:30	Caesar Salad	Sides	Regular	7	1	Y	Michael	Rivera	789 Elm St		Ottawa
3	ORD_003	20/05/2024 2:07	Diavola	Pizza	Large	19	1	Y	Sarah	Nguyen	789 Elm St		Mississauga
4	ORD_004	26/05/2024 12:22	Coca Cola	Beverages	1.5L	4	2	Y	Emily	Rodriguez	789 Elm St	773 Elm St	Hamilton
5	ORD_005	01/06/2024 11:00	Garlic Bread	Sides	Regular	6	1	Y	David	Johnson	789 Elm St		London

Full Table | Address Table | Orders Table | Customer Table | Ingredient Table | Recipe Table | Menu Item Table | Inventory Table | Shift Table

Here's a mock-up of what the table could potentially look like in Excel.

3. Benvenuto's Pizzeria Menu:

The menu provides crucial information to define product-related fields in the database, such as product size and category. By analyzing the menu, we can better organize the data and prepare it for inclusion in the database.

Special Pizza

- **Margherita:** Tomato sauce, mozzarella, oregano. Reg: $13, Large: $15
- **Diavola:** Mozzarella, spicy salami, chili pepper. Reg: $16, Large: $19
- **Parmigiana:** Mozzarella, eggplants, parmesan cheese. Reg: $15, Large: $18
- **Quattro formaggi:** Mozzarella, gorgonzola, ricotta, and parmesan. Reg: $16, Large: $19

Signature Pizza

- **Napolitana:** Mozzarella, anchovies, capers. Reg: $16, Large: $18
- **Pepperoni:** Mozzarella, pepperoni. Reg: $15, Large: $17
- **Seafood:** Mozzarella, shrimps, tuna, calamari. Reg: $17, Large: $20
- **Hawaiian:** Mozzarella, ham, pineapple. Reg: $15, Large: $17

Sides

- **Garlic Bread:** Ciabatta with garlic butter and parsley. $6
- **Chicken Wings:** Extra juicy with a healthy crunch. $7
- **Breadsticks:** Your favourite Italian classic side. $5
- **Caesar Salad:** Chicken, croutons, lettuce, Caesar dressing. $7

Desserts

- **Ice cream:** Vanilla, chocolate, strawberry or pistachio. $6
- **Chocolate Brownie:** Rectangular chocolate baked perfection. $5
- **Banoffee Pie:** Bananas, cream, caramel sauce on a biscuit base. $7
- **Fruit Salad:** Fresh fruit served in our special homemade syrup. $5

Beverages

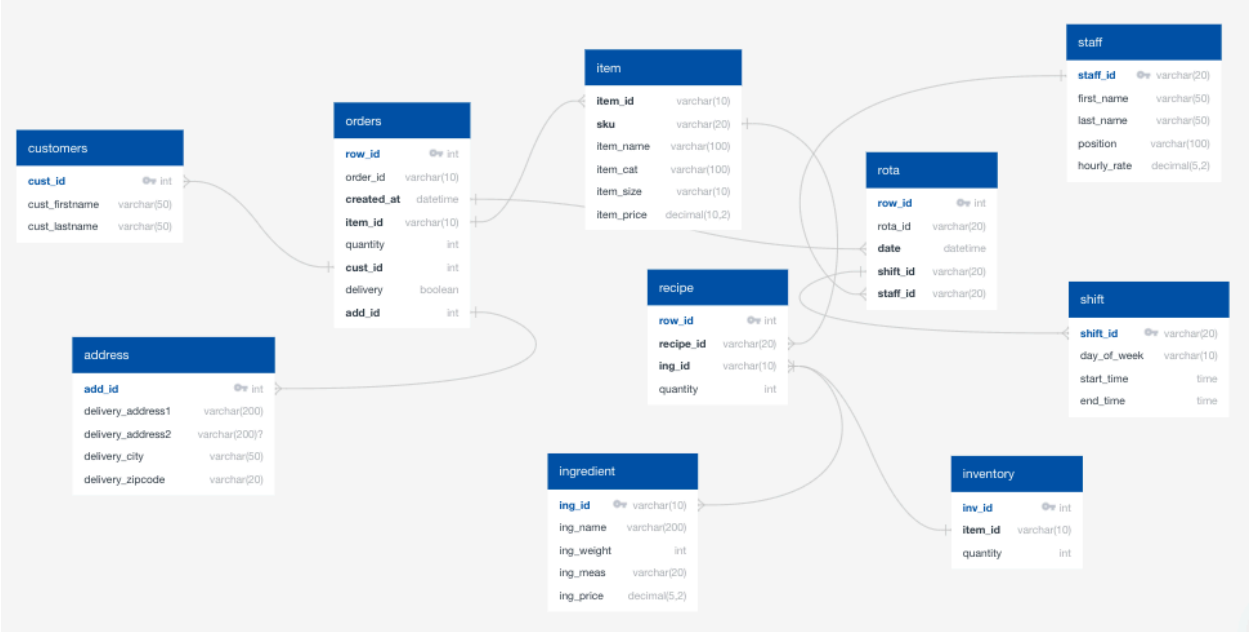
- **Coca-Cola:** Regular or diet. 33cl: $3, 1.5L: $6
- **7 Up: Regular or diet. 33cl:** $3, 1.5L: $6
- **Fanta: Regular or diet. 33cl:** $3, 1.5L: $6
- **Sparkling Water:** San Pellegrino, Perrier. 33cl: $2, 1.5L: $4

By analyzing Ben's provided menu, we can start to see potential fields that must be created to organize the data better. Examples of these fields include product size and category. With this, we can head to Microsoft Excel (Google Sheets/Apple Numbers) to create a mock-up of what the table could potentially look like.

4. Utilizing QuickDBD for Quick and Easy Database Creation

QuickDBD, an online database diagramming tool, will be used to create the layout for the database tables and fields. This diagram will visually represent the database structure.

5. Benvenuto's Pizzeria QuickDBD Layout



Data normalization will be applied to reduce redundancy and improve data efficiency. This involves creating additional tables for customer data and addresses, linking them to the main table using unique identifiers. We can start by normalizing the existing table through the creation of two additional tables—one for customer data and another for customer addresses. After creating these tables, it's essential to define the relationships between them, which can be done in QuickDBD by linking the table IDs to the main table. Additionally, we should create an item table, which serves two key purposes: reducing the data load on the main table and providing a convenient way for Ben to update or adjust item details, such as names or prices.

6. Stock Control Requirements

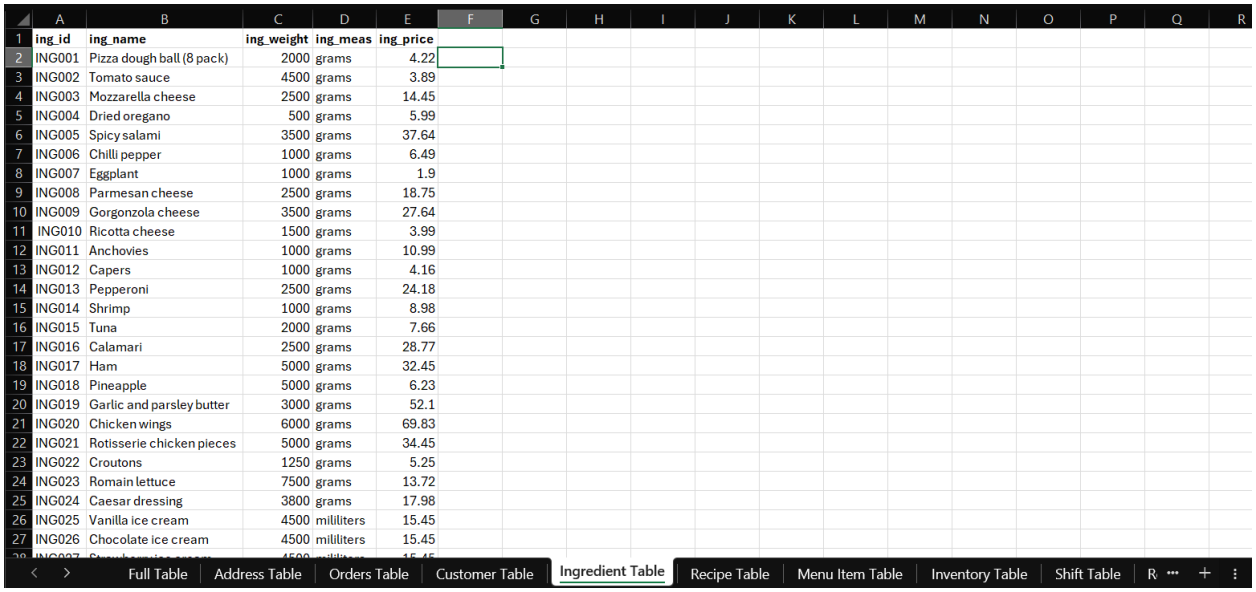
These are the desired stock requirements that Ben has provided for us:

- Wants to be able to know when it's time to order new stock
- To do this we're going to need more information about:
    - What ingredients go into each pizza
    - Their quantity is based on the size of the pizza
    - The existing stock level
- We'll assume the lead time for delivery by suppliers is the same for all ingredients

With this, we can create a recipe table and an ingredient table that will help Ben calculate the exact price of a pizza. Data normalization comes into play here as if something like prices increase for the ingredients all Ben would have to do is update the content of the ingredient table. An inventory table will also help hold the stock levels for each ingredient.

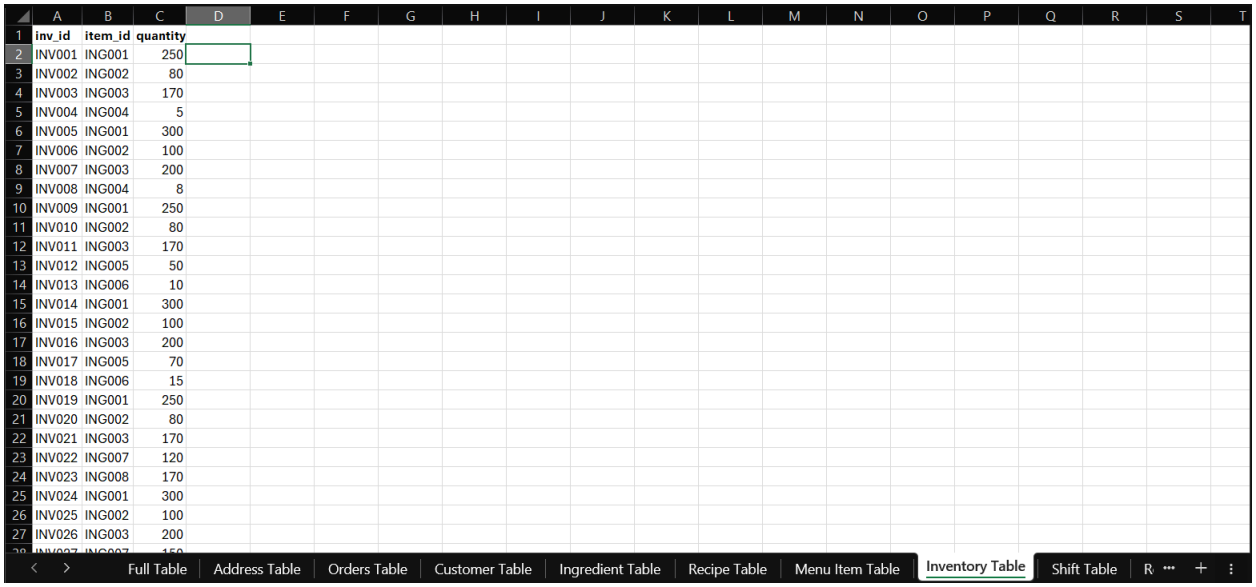
row_id	recipe_id	ingred_id	quantity
1	PIZZ-MARG-R	ING001	250
2	PIZZ-MARG-R	ING002	80
3	PIZZ-MARG-R	ING003	170
4	PIZZ-MARG-R	ING004	5
5	PIZZ-MARG-L	ING001	300
6	PIZZ-MARG-L	ING002	100
7	PIZZ-MARG-L	ING003	200
8	PIZZ-MARG-L	ING004	8
9	PIZZ-DIAV-R	ING001	250
10	PIZZ-DIAV-R	ING002	80
11	PIZZ-DIAV-R	ING003	170
12	PIZZ-DIAV-R	ING005	50
13	PIZZ-DIAV-R	ING006	10
14	PIZZ-DIAV-L	ING001	300
15	PIZZ-DIAV-L	ING002	100
16	PIZZ-DIAV-L	ING003	200
17	PIZZ-DIAV-L	ING005	70
18	PIZZ-DIAV-L	ING006	15
19	PIZZ-PARM-R	ING001	250
20	PIZZ-PARM-R	ING002	80
21	PIZZ-PARM-R	ING003	170
22	PIZZ-PARM-R	ING007	120
23	PIZZ-PARM-R	ING008	170
24	PIZZ-PARM-L	ING001	300
25	PIZZ-PARM-L	ING002	100
26	PIZZ-PARM-L	ING003	200

The recipe table links ingredients to specific pizzas, allowing for accurate cost calculation.



ing_id	ing_name	ing_weight	ing_meas	ing_price
ING001	Pizza dough ball (8 pack)	2000	grams	4.22
ING002	Tomato sauce	4500	grams	3.89
ING003	Mozzarella cheese	2500	grams	14.45
ING004	Dried oregano	500	grams	5.99
ING005	Spicy salami	3500	grams	37.64
ING006	Chilli pepper	1000	grams	6.49
ING007	Eggplant	1000	grams	1.9
ING008	Parmesan cheese	2500	grams	18.75
ING009	Gorgonzola cheese	3500	grams	27.64
ING010	Ricotta cheese	1500	grams	3.99
ING011	Anchovies	1000	grams	10.99
ING012	Capers	1000	grams	4.16
ING013	Pepperoni	2500	grams	24.18
ING014	Shrimp	1000	grams	8.98
ING015	Tuna	2000	grams	7.66
ING016	Calamari	2500	grams	28.77
ING017	Ham	5000	grams	32.45
ING018	Pineapple	5000	grams	6.23
ING019	Garlic and parsley butter	3000	grams	52.1
ING020	Chicken wings	6000	grams	69.83
ING021	Rotisserie chicken pieces	5000	grams	34.45
ING022	Croutons	1250	grams	5.25
ING023	Romain lettuce	7500	grams	13.72
ING024	Caesar dressing	3800	grams	17.98
ING025	Vanilla ice cream	4500	mililiters	15.45
ING026	Chocolate ice cream	4500	mililiters	15.45

The ingredient table tracks stock levels and updates ingredient prices as needed.



inv_id	item_id	quantity
INV001	ING001	250
INV002	ING002	80
INV003	ING003	170
INV004	ING004	5
INV005	ING001	300
INV006	ING002	100
INV007	ING003	200
INV008	ING004	8
INV009	ING001	250
INV010	ING002	80
INV011	ING003	170
INV012	ING005	50
INV013	ING006	10
INV014	ING001	300
INV015	ING002	100
INV016	ING003	200
INV017	ING005	70
INV018	ING006	15
INV019	ING001	250
INV020	ING002	80
INV021	ING003	170
INV022	ING007	120
INV023	ING008	170
INV024	ING001	300
INV025	ING002	100
INV026	ING003	200

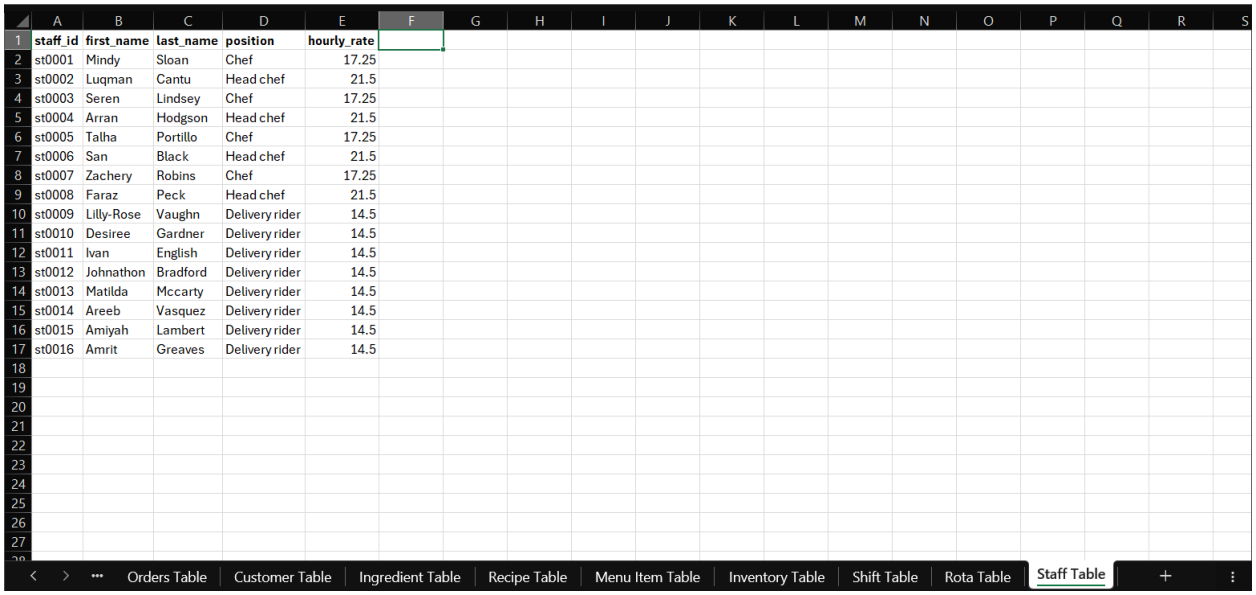
The inventory table to monitor stock quantities and reorder supplies as necessary.

Staff Data Requirements

These are the desired staff requirements that Ben has provided for us:

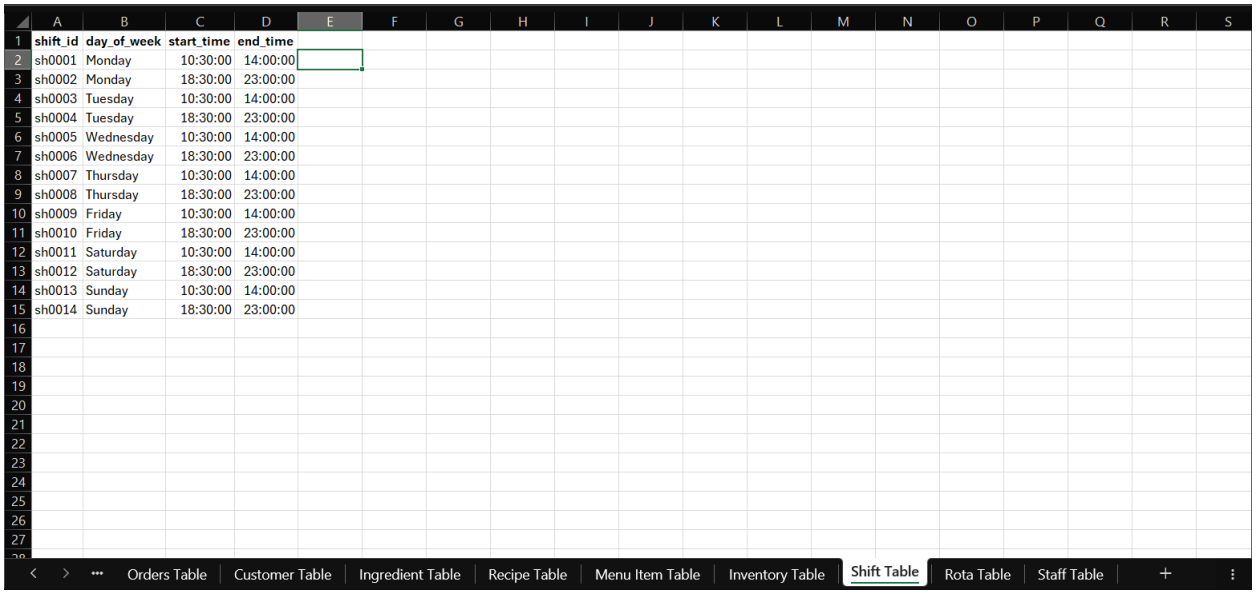
- Wants to know which staff members are working when
- Based on the staff salary information, how much does each pizza cost (ingredients+chefs+delivery)

The first table that should be created is the staff table to track current staff members and their positions and hourly payment rates. Another table that we should create is the shift table, this would contain the employee's start day and time they work. The rota table is the final table we need to create and it displays who is working when. All these tables possess elements that link to each other to help normalize and streamline data across the database.



staff_id	first_name	last_name	position	hourly_rate
st0001	Mindy	Sloan	Chef	17.25
st0002	Luqman	Cantu	Head chef	21.5
st0003	Seren	Lindsey	Chef	17.25
st0004	Arran	Hodgson	Head chef	21.5
st0005	Talha	Portillo	Chef	17.25
st0006	San	Black	Head chef	21.5
st0007	Zachery	Robins	Chef	17.25
st0008	Faraz	Peck	Head chef	21.5
st0009	Lilly-Rose	Vaughn	Delivery rider	14.5
st0010	Desiree	Gardner	Delivery rider	14.5
st0011	Ivan	English	Delivery rider	14.5
st0012	Johnathon	Bradford	Delivery rider	14.5
st0013	Matilda	Mccarty	Delivery rider	14.5
st0014	Areeb	Vasquez	Delivery rider	14.5
st0015	Amiyah	Lambert	Delivery rider	14.5
st0016	Amrit	Greaves	Delivery rider	14.5

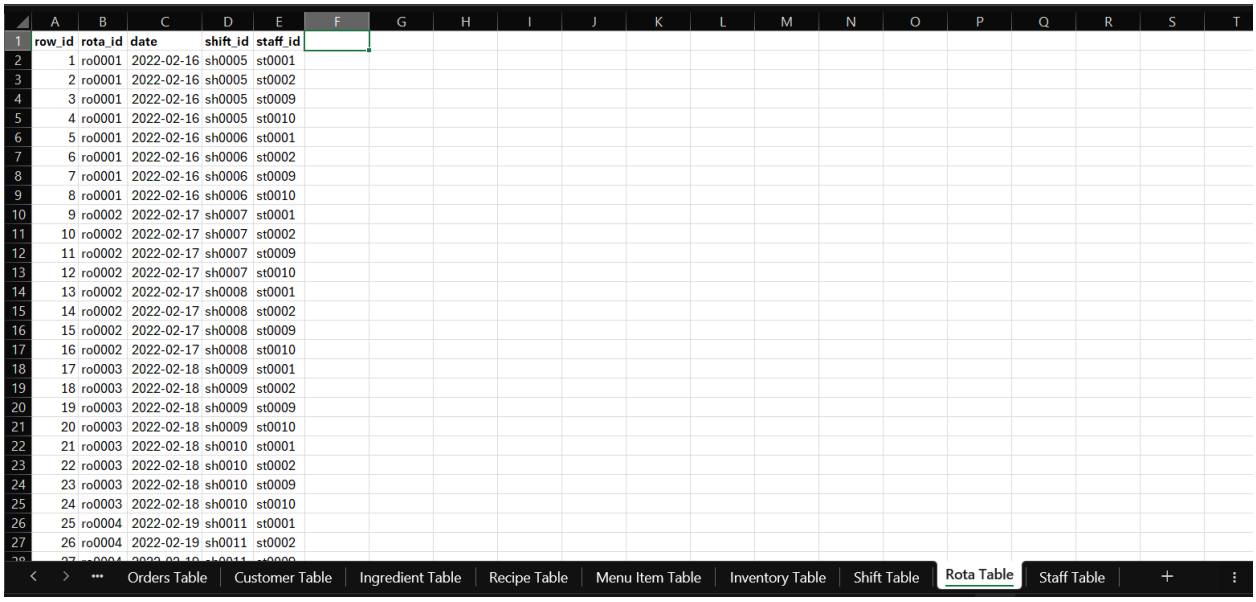
A staff table to store employee details, including positions and hourly rates.



shift_id	day_of_week	start_time	end_time
sh0001	Monday	10:30:00	14:00:00
sh0002	Monday	18:30:00	23:00:00
sh0003	Tuesday	10:30:00	14:00:00
sh0004	Tuesday	18:30:00	23:00:00
sh0005	Wednesday	10:30:00	14:00:00
sh0006	Wednesday	18:30:00	23:00:00
sh0007	Thursday	10:30:00	14:00:00
sh0008	Thursday	18:30:00	23:00:00
sh0009	Friday	10:30:00	14:00:00
sh0010	Friday	18:30:00	23:00:00
sh0011	Saturday	10:30:00	14:00:00
sh0012	Saturday	18:30:00	23:00:00
sh0013	Sunday	10:30:00	14:00:00
sh0014	Sunday	18:30:00	23:00:00

Orders Table | Customer Table | Ingredient Table | Recipe Table | Menu Item Table | Inventory Table | Shift Table | Rota Table | Staff Table

A shift table to record work schedules.



row_id	rota_id	date	shift_id	staff_id
1	ro0001	2022-02-16	sh0005	st0001
2	ro0001	2022-02-16	sh0005	st0002
3	ro0001	2022-02-16	sh0005	st0009
4	ro0001	2022-02-16	sh0005	st0010
5	ro0001	2022-02-16	sh0006	st0001
6	ro0001	2022-02-16	sh0006	st0002
7	ro0001	2022-02-16	sh0006	st0009
8	ro0001	2022-02-16	sh0006	st0010
9	ro0002	2022-02-17	sh0007	st0001
10	ro0002	2022-02-17	sh0007	st0002
11	ro0002	2022-02-17	sh0007	st0009
12	ro0002	2022-02-17	sh0007	st0010
13	ro0002	2022-02-17	sh0008	st0001
14	ro0002	2022-02-17	sh0008	st0002
15	ro0002	2022-02-17	sh0008	st0009
16	ro0002	2022-02-17	sh0008	st0010
17	ro0003	2022-02-18	sh0009	st0001
18	ro0003	2022-02-18	sh0009	st0002
19	ro0003	2022-02-18	sh0009	st0009
20	ro0003	2022-02-18	sh0009	st0010
21	ro0003	2022-02-18	sh0010	st0001
22	ro0003	2022-02-18	sh0010	st0002
23	ro0003	2022-02-18	sh0010	st0009
24	ro0003	2022-02-18	sh0010	st0010
25	ro0004	2022-02-19	sh0011	st0001
26	ro0004	2022-02-19	sh0011	st0002

Orders Table | Customer Table | Ingredient Table | Recipe Table | Menu Item Table | Inventory Table | Shift Table | Rota Table | Staff Table

A rota table to organize and track employee shifts.

Creating the Database

With the schema and tables defined, QuickDBD will generate MySQL code, which will be imported into MySQL Workbench to create the database. The corresponding Excel sheets will be used to populate the database, and custom SQL queries will be coded to

extract and analyze data, supporting the overall business performance monitoring objectives of Benvenuto's Pizzeria.

PART 2 - Writing Custom SQL Queries

Orders Data

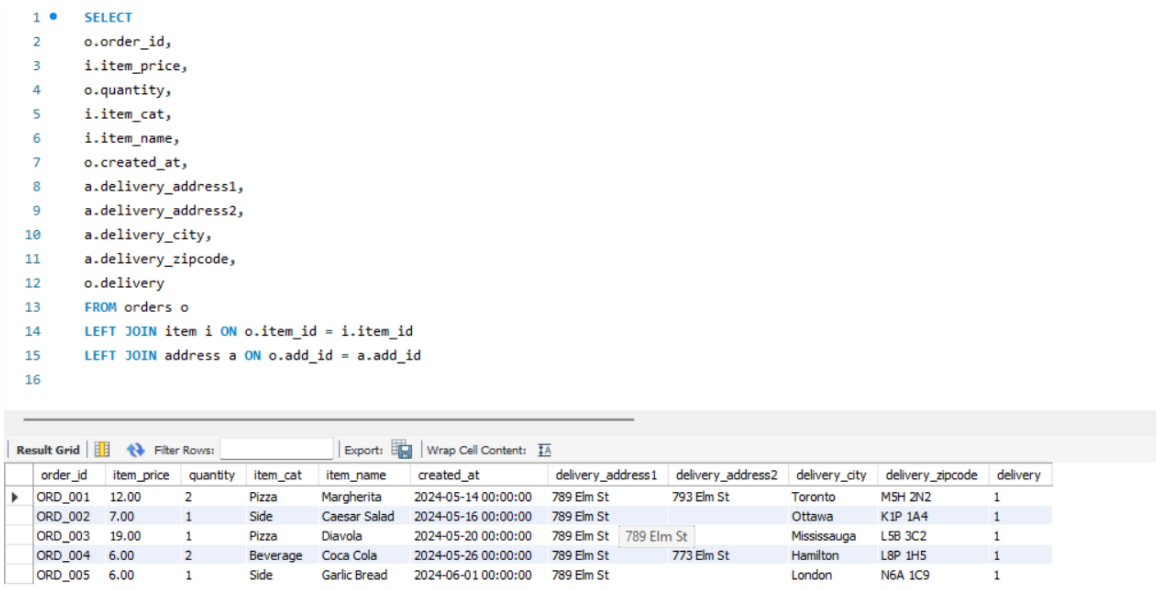
This section analyzes the order data stored across multiple tables, including the **‘item’**, **‘orders’**, **‘customer’**, and **‘delivery_address’** tables. Our goal is to extract actionable insights that will inform the creation of a comprehensive dashboard displaying key performance metrics such as:

1. **Total Orders**
2. **Total Sales**
3. **Total Items Sold**
4. **Average Order Value**
5. **Sales by Category**
6. **Top Selling Items**
7. **Orders by Hour**
8. **Sales by Hour**
9. **Orders by Address**
10. **Orders by Delivery Method (Delivery/Pick-Up)**

To begin, we import the QuickDBD schema into MySQL Workbench to establish the database structure. We then populate the database by importing sample data into each table via CSV files, setting the stage for querying.

The next step involves constructing SQL queries that join these tables to retrieve the necessary data for our dashboard. Here's an example query that demonstrates how we extract relevant information by joining the **'orders'**, **'item'**, and **'address'** tables:

```
SELECT
    o.order_id,
    i.item_price,
    o.quantity,
    i.item_cat,
    i.item_name,
    o.created_at,
    a.delivery_address1,
    a.delivery_address2,
    a.delivery_city,
    a.delivery_zipcode,
    o.delivery
FROM orders o
LEFT JOIN item i ON o.item_id = i.item_id
LEFT JOIN address a ON o.add_id = a.add_id;
```



```
1  SELECT
2  o.order_id,
3  i.item_price,
4  o.quantity,
5  i.item_cat,
6  i.item_name,
7  o.created_at,
8  a.delivery_address1,
9  a.delivery_address2,
10 a.delivery_city,
11 a.delivery_zipcode,
12 o.delivery
13 FROM orders o
14 LEFT JOIN item i ON o.item_id = i.item_id
15 LEFT JOIN address a ON o.add_id = a.add_id
16
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

order_id	item_price	quantity	item_cat	item_name	created_at	delivery_address1	delivery_address2	delivery_city	delivery_zipcode	delivery
ORD_001	12.00	2	Pizza	Margherita	2024-05-14 00:00:00	789 Elm St	793 Elm St	Toronto	M5H 2N2	1
ORD_002	7.00	1	Side	Caesar Salad	2024-05-16 00:00:00	789 Elm St		Ottawa	K1P 1A4	1
ORD_003	19.00	1	Pizza	Diavola	2024-05-20 00:00:00	789 Elm St	789 Elm St	Mississauga	L5B 3C2	1
ORD_004	6.00	2	Beverage	Coca Cola	2024-05-26 00:00:00	789 Elm St	773 Elm St	Hamilton	L8P 1H5	1
ORD_005	6.00	1	Side	Garlic Bread	2024-06-01 00:00:00	789 Elm St		London	N6A 1C9	1

MySQL Workbench results.

Explanation:

- **Joins and Selection:** This query retrieves data from the **‘orders’**, **‘item’**, and **‘address’** tables. The **‘LEFT JOIN’** ensures that all records from the **‘orders’** table are included, even if there is no corresponding record in the **‘item’** or **‘address’** tables. This approach maintains a comprehensive dataset that includes all orders, irrespective of item or address details.

- **Columns:** The query selects specific columns relevant to the dashboard, such as order details, item prices, categories, timestamps, and delivery information. By combining data across these tables, this query sets the foundation for calculating essential metrics like total sales and order counts.

Inventory Management

Inventory management requires precise calculations to track ingredient usage and reordering. The inventory management dashboard will provide insights into:

1. Total Quantity by Ingredient
2. Total Cost of Ingredients
3. Calculated Cost of Pizza Production
4. Percentage of Stock Remaining by Ingredient

We accomplish this by creating SQL views that aggregate and calculate the necessary data. Below is an SQL query that generates these metrics:

```
1 •  SELECT
2     O.item_id,
3     i.sku,
4     i.item_name,
5     sum(o.quantity) as order_quantity
6     from orders o
7     left join item i on o.item_id=i.item_id
8     group by o.item_id,i.sku,i.item_name
9
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

item_id	sku	item_name	order_quantity
it001	PIZZ-MARG-R	Margherita	2
it020	SIDE-CAES-R	Caesar Salad	1
it004	PIZZ-DIAV-L	Diavola	1
it029	BEVA-CC-REG1500	Coca Cola	2
it017	SIDE-GARL-R	Garlic Bread	1

MySQL Workbench results for finding the total quantity by ingredients.

```
SELECT
    s1.item_name,
    s1.ing_id,
    s1.ing_name,
    s1.ing_weight,
    s1.ing_price,
    s1.order_quantity,
    s1.recipe_quantity,
    s1.order_quantity * s1.recipe_quantity AS ordered_weight,
    s1.ing_price / s1.ing_weight AS unit_cost,
    (s1.order_quantity * s1.recipe_quantity) * (s1.ing_price /
s1.ing_weight) AS ingredient_cost
FROM (
    SELECT
        o.item_id,
        i.sku,
        i.item_name,
        r.ing_id,
        ing.ing_name,
        r.quantity AS recipe_quantity,
        SUM(o.quantity) AS order_quantity,
        ing.ing_weight,
```

```
        ing.ing_price
    FROM orders o
    LEFT JOIN item i ON o.item_id = i.item_id
    LEFT JOIN recipe r ON i.sku = r.recipe_id
    LEFT JOIN ingredient ing ON ing.ing_id = r.ing_id
    GROUP BY
        o.item_id,
        i.sku,
        i.item_name,
        r.ing_id,
        r.quantity,
        ing.ing_name,
        ing.ing_weight,
        ing.ing_price
) s1;
```

```
select
s1.item_name,
s1.ing_id,
s1.ing_name,
s1.ing_weight,
s1.ing_price,
s1.order_quantity,
s1.recipe_quantity,
s1.order_quantity*s1.recipe_quantity as ordered_weight,
s1.ing_price/s1.ing_weight as unit_cost,
(s1.order_quantity*s1.recipe_quantity)*(s1.ing_price/s1.ing_weight) as ingredient_cost
from (SELECT
o.item_id,
i.sku,
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

item_name	ing_id	ing_name	ing_weight	ing_price	order_quantity	recipe_quantity	ordered_weight	unit_cost	ingredient_cost
Margherita	ING004	Dried oregano	500	5.99	2	5	10	0.011980	0.119800
Margherita	ING003	Mozzarella cheese	2500	14.45	2	170	340	0.005780	1.965200
Margherita	ING002	Tomato sauce	4500	3.89	2	80	160	0.000864	0.138311
Margherita	ING001	Pizza dough ball (8 pack)	2000	4.22	2	250	500	0.002110	1.055000
Caesar Salad	NULL	NULL	NULL	NULL	1	NULL	NULL	NULL	NULL
Diavola	ING006	Chilli pepper	1000	6.49	1	15	15	0.006490	0.097350
Diavola	ING005	Spicy salami	3500	37.64	1	70	70	0.010754	0.752800
Diavola	ING003	Mozzarella cheese	2500	14.45	1	200	200	0.005780	1.156000
Diavola	ING002	Tomato sauce	4500	3.89	1	100	100	0.000864	0.086444
Diavola	ING001	Pizza dough ball (8 pack)	2000	4.22	1	300	300	0.002110	0.633000
Coca Cola	NULL	NULL	NULL	NULL	2	NULL	NULL	NULL	NULL
Garlic Bread	NULL	NULL	NULL	NULL	1	NULL	NULL	NULL	NULL

MySQL Workbench results for finding the total quantity by both cost to make each pizza and ingredients.

Explanation:

- **Aggregation and Calculations:** This query aggregates data at the ingredient level to calculate the total quantity used based on orders and the cost per ingredient. The calculation involves multiplying order quantities by recipe quantities to determine the required weight of each ingredient. The ingredient cost is then computed by dividing the ingredient price by its weight and multiplying it by the quantity needed.

- **Subqueries:** The inner query (**'s1'**) aggregates the data, which is crucial for calculating accurate costs and quantities. This structured approach simplifies the handling of complex aggregations and calculations.

To track remaining inventory and determine which ingredients need to be reordered, we use the following query:

```
SELECT
    s2.ing_name,
    s2.ordered_weight,
    ing.ing_weight * inv.quantity AS total_inv_weight,
    (ing.ing_weight * inv.quantity) - s2.ordered_weight AS
remaining_weight
FROM (
    SELECT
        ing_id,
        ing_name,
        SUM(ordered_weight) AS ordered_weight
    FROM pizzadb.stock1
    GROUP BY ing_name, ing_id
) s2
LEFT JOIN inventory inv ON inv.item_id = s2.ing_id
LEFT JOIN ingredient ing ON ing.ing_id = s2.ing_id;
```

```sql
SELECT
s2.ing_name,
s2.ordered_weight,
ing.ing_weight*inv.quantity as total_inv_weight,
(ing.ing_weight * inv.quantity)-s2.ordered_weight as remaining_weight
FROM (SELECT
ing_id,
ing_name,
sum(ordered_weight) as ordered_weight
FROM
pizzadb.stock1
group by ing_name,ing_id) s2

LEFT JOIN inventory inv ON inv.item_id = s2.ing_id
LEFT JOIN ingredient ing ON ing.ing_id = s2.ing_id
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

ing_name	ordered_weight	total_inv_weight	remaining_weight
Dried oregano	10	4000	3990
Dried oregano	10	2500	2490
Mozzarella cheese	540	500000	499460
Mozzarella cheese	540	425000	424460
Mozzarella cheese	540	500000	499460
Mozzarella cheese	540	425000	424460
Mozzarella cheese	540	500000	499460
Mozzarella cheese	540	425000	424460
Mozzarella cheese	540	500000	499460
Mozzarella cheese	540	425000	424460
Tomato sauce	260	450000	449740

MySQL Workbench results for tracking inventory.

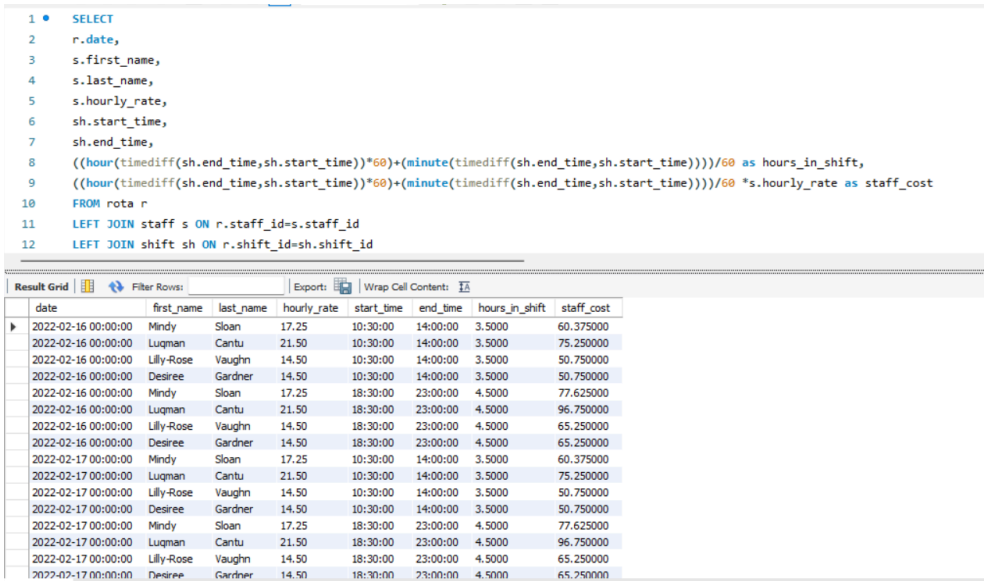
Explanation:

- **Inventory Tracking:** This query calculates the remaining stock of each ingredient by comparing the total available stock (**‘total_inv_weight’**) with the quantity of ingredients used (**‘ordered_weight’**). This provides a clear overview of inventory levels, identifying which ingredients are running low and may require reordering.

Employee Management

Effective employee management involves tracking work hours and calculating labor costs. The following SQL query retrieves the necessary data:

```
SELECT
    r.date,
    s.first_name,
    s.last_name,
    s.hourly_rate,
    sh.start_time,
    sh.end_time,
    ((HOUR(TIMEDIFF(sh.end_time, sh.start_time)) * 60) +
(MINUTE(TIMEDIFF(sh.end_time, sh.start_time)))) / 60 AS
hours_in_shift,
    ((HOUR(TIMEDIFF(sh.end_time, sh.start_time)) * 60) +
(MINUTE(TIMEDIFF(sh.end_time, sh.start_time)))) / 60 * s.hourly_rate
AS staff_cost
FROM rota r
LEFT JOIN staff s ON r.staff_id = s.staff_id
LEFT JOIN shift sh ON r.shift_id = sh.shift_id;
```



```
1  SELECT
2      r.date,
3      s.first_name,
4      s.last_name,
5      s.hourly_rate,
6      sh.start_time,
7      sh.end_time,
8      ((hour(timediff(sh.end_time,sh.start_time))*60)+(minute(timediff(sh.end_time,sh.start_time))))/60 as hours_in_shift,
9      ((hour(timediff(sh.end_time,sh.start_time))*60)+(minute(timediff(sh.end_time,sh.start_time))))/60 *s.hourly_rate as staff_cost
10     FROM rota r
11     LEFT JOIN staff s ON r.staff_id=s.staff_id
12     LEFT JOIN shift sh ON r.shift_id=sh.shift_id
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

date	first_name	last_name	hourly_rate	start_time	end_time	hours_in_shift	staff_cost
2022-02-16 00:00:00	Mindy	Sloan	17.25	10:30:00	14:00:00	3.5000	60.375000
2022-02-16 00:00:00	Luqman	Cantu	21.50	10:30:00	14:00:00	3.5000	75.250000
2022-02-16 00:00:00	Lilly-Rose	Vaughn	14.50	10:30:00	14:00:00	3.5000	50.750000
2022-02-16 00:00:00	Desiree	Gardner	14.50	10:30:00	14:00:00	3.5000	50.750000
2022-02-16 00:00:00	Mindy	Sloan	17.25	18:30:00	23:00:00	4.5000	77.625000
2022-02-16 00:00:00	Luqman	Cantu	21.50	18:30:00	23:00:00	4.5000	96.750000
2022-02-16 00:00:00	Lilly-Rose	Vaughn	14.50	18:30:00	23:00:00	4.5000	65.250000
2022-02-16 00:00:00	Desiree	Gardner	14.50	18:30:00	23:00:00	4.5000	65.250000
2022-02-17 00:00:00	Mindy	Sloan	17.25	10:30:00	14:00:00	3.5000	60.375000
2022-02-17 00:00:00	Luqman	Cantu	21.50	10:30:00	14:00:00	3.5000	75.250000
2022-02-17 00:00:00	Lilly-Rose	Vaughn	14.50	10:30:00	14:00:00	3.5000	50.750000
2022-02-17 00:00:00	Desiree	Gardner	14.50	10:30:00	14:00:00	3.5000	50.750000
2022-02-17 00:00:00	Mindy	Sloan	17.25	18:30:00	23:00:00	4.5000	77.625000
2022-02-17 00:00:00	Luqman	Cantu	21.50	18:30:00	23:00:00	4.5000	96.750000
2022-02-17 00:00:00	Lilly-Rose	Vaughn	14.50	18:30:00	23:00:00	4.5000	65.250000
2022-02-17 00:00:00	Desiree	Gardner	14.50	18:30:00	23:00:00	4.5000	65.250000

MySQL Workbench results for shift calculation.

Explanation:

- **Shift Calculation:** This query calculates the total hours worked during a shift by subtracting the start time from the end time. The calculated hours are then multiplied by the employee's hourly rate to determine the total labor cost for each shift. This is crucial for monitoring and controlling labor expenses, ensuring efficient staff management.

Conclusion

The completion of this Database Management Project for Benvenuto's Pizzeria marks a significant milestone in the development of my SQL and database management skills. Through this project, I focused on two critical aspects: the design and creation of a relational database and the development of custom SQL queries to extract meaningful insights from the stored data.

The first component of the project involved designing a robust relational database that met the specific needs of Benvenuto's Pizzeria. By thoroughly understanding the business requirements, I was able to create a well-structured schema that efficiently organized data related to orders, inventory, and staff management. The process of normalizing the data and defining relationships between tables was crucial in ensuring data integrity and minimizing redundancy. This foundational work provided a scalable and maintainable database system that could adapt to the pizzeria's evolving needs.

In the second component, I developed custom SQL queries that enabled the extraction, manipulation, and analysis of data stored within the database. These queries were designed to address real-world business scenarios, such as monitoring stock levels,

tracking employee hours, and analyzing sales patterns. The ability to craft complex SQL queries demonstrated my proficiency in data management and my ability to translate business requirements into actionable insights.

This project not only solidified my understanding of database concepts but also showcased my ability to apply these skills in a practical context. By focusing on the design and querying aspects of the database, I was able to deliver a solution that is both technically sound and aligned with the operational needs of Benvenuto's Pizzeria. The knowledge and experience gained through this project will undoubtedly enhance my ability to tackle future challenges in the field of data management and analysis.