Ben's Pizza Shop Database Project

# Project Brief:

Ben is opening a new pizza shop in his town, focusing on takeout and delivery services. Our project involves designing and building a custom relational database to store and generate all the necessary business data. This database will support the creation of dashboards to monitor business performance.

# Database Design:

Key areas of focus:

1. Orders

2. Stock levels

3. Staff

# Order Requirements:

Initially, Ben provided a basic outline for the order table, including:  
- Order ID  
- Item name  
- Item price  
- Quantity  
- Customer name  
- Delivery address  
  
Using data normalization, we identified additional necessary columns by examining the menu and business needs. The refined order table now includes:  
- 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

# Stock Control Requirements:

To manage stock efficiently, we need detailed information about:  
- Ingredients used in each pizza  
- Ingredient quantities based on pizza size  
- Existing stock levels  
  
We created separate tables for recipes and ingredients and an inventory table to manage stock levels.

# Staff Data Requirements:

Ben wants to track staff schedules and associated costs. We created the following tables:  
- Staff shift  
- Rota  
  
These tables help manage staff schedules and calculate the cost of each pizza, factoring in ingredients, chef labor, and delivery costs.

# Database Creation:

After designing the tables, we wrote SQL queries to create the database. The client requested various dashboards, so we first used SQL to create views and then employed these SQL queries in Looker Studio to generate the necessary visualizations.

# Dashboard Visualizations:

1. Dashboard 1:

- Total orders  
- Total sales  
- Total items  
- Average order value  
- Sales by category  
- Top-selling items  
- Orders by hour  
- Sales by hour  
- Orders by address  
- Orders by delivery/pickup  
  
SQL Query Example:  
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

2. Dashboard 2:

- Total quantity by ingredient  
- Total cost by ingredient  
- Calculated cost of pizza  
- Percentage cost remaining by ingredient  
- List of ingredients to reorder based on remaining inventory  
  
SQL Query Example:  
SELECT  
 s1.item\_name,  
 s1.ing\_name,  
 s1.ing\_id,  
 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,  
 ing.ing\_weight,  
 ing.ing\_price,  
 SUM(o.quantity) AS order\_quantity,  
 r.quantity AS recipe\_quantity  
 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\_weight,  
 ing.ing\_price) s1

3. Dashboard 3:

- Total staff cost  
- Total hours worked  
- Hours worked by staff members  
- Cost per staff member  
  
SQL Query Example:  
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

# Visualization and Data Storage:

To create the visualizations, the following technical tools and techniques were used:  
  
1. Google Cloud Platform (GCP):  
- A MySQL instance was created in Google Cloud to store and manage the database.  
- A Google Cloud Storage bucket was set up to store data files.  
  
2. Google Cloud SQL:  
- SQL queries were executed in Google Cloud SQL to create the database views required for visualizations.  
  
3. Looker Studio:  
- Looker Studio was connected to the MySQL instance in Google Cloud.  
- Custom queries were run in Looker Studio to extract the required data for visualizations.  
  
4. Visualization Techniques:  
- Scorecards were used to display total orders, total sales, and total items.  
- Calculated fields were created for metrics such as total sales (item\_price \* quantity) and average order value (SUM(total\_sale) / COUNT(DISTINCT(order\_id))).  
- Donut charts were used for sales by category, with total sales as the metric and item category as the dimension.  
- Bar charts were used for top-selling items, with total sales as the metric and item name as the dimension.  
- Line charts were used for orders by hour and sales by hour, displaying total orders and total sales.  
- Google Maps bubble maps were used for orders by address, with concatenated address fields to avoid location errors.  
- Pie charts were used for orders by delivery/pickup, with delivery as the dimension and order ID as the metric.  
- Tables were used for ingredient costs and quantities, with conditional formatting to highlight ingredients that need reordering.  
- Blended tables combined multiple tables to create comprehensive views.  
- Date range controls and data filters were added to ensure data consistency across different dates.  
  
These visualizations enable Ben to efficiently manage his pizza shop, monitor performance, and make informed business decisions.