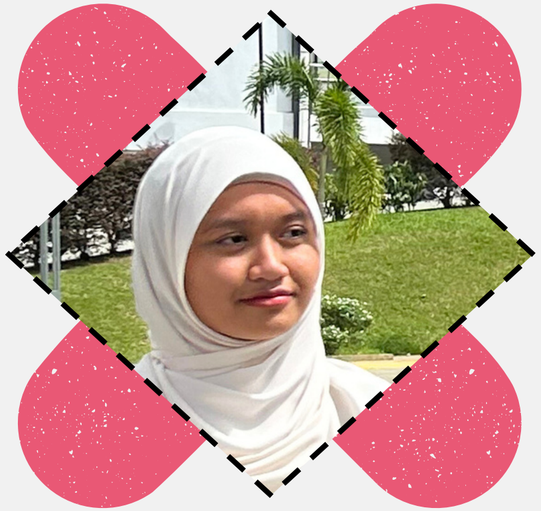
UNIVERSITI TEKNOLOGI MARA
Cawangan Pahang
Kampus Raub

ICT 200: INTRODUCTION TO DATABASE DESIGN

Dato's Bakery Shop System

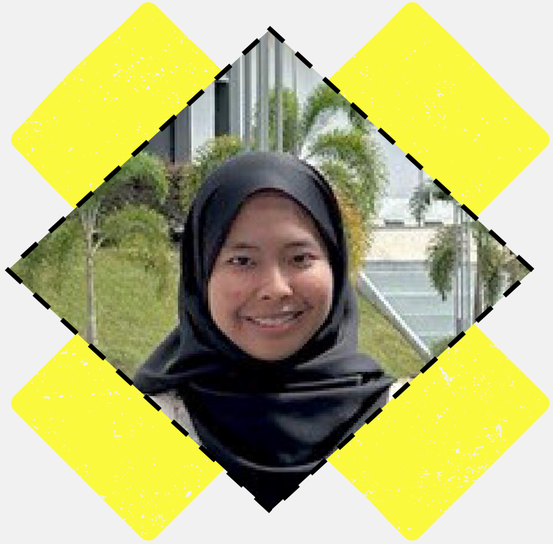
GROUP 1
CDCS110 3D

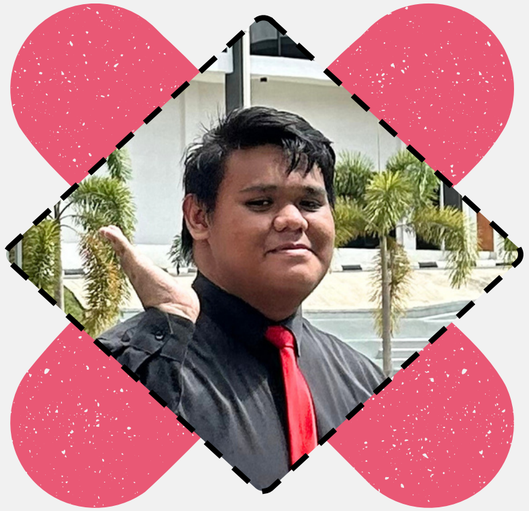
TEAM MEMBERS



NURIN IMAN
BINTI MASNGOT

SHAZWANA HUSNA
BINTI SAARI

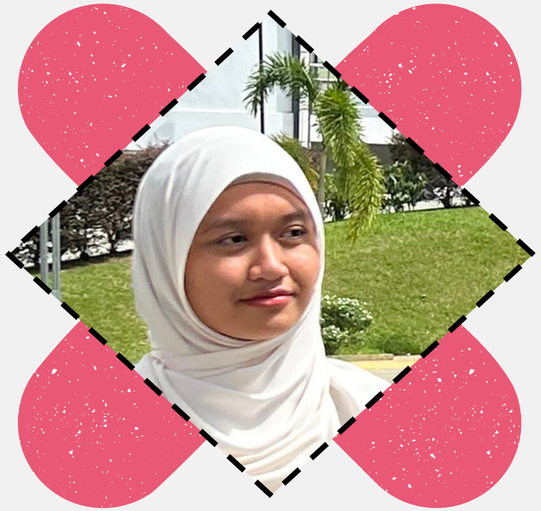




MUHAMMAD AIDIEL
BIN MOHAMAD
HUSSIN

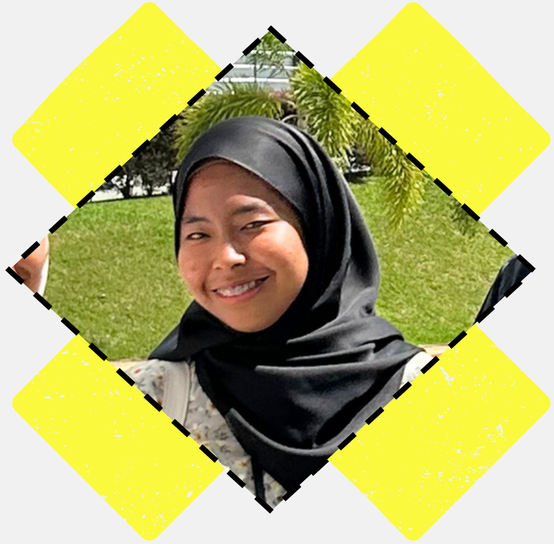
MUHAMMAD NAZHAN
BIN ROZAINI

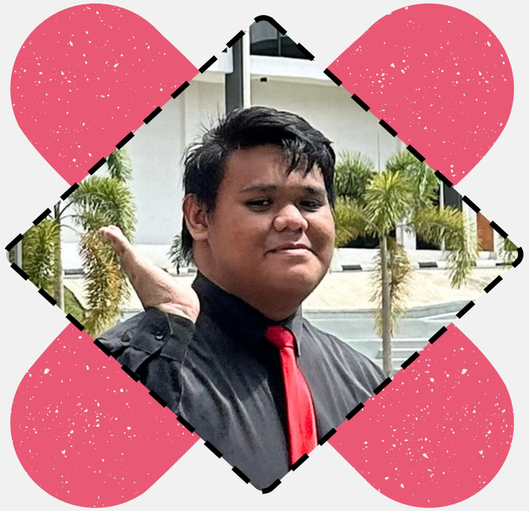
TEAM MEMBERS



NURIN IMAN
BINTI MASNGOT

SHAZWANA HUSNA
BINTI SAARI

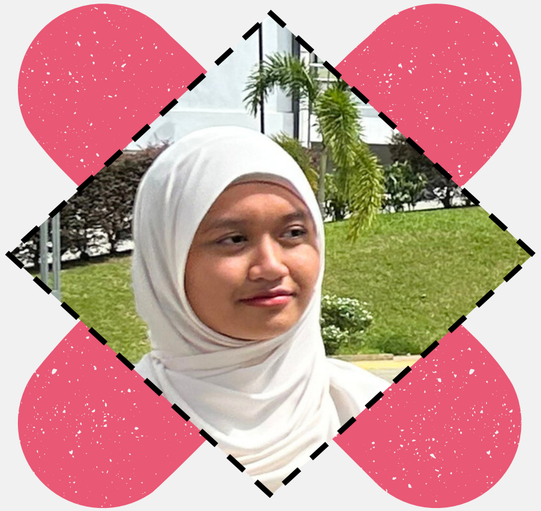




MUHAMMAD AIDIEL
BIN MOHAMAD
HUSSIN

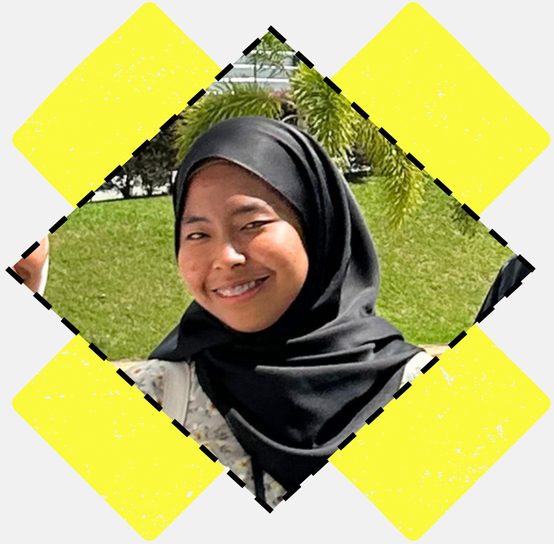
MUHAMMAD NAZHAN
BIN ROZAINI

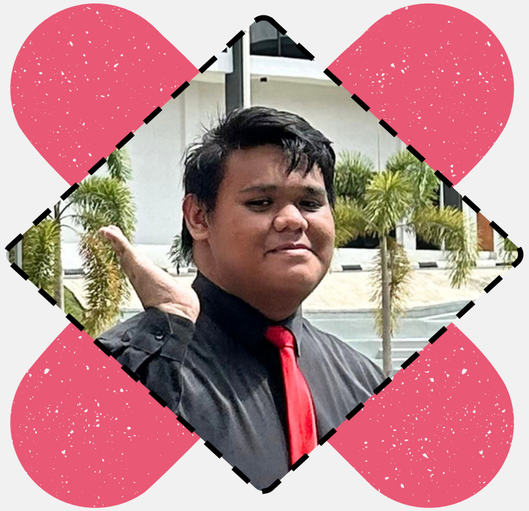
TEAM MEMBERS



NURIN IMAN
BINTI MASNGOT

SHAZWANA HUSNA
BINTI SAARI

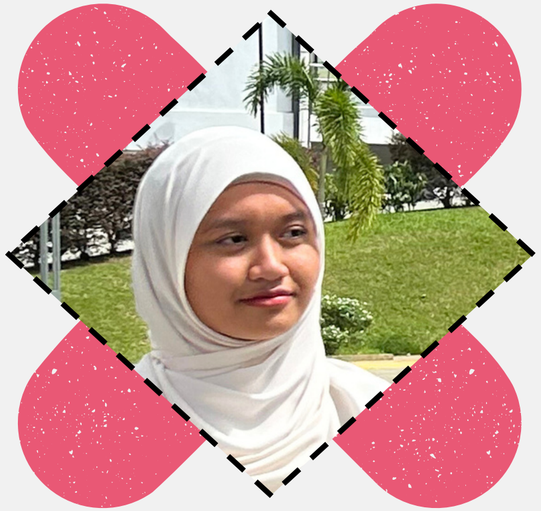




MUHAMMAD AIDIEL
BIN MOHAMAD
HUSSIN

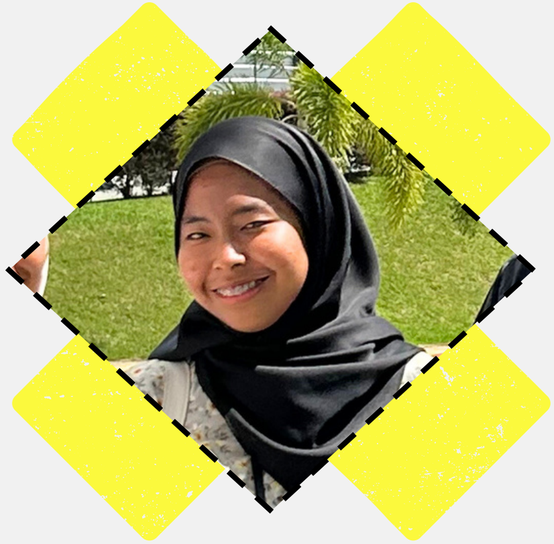
MUHAMMAD NAZHAN
BIN ROZAINI

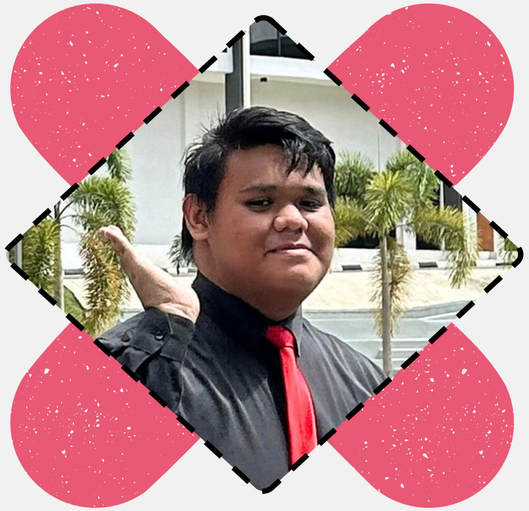
TEAM MEMBERS



NURIN IMAN
BINTI MASNGOT

SHAZWANA HUSNA
BINTI SAARI

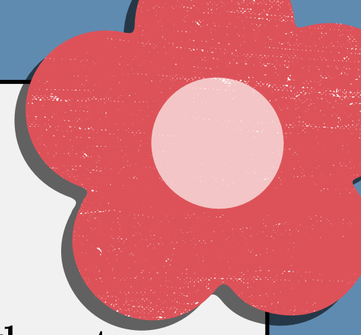




MUHAMMAD AIDIEL
BIN MOHAMAD
HUSSIN

MUHAMMAD NAZHAN
BIN ROZAINI

COMPANY BACKGROUND



- Dato's Bakery Shop is a highly prestigious company that is well-known throughout Malaysia.
- Established in 2016 when one of the owners of this company began to dabble in the field of entrepreneurship.
- Dato's Bakery Shop has 25 branches throughout Malaysia.
- Have such a classic and elegant shop decoration that attracts the attention of people.

VISION

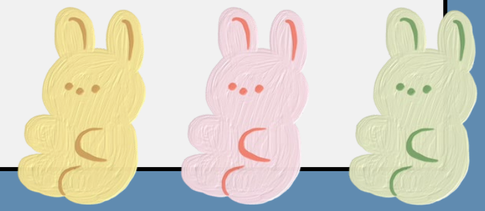
To expand the company's legacy to the world level.

MISSION

To serve desserts at reasonable prices using premium ingredients.

OBJECTIVES

- To mediate and promote all dessert innovations.
- ensures that sales products can be exported throughout the country.



CURRENT SYSTEM DESCRIPTION

The Bakery Shop uses a manual system which is a file system. In a manual system, the orders are taken and processed by hand instead of digital files.

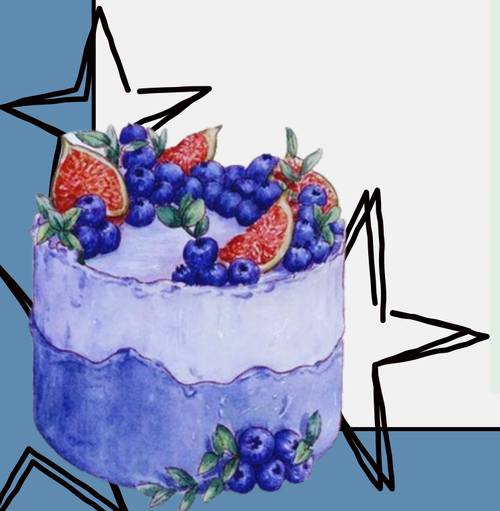
- Orders are written on paper and manually delivered to the kitchen
- Bill's calculations would be managed by hand
- Financial accounts using Microsoft Excel.

Staff in the bakery shop manually handle customer orders instead of using online systems.

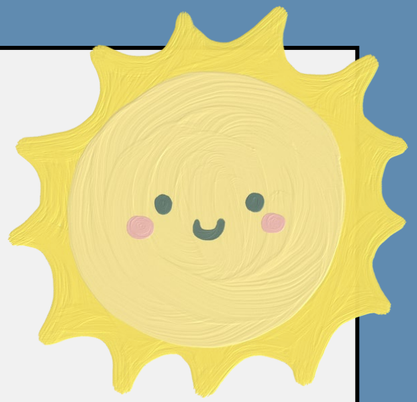
- Menu in the form of a book
- Relay it to the person serving the food.
- Customer would have to confirm their orders.
- The total of each food will be calculated to be included in their bills.

This bakery shop does not offer delivery .

- If a customer chooses to dine in, staff members would deliver the food to the customer's table.
- Apart from that, customers can pick up their orders at the counter
- Staff will process cash transactions instead of using an electronic payment method



CURRENT SYSTEM DESCRIPTION



COMPANY'S OPERATIONAL COMPONENTS (ENTITIES)

- STAFF
- CUSTOMER
- ORDER_ITEMS
- DESSERT
- CAKE
- PASTRY
- PAYMENT

RELATIONSHIP BETWEEN ENTITIES

- A STAFF can handle MANY CUSTOMER at a time. (1:M)
- ONE PAYMENT can be attached to MANY ORDER_ITEMS (1:M)
- A STAFF can handle MANY ORDER_ITEMS (1:M)
- A CUSTOMER may submit MANY ORDER_ITEMS at a time (1:M)
- ONE DESSERT can include MANY ORDER_ITEMS. (1:M)

PROBLEM STATEMENT

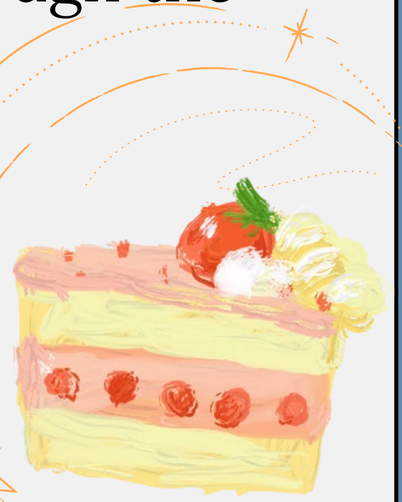
1. Data accessibility issue
   - Lack of data accessibility for the employees to use.
   - Creates inconsistency in order detail each employee receives.

2. Manual order calculation
   - Employees manually calculate order they received. Leading to complications in the bakery's sales reporting and potential mistakes.

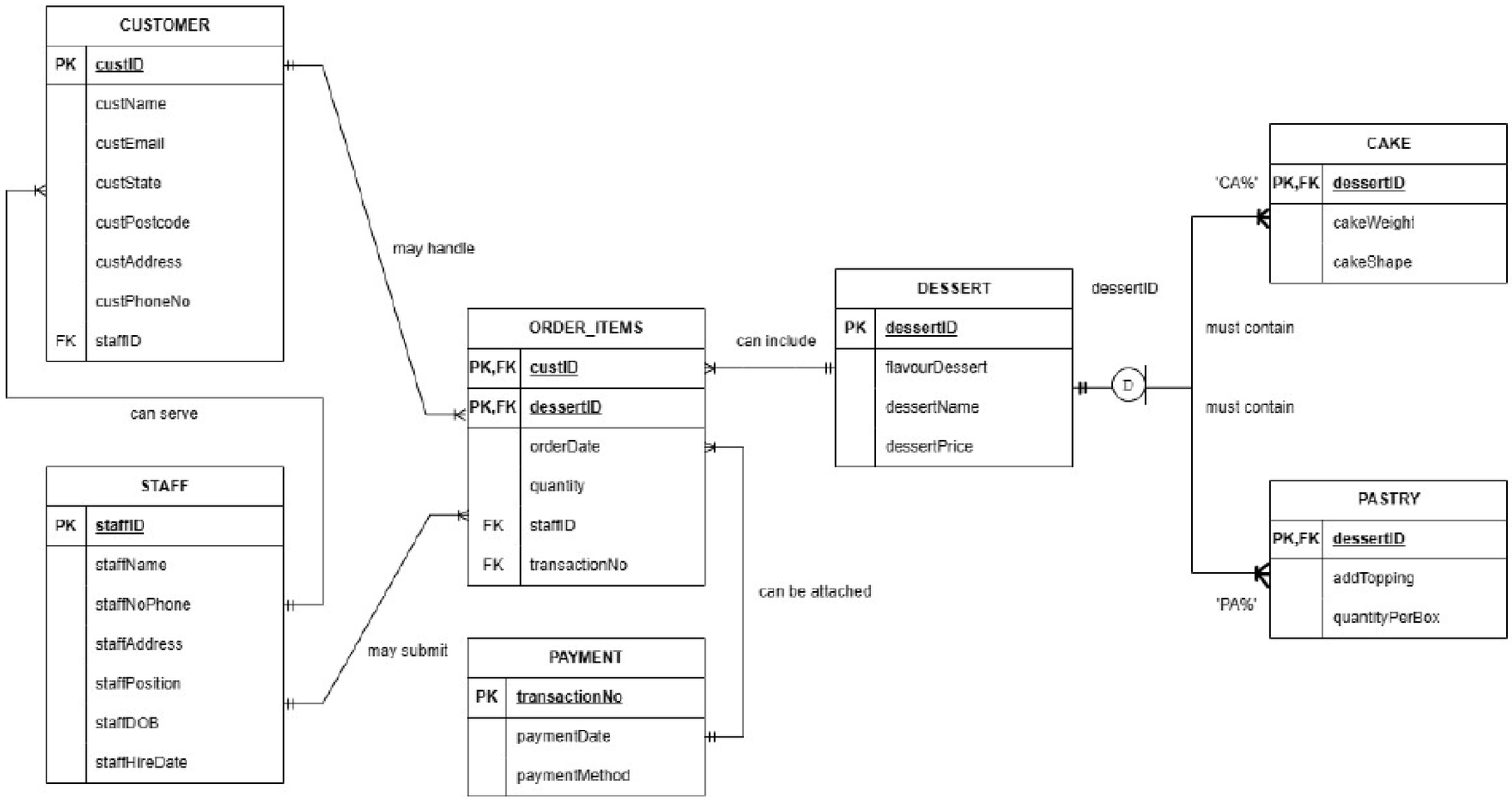
3. Inability to record electronic payments
   - The current system does not support electronic payment recording, limiting purchasing options for customer.

OBJECTIVES

- To increase the efficiency and productivity of the system by making the data easier to access established in the Bakery Shop.
- To create a database that can hold customer orders.
- To improve the organization of customer data collected through the creation of a database.
- To enhance customer experience while using the updated system.



ENTITIY RELATIONSHIP DIAGRAM (ERD)



DML

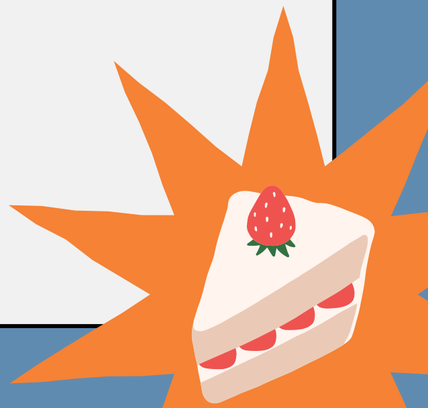
RETRIEVING DATA FROM MULTIPLE TABLE:

- Display the dessert names, pastry types, and quantity per box for each dessert that has pastries .

```
SELECT  d.dessertID, d.dessertName, p.addTopping, p.quantityPerBox
FROMdessert d
JOIN   pastry p      ON d.dessertID = p.dessertID;
```

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

dessertID	dessertName	addTopping	quantityPerBox
PA001	SOFTWAFFLE	CHOCOLATE RICE	6
PA002	SPEAK NOW ICE CREAM	CREAM CHEESE	4
PA003	CROMBOLONI	WHIPPING CREAM	3
PA004	CINAMON GIRL PIE	RAINBOW RICE CHOCOLATE	5
PA005	SWEET SERENDIPITY TREATS	ALMOND	5
PA006	BOMBOLONI	GRATED CHEESE	6
PA007	CREPE BANANA	MIX FRUIT	4
PA008	WAFFLE	NUTELLA	3
PA009	MLIK DONUT	CORNFLAKES	2
PA010	PIE RASPBERRY	SUGAR	4



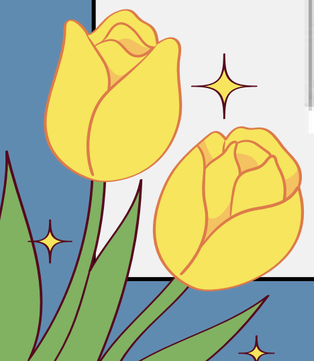
DML

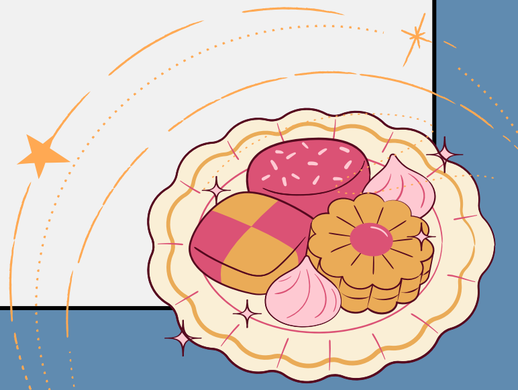
SIMPLE SQL QUERIES

- Display the information of order item id for each order item that only customer who ordered for cake and quantity less than 2 . Data that must show are dessertID, dessertName, and quantity .

```
SELECT oi.dessertID, d.dessertName, oi.quantity
FROM order_items oi
JOIN dessert d ON oi.dessertID = d.dessertID
WHERE d.dessertID LIKE 'CA%' AND oi.quantity < 2;
```

	dessertID	dessertName	quantity
▶	CA004	ALMOND CHEESECAKE	1
	CA010	FLUFFY BANANA CAKE	1





DML

COLUMN FUNCTION AND GROUPING

- Count how many orders for each customer ID and display the customer ID, customer name and total orders where customer ID must be sorting into ascending .

	custID	custName	totalOrders
▶	C001	NORAIMAN IKRAM	1
	C002	NUR NAJWA	1
	C003	NUR SABRINA	2
	C004	ASJAD HAMIZAN	2
	C005	MELISSA SOFIA	3
	C006	SHUHADA AMIR	2
	C007	SYAKIR IMAN	2
	C008	AIDA ZAMANI	3
	C009	AHMAD AMIRUL NAZIM	1
	C010	SITI FATIMAH ALIAS	1

```
SELECT c.custID, c.custName,
COUNT(o.transactionNo) AS totalOrders
FROM customer c
LEFT JOIN order_items o ON c.custID = o.custID
GROUP BY c.custID, c.custName
ORDER BY c.custID ASC;
```

DML

SUBQUERY

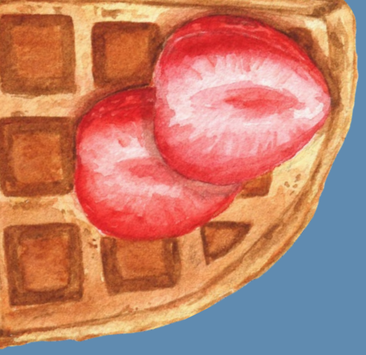
- Identify the five most popular dessert flavors (the most sold). Calculate the sales for each of the dessert to compare the result.

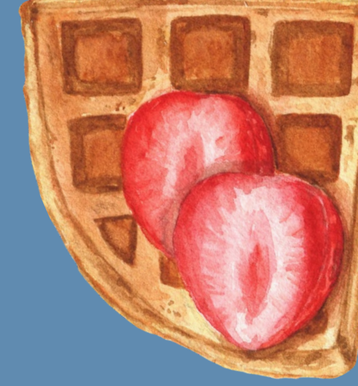
```
SELECT flavourDessert, totalQuantity, numCustomers, totalSales
FROM(
    SELECT d.flavourDessert,
        COUNT(DISTINCT oi.custID) AS numCustomers,
        SUM(d.dessertPrice * oi.quantity) AS totalSales,
        RANK() OVER(ORDER BY SUM(oi.quantity) DESC) AS ranking,
        SUM(oi.quantity) AS totalQuantity
    FROM dessert d
    JOIN order_items oi ON d.dessertID = oi.dessertID
    GROUP BY d.flavourDessert
) AS ranked
WHERE ranking <= 5;
```

flavourDessert	totalQuantity	numCustomers	totalSales
GREEN TEA	6	3	121.20
CHEESE	5	3	326.00
BLUEBERRY	4	2	192.40
VANILLA	4	2	362.00
MATCHA	4	2	319.60

CONCLUSION

- Relational Structure :
  - Complex structure designed for different preparing processes.
  - Tables include orders, customers, and cakes, linked by foreign key constraints.
- Key Connection :
  - staffID property links staff and customer tables, promoting accountability and transparency.
- System Features :
  - Well-selected sample data in each table provides insights into system functionality.
- Efficiency Enhancement :
  - Strategic use of indexes on columns expedites query processing.
- Strategic Positioning :
  - Dato's Bakery Shop is well-equipped for industry challenges.
  - Relational strategy showcases dedication to data integrity, accountability, and performance optimization.





Thank You !