Dato's Bakery Shop System
Final Report

ICT200
Introduction to Database Design

SEMESTER OCTOBER 2023 – FEBRUARY 2024

GROUP: CDCS1103D

NURIN IMAN BINTI MASNGOT, 2022877512

SHAZWANA HUSNA BINTI SAARI, 2022614994

MUHAMMAD AIDIEL BIN MOHAMAD HUSSIN, 2022478924

MUHAMMAD NAZHAN BIN ROZAINI, 2022605596

DR. AISYAH BINTI MAT JASIN

KPPIM, UiTM Pahang

UNIVERSITI TEKNOLOGI MARA

College of Computing, Informatics and Mathematics

ICT200:

INTRODUCTION TO DATABASE DESIGN

DATO'S BAKERY SHOP SYSTEM

GROUP :

CDCS1103D

PREPARED BY:

NURIN IMAN BINTI MASNGOT (2022877512)

SHAZWANA HUSNA BINTI SAARI (2022614994)

MUHAMMAD AIDIEL BIN MOHAMAD HUSSIN (2022478924)

MUHAMMAD NAZHAN BIN ROZAINI (2022605596)

PREPARED FOR:

DR. AISYAH BINTI MAT JASIN

DATE SUBMISSION: 22 JANUARY 2024

SESSION: OCTOBER 2023 – FEBRUARY 2024

Evaluation Form for the Final Database Project (ICT200)-20%

Project Report Proposal & Final Report Grading Rubric

Project Title : DATO'S BAKERY SHOP SYSTEM **Group No:** 1

Student's Name:

1. NURIN IMAN BINTI MASNGOT (2022877512)
2. SHAZWANA HUSNA BINTI SAARI (2022614994)
3. MUHAMMAD AIDIEL BIN MOHAMAD HUSSIN (2022478924)
4. MUHAMMAD NAZHAN BIN ROZAINI (2022605596)
5. _____

Criteria	Score					Weight	Total score	Total / section
	1	2	3	4	5	*w*	*w x score*	
Format	Major errors in formatting	Minor errors in formatting	All specified formatting style has been portrait.	NA	NA	1		3
Company background	Poor description of facts. Much information is not provided	Paragraph lack of clear ideas. Not enough information provided from the organization.	Some paragraphs have clear ideas, but the paragraph transitions are weak. Provide basic information from the organization	Most paragraphs have clear ideas, and provide additional information from the organization	All paragraphs have clear ideas and provide additional information and necessary pictures or images from the organization.	1		5
Current System Descriptions	Content is not sound.	Content is sound and solid; description is present but not	Well-presented; descriptions are detailed, developed, and supported	Exceptionally well-presented; descriptions are detailed, well-developed, supported	Exceptionally well-presented; descriptions are detailed, well-developed, supported	1		5

		particularly developed or supported; some evidence, but usually of a generalized nature. Only provide process flow chart.	with evidence and details, mostly specific.	with specific evidence & facts, as well as examples	with specific evidence & facts, as well as examples and process flow chart			
Problem statements	Provide less than three statements. No descriptions and examples given. Provide wrong facts.	Provide less than three statements with brief description. No examples given. Irrelevant explanations and examples given.	Provide three statements. No descriptions and examples given	Provide three statement and clearly identifies and summarizes main issues related with manual system or file-based approach.	Provide three statement and clearly identifies and summarizes main issues related with manual system or file-based approach. They also clearly stated some relevant sub-problems because of the manual system. Examples of the problems are given.	**1**		**5**
System Objectives	Provide less than three objectives. No descriptions and examples given. Provide wrong facts.	Provide less than three objectives with brief description. No examples given. Irrelevant explanations and examples given.	Provide three objectives. No descriptions and examples given	Provide three objectives of the project. The objectives have been specified clearly.	Provide three objectives of the project. The objectives have been specified clearly. They stated the that they want to analyse, design, and develop as the objective.	**1**		**5**
DATABASE DESIGN								
Final ERD	Wrong ERD	Many errors in connectivities, labels and notations.	Some errors in connectivities, labels and notations.	Some errors in labels only. All connectivities and notation are correct.	Use correct notation, provide correct connectivity, has relationship and cardinality.	**1**		**5**
3NF Relational schema	Incorrect form of relational schema	>3 tables are not in 3NF.	1-2 tables are not in 3NF.	All tables have been in 3NF but PK and FK not proper display. Use proper table name.	All tables have been in 3NF. Show proper PK and FK. Also use proper table name.	**1**		**5**
Data dictionary	Incorrect form of data dictionary	>3 table incorrect of data type, PK and FK	1-2 table incorrect of data type, PK and FK	All the table with correct data type but PK Not complete.	All the table with correct data type, complete PK and FK	**1**		**5**

DATABASE IMPLEMENTATION								
Data Definition Language (DDL)								
CREATE TABLE statement	There's command with major errors.	There's command with minor errors.	More than three tables' commands are not included.	One/two of the tables' commands are not included	Provide commands for ALL tables	**1**		**5**
Data Manipulation Language (DML)								
1. SELECT statement for ALL tables with minimum 10 records for appropriate tables	The records from more than five tables are not included	The records from three/four tables are not included	The records from two tables are not included	The records from one table are not included	The relevant records for ALL tables are included	**1**		**5**
2. Retrieving Data from Multiple Tables. (at least 5 queries)	One query is included and incorrect command	Two queries are included. Irrelevant but correct command	Three queries are included and correct command	Four queries are included. Provide correct, good, and relevant commands	Five queries are included. Provide correct, meaningful, excellent, and relevant commands	**2**		**10**
3. Simple Query (at least 5 queries) (Comparison operator, logic operator, IN, BETWEEN, LIKE)	One query is included and incorrect command	Two queries are included. Irrelevant but correct command	Three queries are included and correct command	Four queries are included. Provide correct, good, and relevant commands	Five queries are included. Provide correct, meaningful, excellent, and relevant commands	**1**		**5**
4. Column Functions (at least 3 queries)	One query is included and incorrect command	Two queries are included. Irrelevant but correct command.	Three queries are included and correct command	Three queries are included. Provide correct, good, and relevant commands	Three queries are included. Provide correct, meaningful, excellent, and relevant commands	**1**		**5**
5. Using Subqueries (at least 3 queries)	One query is included and incorrect command	Two queries are included. Irrelevant but correct command	Three queries are included and correct command	Three queries are included. Provide correct, good, and relevant commands	Three queries are included. Provide correct, meaningful, excellent, and relevant commands	**1**		**5**
References	The sources are not cited correctly according to APA style, nor listed correctly. Only one reference is listed	There may be a few errors in APA style citation. Only two references are listed	Listed references conform to APA style citation. Provide at least three references.	NA	NA	**1**		**3**
						TOTAL		**76**

Project presentation

Project Title : DATO'S BAKERY SHOP SYSTEM **Group No:** 1 **Time Present:** _____

Student's Name:

1. NURIN IMAN BINTI MASNGOT (2022877512)
2. SHAZWANA HUSNA BINTI SAARI (2022614994)
3. MUHAMMAD AIDIEL BIN MOHAMAD HUSSIN (2022478924)
4. MUHAMMAD NAZHAN BIN ROZAINI (2022605596)
5. _____

Criteria	Score					weight	Total Score	Total / section
	1	2	3	4	5	*w*	*w x score*	
Individual Marks								
Delivery (fluent/confidence/not reading) & Presentation Flow	Audience cannot understand the presentation because there is no sequence of information. Reads all of report with no eye contact	Audience has difficulty following presentation because there is no sequence of information. Most of the time reads the notes/slides	Presents information in logical sequence which audience can follow. Frequently returns to notes/slides	Presents information in logical sequence which audience can follow. Seldom returning to notes/slides	Presents information in logical and interesting sequence which audience can follow. Seldom returning to notes/slides	1		5
Teamwork (Foster good relationship)	No clear evidence of ability to foster good relationships and work	No clear evidence of ability to foster good relationships and	clear evidence of ability to foster good relationships	Clear evidence of ability to foster good relationships and	Very Clear evidence of ability to foster good	1		5

	together effectively with other group members towards goal achievement	work together effectively with other group members towards goal achievement	and work together effectively with other group members towards goal achievement.	work together effectively with other group members towards goal achievement	relationships and work together effectively with other group members towards goal achievement.			
Participation cooperation	Lacks any ability to be a good listener and follow direction.	Inconsistent ability to listen and follow direction.	Mostly able to listen to and follow directions. Ability to employ social skills substantially successful.	Demonstrate willingness to work with others in new activities. Demonstrates ability to listen and follow direction in most cases.	Shows an ability to accept and respect achievement level in self and others.	**1**		**5**
Slides Content								
Database design and case study.	All points are incorrect and Wrong ERD	Some incorrect points and explanations are given.	Only stated the points without explained.	Well explained but there's no further elaboration on the examples of the case study	Well explained and support with relevant examples of the case study	**1**		**5**
Questions on Queries								
DML-Simple query	Not attempt to answer the question	An incorrect command	Correct command in third attempts	A correct command in two attempts	A correct command with one attempt	**1**		**5**
DML – group function	Not attempt to answer the question	An incorrect command	Correct command in third attempts	A correct command in two attempts	A correct command with one attempt	**1**		**5**
DML-Join on Multiple (Joining) Tables	Not attempt to answer the question	An incorrect command	Correct command in third attempts	A correct command in two attempts	A correct command with one attempt	**1**		**5**
DML - subquery	Not attempt to answer the question	An incorrect command	Correct command in third attempts	A correct command in two attempts	A correct command with one attempt	**1**		**5**
						TOTAL		**40**

TOTAL : / 116 MARKS

TABLE OF CONTENT

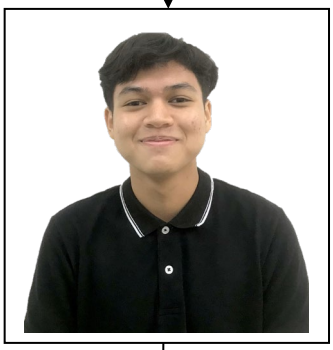
1.0	MEMBER'S PROFILE	1
2.0	COMPANY BACKGROUND	2
3.0	CURRENT SYSTEM DESCRIPTION	3 - 6
4.0	PROBLEM STATEMENT	7
5.0	OBJECTIVES OF PROPOSED DATABASE	8
6.0	DATA DESIGN	
	6.1 ENTITY DIAGRAM RELATIONSHIP (ERD)	9
	6.2 3NF RELATIONAL SCHEMA	9
	6.3 DATA DICTIONARY	10 - 13
7.0	DATABASE IMPLIMENTATION	
	7.1 SECTION A: DATA DEFINITION LANGUAGE (DDL)	14 - 17
	7.2 SECTION B: DATA MANIPULATING LANGUAGE (DML)	18 - 30
8.0	CONCLUSION	31
9.0	REFERENCES	32

1.0 MEMBER'S PROFILE

NURIN IMAN BINTI MASNGOT
(2022877512)
PROJECT MANAGER

MUHAMMAD AIDIEL BIN MOHAMAD HUSSIN
(2022478924)
WEB DESIGNER

SHAZWANA HUSNA BINTI SAARI
(2022614994)
GRAPHIC DESIGNER



MUHAMMAD NAZHAN BIN ROZAINI
(2022605596)
WEB ANALYST

2.0 COMPANY BACKGROUND

Dato's Bakery Shop is a highly prestigious company that is well-known throughout Malaysia. This company has four shareholders, and they are also the main leaders of this company. This company was first established in 2016 when one of the owners of this company began to dabble in the field of entrepreneurship. Now, Dato's Bakery Shop has 30 branches throughout Malaysia. Dato's Bakery Shop is famous for selling delicious and cheap desserts. Not only that, but they also have such a classic and elegant shop decoration that attracts the attention of people to come to this shop.

Dato's Bakery Shop has its vision and mission. Their vision is to expand the company's legacy to the world level where all countries will recognize our company. Their mission is to serve desserts at reasonable prices using premium ingredients that all sections of society can get.

Their objectives are to mediate and promote all dessert innovations that are healthy and can be enjoyed by all ages. In addition, it ensures that sales products can be exported throughout the country with the best quality.

3.0 CURRENT SYSTEM DESCRIPTION

The Bakery Shop typically 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. For instance, they would write down the orders on paper and manually deliver them to the kitchen instead of using electronic devices to take orders. Similarly, the bill's calculations would be managed by hand instead of using a computerized system. Additionally, they manage their financial accounts using Microsoft Excel.

Then, staff in the bakery shop may manually handle customer orders instead of using online systems. For instance, when customers want to look at the menu and place an order for cake or pastries, they will provide the tangible menu in the form of a book that has already been printed. When a customer selects a menu item and quantity that they want, the staff will note it and relay it to the person serving the food. When the customer has made their order, then they would need to confirm their orders. Then, the total of each food will be calculated to be included in their bills.

Other than that, this system of bakery shops does not provide for delivery, they would need to choose either dine-in or takeaway. If a customer chooses to dine in, staff members would physically deliver the food plates from the kitchen to the customer's table. Apart from that, staff members will ring a bell to let customers pick up their orders at the counter when the meal is ready for takeaway. Furthermore, in the case that a customer wants to pay for their order, the staff will process cash transactions by taking payments in person and giving them a change instead of using an electronic payment method like a QR code scan.

3.1 COMPANY'S OPERATIONAL COMPONENTS (ENTITIES)

1. **STAFF:** Contains details about the staff of Dato's Bakery Shop. These are the attributes for the staff table, which includes staffID, staffName, staffNoPhone, staffAddress, staffPosition, staffDOB, and staffHireDate.

2. **CUSTOMER:** Represent information about the customers at the Dato's Bakery Shop. The attributes for the customer table are custID, custName, custPhoneNo, custEmail, custAddress, custPostcode, custState, and staffID.

3. **CAKE:** It is a subtype entity that inherits from DESSERT. Keep records of the variety of cakes that the customers may choose from. The attributes for the cake table are dessertID, cakeWeight, cakeShape.

4. **PASTRY:** It is a subtype entity that inherits from DESSERT. The customer can also choose many pastry that they want as this entity contains information about the pastries that the bakery sells. The attributes for the pastry table are dessertID, addTopping and quantityPerBox.

5. **DESSERT:** A bridge entity meant to convey information about cake orders included in customers' orders. This entity effectively links the ORDER_ITEMS entity with CAKE entity and PASTRY. It is also a supertype entity which is a parent to CAKE and PASTRY. The attributes for the order are dessertID, flavourDessert, dessertName, and dessertPrice.

6. **PAYMENT:** Represent information about the payment method that the customer needs to make a payment after confirming their order. The attributes for payment are transactionNo, paymentMethod, amount, and paymentDate.

7. **ORDER_ITEMS:** Represent information about the order details of each customer. The attributes are custID, dessertID, orderDate, quantity, staffID, and transactionNo.

3.2 RELATIONSHIP BETWEEN ENTITIES

1. **A STAFF** can handle **MANY CUSTOMER** at a time. This means that the staff can take many customers' order. The relationship **STAFF** between **CUSTOMER** is **ONE TO MANY**.

2. **ONE PAYMENT** can make **MANY ORDER_ITEMS**. The relationship is **ONE TO MANY**.

3. **A STAFF** can handle **MANY ORDER_ITEMS**. So, the relationship is **ONE TO MANY**.

4. **A CUSTOMER** may submit **MANY ORDER_ITEMS** at a time. This is typically the relationship is **ONE TO MANY**.

5. The relationship between **DESSERT** and **ORDER_ITEMS** is **ONE TO MANY** because **ONE DESSERT** can be included **MANY ORDER_ITEMS**.

6. **ONE DESSERT** can have **MANY PASTRY**. The customers can order and choose the types of pastry that they want along with different types of toppings as addition like cherry, chocolate, and cheese. Customers will also be able to pick how many pastries they want in a box as it is sold in sets. The relationship between **DESSERT** to **PASTRY** is **ONE TO MANY**.

7. **ONE DESSERT** can have **MANY CAKE**. The customer can order and choose the types of cake that they want with variety of shape and weight. So, **ONE TO MANY** is the relationship **DESSERT** between **CAKE**.

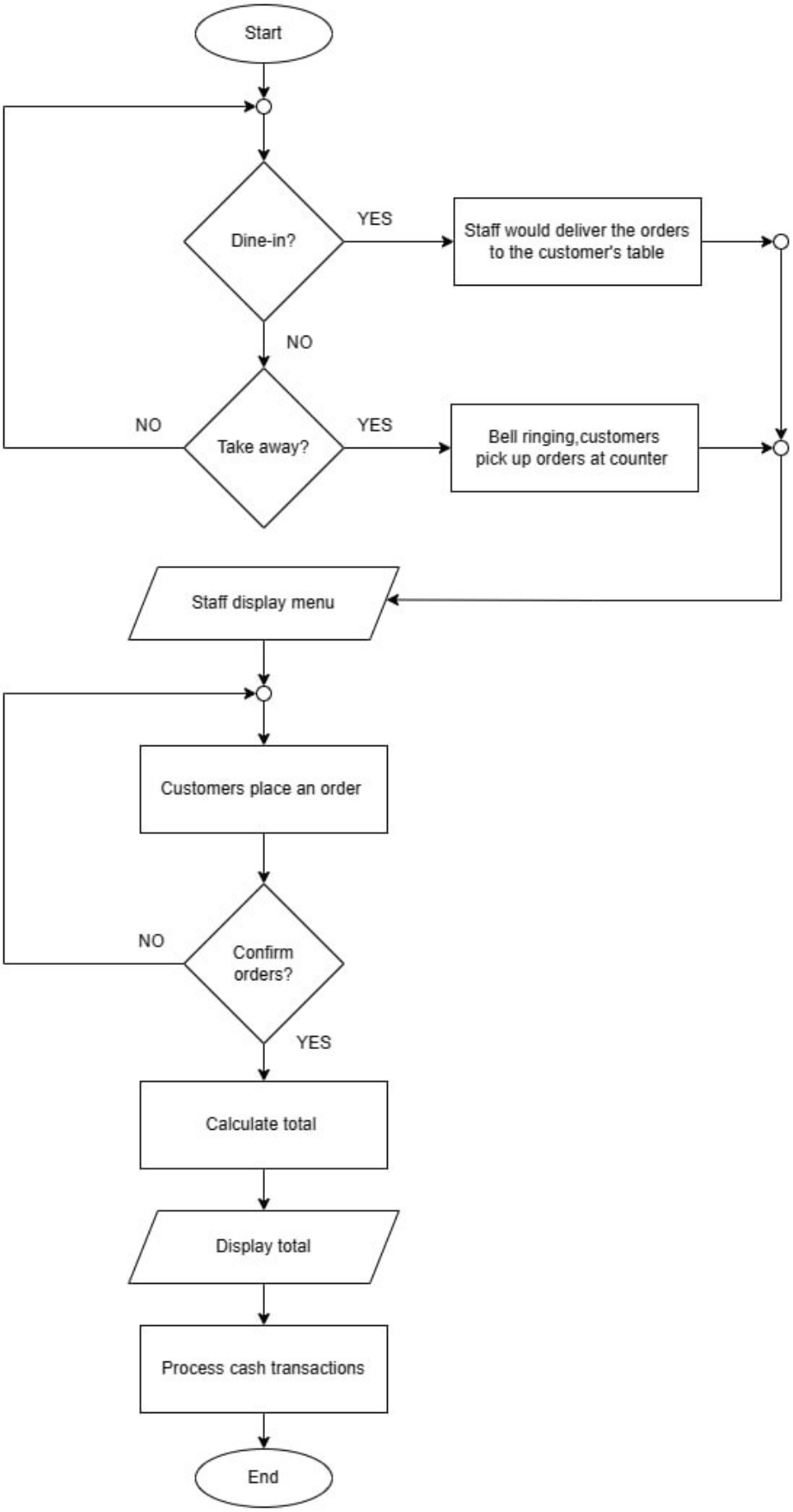


Diagram 1: Flowchart of flow segment system

4.0 PROBLEM STATEMENT

While the current system of Dato's Bakery Shop is functional and works well with the employees, we have identified several shortcomings that we noticed could be remedied.

The biggest issue in the system is the lack of data accessibility for the employees to use. This creates inconsistency between the orders and the details each employee receives. It means that specific employees must receive orders before they can be relayed to the kitchen employees. As a result, it puts a hindrance on communication within the bakery, allowing mistakes in customer service that are bound to bring negative consequences in the future.

Additionally, the employees would have to manually calculate the orders they received and enter them in their preferred system for sales analysis. This would make sales reports tedious, and mistakes and oversights are likely to occur. This factor could lead to complications in the bakery's sales reporting process.

Even though the current system is familiar to our customers since it is traditional, it adds more tasks for employees as its system is dependent on administrators who must allocate more of their time to serve the customers. Thus, suppressing the potential productivity of the employees.

Moreover, the current system does not allow electronic payments to be recorded, making it challenging for customers without cash to make purchases at the shop.

5.0 OBJECTIVE OF PROPOSED DATABASE

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

6.0 DATABASE DESIGN

6.1 ENTITY RELATIONAL DIAGRAM (ERD)

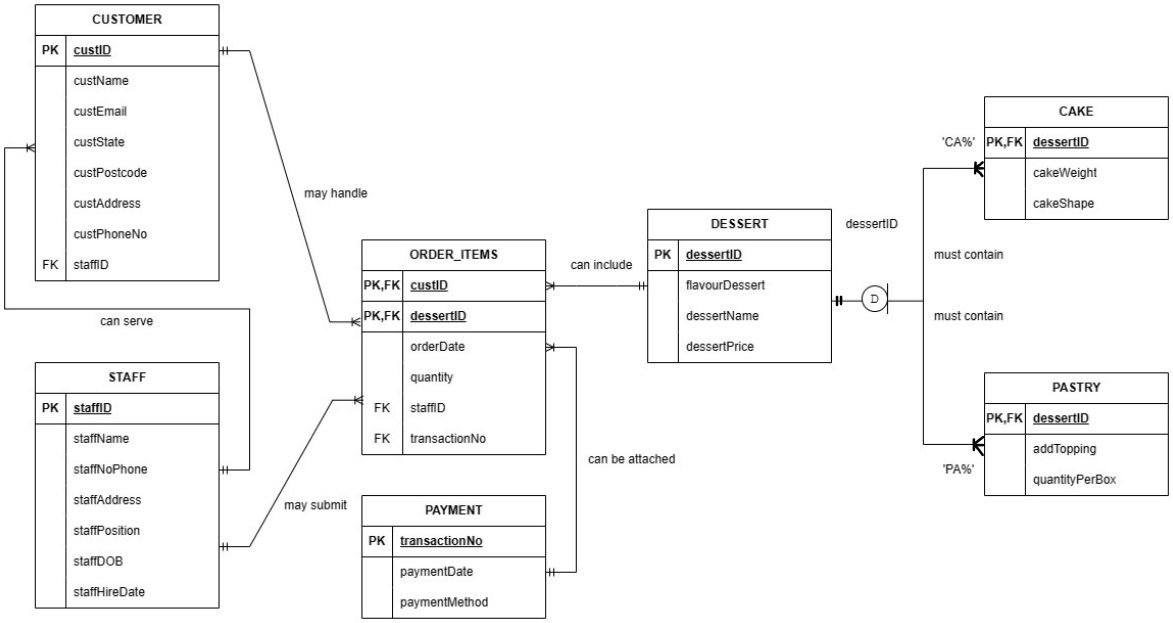


Diagram 2: Entity Relationship Diagram (ERD)

6.2 3NF RELATIONAL SCHEMA

CUSTOMER(custID, custName, custEmel, custState, custPostcode, custAddress, custPhoneNo, *staffID)

STAFF(StaffID, staffName, staffNoPhone, staffAddress, staffPosition, staffDOB, staffHireDate)

PAYMENT(transactionNo, paymentDate, paymentMethod)

ORDER_ITEMS(*custID, *dessertID, orderDate, quantity, *staffID, *transactionNo)

DESSERT(dessertID, flavourDessert, dessertName, dessertPrice)

CAKE(*dessertID, cakeWeight, cakeShape)

PASTRY(*dessertID, addTopping, quantityPerBox)

6.3 DATA DICTIONARY

TABLE NAME	ATTRIBUTE NAME	CONTENTS	DATA TYPE	FORMAT	RANGE	REQUIRED	PK OR FK	FK REFERENCE TABLE
CUSTOMER	custID	Customer ID	VARCHAR (10)	Xxxxxxxx		YES	PK	
	custName	Customer Name	VARCHAR (50)	Xxxxxxxx				
	custEmail	Customer Email	VARCHAR (35)	Xxxxxxxx				
	custState	Customer State	VARCHAR (30)	Xxxxxxxx				
	custPostcode	Customer Postcode	VARCHAR (6)	Xxxxxxxx				
	custAddress	Customer Address	VARCHAR (50)	Xxxxxxxx				
	custPhoneNo	Customer Phone Number	VARCHAR (10)	Xxxxxxxx				
	staffID	Staff ID	VARCHAR (10)	Xxxxxxxx			FK	STAFF
STAFF	staffID	Staff ID	VARCHAR (10)	Xxxxxxxx		YES	PK	

	StaffName	Staff Name	VARCHAR (50)	Xxxxxxxx				
	staffNoPhone	Staff Number Phone	VARCHAR (10)	Xxxxxxxx				
	staffAddress	Staff Address	VARCHAR (50)	Xxxxxxxx				
	staffPosition	Staff Position	VARCHAR (30)	Xxxxxxxx				
	staffDOB	Staff Date of Birth	DATE	mm-dd-yyyy				
	staffHireDate	Staff Hire Date	DATE	mm-dd-yyyy				
ORDER_ITEMS	custID	Customer ID	VARCHAR (10)	Xxxxxxxx		YES	PK, FK	CUSTOMER
	staffID	Order Staff ID	VARCHAR (10)	Xxxxxxxx		YES	PK, FK	STAFF
	orderDate	Order Date	DATE	Xxxxxxxx				
	quantity	Quantity	INTEGER	011	1-2901291256			
	staffID	Order Staff ID	VARCHAR (10)	Xxxxxxxx			FK	STAFF
	transactionNo	Transaction No	VARCHAR (15)	Xxxxxxxx			FK	PAYMENT

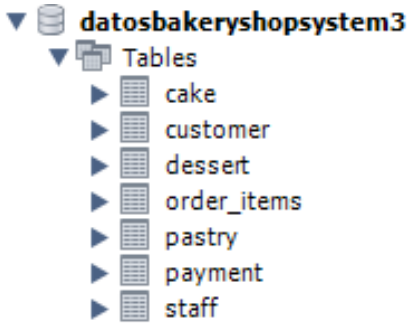
PAYMENT	transactionNo	Transaction No	VARCHAR (15)	Xxxxxxxx		YES	PK	
	paymentDate	Payment Date	DATE	mm-dd-yyyy				
	paymentMethod	Payment Method	VARCHAR (15)	Xxxxxxxx				
DESSERT	dessertID	Dessert ID	VARCHAR (10)	Xxxxxxxx		YES	PK	
	flavourDessert	Dessert Flavour	VARCHAR (50)					
	dessertName	Dessert Name	VARCHAR (50)					
	dessertPrice	Dessert Price	DECIMAL (7,2)					
CAKE	dessertID	Dessert ID	VARCHAR (10)	Xxxxxxxx		YES	PK, FK	DESSERT
	cakeWeight	Cake Weight	DECIMAL (3,2)					
	cakeShape	Cake Shape	VARCHAR (15)	Xxxxxxxx				
PASTRY	dessertID	Dessert ID	VARCHAR (10)	Xxxxxxxx		YES	PK, FK	DESSERT

	addTopping	Topping Dessert	VARCHAR (35)	Xxxxxxxx				
	quantityPerBox	Quantity Dessert per Box	VARCHAR (25)	Xxxxxxxx				

7.0 DATABASE IMPLEMENTATION

7.1 SECTION A: DATA DEFINITION LANGUAGE (DDL)

```
CREATE DATABASE DatosBakeryShopSystem3;
```




```
CREATE TABLE STAFF (

staffID VARCHAR(10) NOT NULL,

staffName VARCHAR(50) NOT NULL,

staffNoPhone VARCHAR(12) NOT NULL,

staffAddress VARCHAR(100) NOT NULL,

staffPosition VARCHAR(20) NOT NULL,

staffDOB date NOT NULL,

staffHireDate date NOT NULL

);
```

Column Name	Datatype	PK	NN	UQ	B	UN	ZF	AI	G	Default/Expression
staffID	VARCHAR(10)	☑	☑	☐	☐	☐	☐	☐	☐	
staffName	VARCHAR(50)	☐	☑	☐	☐	☐	☐	☐	☐	
staffNoPhone	VARCHAR(12)	☐	☑	☐	☐	☐	☐	☐	☐	
staffAddress	VARCHAR(100)	☐	☑	☐	☐	☐	☐	☐	☐	
staffPosition	VARCHAR(20)	☐	☑	☐	☐	☐	☐	☐	☐	
staffDOB	DATE	☐	☑	☐	☐	☐	☐	☐	☐	
staffHireDate	DATE	☐	☑	☐	☐	☐	☐	☐	☐	
		☐	☐	☐	☐	☐	☐	☐	☐	

```
CREATE TABLE CUSTOMER (
custID VARCHAR(10) NOT NULL,
custName VARCHAR(50) NOT NULL,
custEmail VARCHAR(50) NOT NULL,
custState VARCHAR(20) NOT NULL,
custPostcode INT(11) NOT NULL,
custAddress VARCHAR(100) NOT NULL,
custPhoneNo VARCHAR(12) NOT NULL,
staffID VARCHAR(10) NOT NULL,
FOREING KEY (staffID) REFERENCES STAFF(staffID)
);
```

Column Name	Datatype	PK	NN	UQ	B	UN	ZF	AI	G	Default/Expression
custID	VARCHAR(10)	☑	☑	☐	☐	☐	☐	☐	☐	
custName	VARCHAR(50)	☐	☑	☐	☐	☐	☐	☐	☐	
custEmail	VARCHAR(50)	☐	☑	☐	☐	☐	☐	☐	☐	
custState	VARCHAR(20)	☐	☑	☐	☐	☐	☐	☐	☐	
custPostcode	INT(11)	☐	☑	☐	☐	☐	☐	☐	☐	
custAddress	VARCHAR(100)	☐	☑	☐	☐	☐	☐	☐	☐	
custPhoneNo	VARCHAR(12)	☐	☑	☐	☐	☐	☐	☐	☐	
staffID	VARCHAR(10)	☐	☑	☐	☐	☐	☐	☐	☐	
		☐	☐	☐	☐	☐	☐	☐	☐	

```
CREATE TABLE ORDER_DETAILS (
custID VARCHAR(10) NOT NULL,
dessertID VARCHAR (10) NOT NULL,
orderDate date NOT NULL,
quantity INT(11) NOT NULL,
staffID VARCHAR (10) NOT NULL,
transactionNo VARCHAR(10) NOT NULL,
PRIMARY KEY(custID, dessertID),
FOREIGN KEY (custID) REFERENCES CUSTOMER(custID),
FOREIGN KEY (desserID) REFERENCES DESSERT(dessertID),
FOREIGN KEY (staffID) REFERENCES STAFF(staffID)
FOREIGN KEY (transactionNo) REFERENCES PAYMENT(transactionNo)
)
```

Column Name	Datatype	PK	NN	UQ	B	UN	ZF	AI	G	Default/Expression
custID	VARCHAR(10)	☑	☑	☐	☐	☐	☐	☐	☐	
dessertID	VARCHAR(10)	☑	☑	☐	☐	☐	☐	☐	☐	
orderDate	DATE	☐	☑	☐	☐	☐	☐	☐	☐	
quantity	INT(11)	☐	☑	☐	☐	☐	☐	☐	☐	
staffID	VARCHAR(10)	☐	☑	☐	☐	☐	☐	☐	☐	
transactionNo	VARCHAR(10)	☐	☑	☐	☐	☐	☐	☐	☐	
		☐	☐	☐	☐	☐	☐	☐	☐	

```
CREATE TABLE PAYMENT (
transactionNo VARCHAR(10) NOT NULL,
paymentDate date NOT NULL,
paymentMethod VARCHAR (15) NOT NULL
)
```

Column Name	Datatype	PK	NN	UQ	B	UN	ZF	AI	G	Default/Expression
transactionNo	VARCHAR(10)	☑	☑	☐	☐	☐	☐	☐	☐	
paymentDate	DATE	☐	☑	☐	☐	☐	☐	☐	☐	
paymentMethod	VARCHAR(15)	☐	☑	☐	☐	☐	☐	☐	☐	
		☐	☐	☐	☐	☐	☐	☐	☐	

```
CREATE TABLE DESSERT (
dessertID VARCHAR(10) NOT NULL,
flavourDessert VARCHAR(20) NOT NULL,
dessertName VARCHAR(50) NOT NULL,
dessertPrice DECIMAL(7,2) NOT NULL,
PRIMARY KEY(dessertID)
);
```

Column Name	Datatype	PK	NN	UQ	B	UN	ZF	AI	G	Default/Expression
dessertID	VARCHAR(10)	☑	☑	☐	☐	☐	☐	☐	☐	
flavourDessert	VARCHAR(20)	☐	☑	☐	☐	☐	☐	☐	☐	
dessertName	VARCHAR(50)	☐	☑	☐	☐	☐	☐	☐	☐	
dessertPrice	DECIMAL(7,2)	☐	☑	☐	☐	☐	☐	☐	☐	
		☐	☐	☐	☐	☐	☐	☐	☐	

```
CREATE TABLE CAKE (
dessertID VARCHAR(10) PRIMARY KEY,
cakeWeight FLOAT,
cakeShape VARCHAR(15),
FOREIGN KEY (dessertID) REFERENCES DESSERT(dessertID)
);
```

Column Name	Datatype	PK	NN	UQ	B	UN	ZF	AI	G	Default/Expression
dessertID	VARCHAR(10)	☑	☑	☐	☐	☐	☐	☐	☐	
cakeWeight	DECIMAL(7,2)	☐	☑	☐	☐	☐	☐	☐	☐	
cakeShape	VARCHAR(15)	☐	☑	☐	☐	☐	☐	☐	☐	
		☐	☐	☐	☐	☐	☐	☐	☐	

```
CREATE TABLE PASTRY (
dessertID VARCHAR(10) NOT NULL,
addTopping VARCHAR(30) NOT NULL,
quantityPerBox INT(11) NOT NULL
FOREIGN KEY (dessertID) REFERENCES DESSERT(dessertID)
);
```

Column Name	Datatype	PK	NN	UQ	B	UN	ZF	AI	G	Default/Expression
dessertID	VARCHAR(10)	☑	☑	☐	☐	☐	☐	☐	☐	
addTopping	VARCHAR(30)	☐	☑	☐	☐	☐	☐	☐	☐	
quantityPerBox	INT(11)	☐	☑	☐	☐	☐	☐	☐	☐	
		☐	☐	☐	☐	☐	☐	☐	☐	

7.2 SECTION B: DATA MANIPULATING LANGUAGE (DML)

7.2.1 DISPLAY DATA FOR ALL TABLES

CUSTOMER TABLE

```
SELECT *

FROM CUSTOMER;
```

custID	custName	custEmail	custState	custPostcode	custAddress	custPhoneNo	staffID
C001	NORAIMAN IKRAM	aiman23@gmail.com	KUALA LUMPUR	57899	90-23-44, APARTMENT KAYAMAS	019-2234459	S005
C002	NUR NAJWA	najwy9993@gmail.com	KUALA LUMPUR	57009	FLAT PANGSAPURI SRI HARTAMAS	019-2334242	S002
C003	NUR SABRINA	sabby123@gmail.com	SELANGOR	40016	NO.23, JALAN HANG TUAH	019-2334249	S002
C004	ASJAD HAMIZAN	azjadAzzy20231@gmail.com	SELANGOR	40014	PANGSAPURI SRI BAGAN LUAR	019-8942657	S005
C005	MELISSA SOFIA	mellysa23@gmail.com	NEGERI SEMBILAN	70098	NO.29, JALAN SEMBILU ARA	014-5698756	S004
C006	SHUHADA AMIR	syu123@gmail.com	PAHANG	21000	NO 20, TAMAN EMAS PERAK	012-3342349	S010
C007	SYAKIR IMAN	syakirIman99@gmail.com	KUALA LUMPUR	57000	A-2-14, APARTMENT DESA TASIK	012-3489234	S010
C008	AIDA ZAMANI	aidaZamani88@gmail.com	KELANTAN	15008	89A-2, KONDOMINIUM KAYAMAS	012-3349857	S009
C009	AHMAD AMIRUL NAZIM	mirulAhmd@gmail.com	TERENGGANU	20010	5-4-3, KAMPUNG SELAMAT 2	018-2348560	S007
C010	SITI FATIMAH ALIAS	aliasSitiFat@gmail.com	SELANGOR	40095	90A, TAMAN AMARAH MAWAR	012-9854632	S006
NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

STAFF TABLE

```
SELECT *

FROM STAFF;
```

staffID	staffName	staffNoPhone	staffAddress	staffPosition	staffDOB	staffHireDate
S001	AMIRAH NAJIRAH	019-2344499	FLAT PANGSAPURI SRI DAMAI	MANAGER	1994-02-04	2024-01-01
S002	NAZREEN AIDIL	016-3789024	APARTMENT SRI BAYUMAS	CREW	1999-05-18	2024-01-01
S003	SHAZWANA IMANI	012-3349855	NO.29 JALAN RAJA UDANG	CREW	2000-06-20	2024-01-01
S004	AHMAD ZULFIKRI	018-2234569	NO. 309, JALAN SEROJA 23	CREW	1997-03-22	2024-01-01
S005	NURSHAZ IMANI	012-8942645	BLOK 30-3-2, JALAN RAJA SENGGURUT	CREW	2002-04-27	2024-01-01
S006	JEMAH JAMAL	015-8965544	N0.52 JALAN BATIK TUN SRI LANANG	ASSISTANT MANAGER	1993-04-20	2024-01-02
S007	HARIS ANNUAR	019-2321124	20-A-2, APARTMENT DELIMA	CREW	1998-11-19	2024-01-02
S008	RAFIZI HAKIM	013-2912249	LOT 23, JALAN RAJA IBRAHIM, TAMAN NIAGA	CREW	2000-09-30	2024-01-02
S009	SITI SURAYA	012-2234432	23A-2 TAMAN TUN AMIR	INTERN	2003-04-10	2024-01-02
S010	AIDIL ADHA	012-9112345	NO.35/2, JALAN MAWAR, TAMAN JINJANG	INTERN	2002-01-12	2024-01-02
NULL	NULL	NULL	NULL	NULL	NULL	NULL

ORDER_ITEMS TABLE

```
SELECT *

FROM ORDER_ITEMS;
```

	custID	dessertID	orderDate	quantity	staffID	transactionNo
▶	C001	CA001	2024-01-02	2	S002	R001
	C002	PA001	2024-01-02	1	S002	R002
	C003	CA002	2024-01-03	2	S003	R003
	C003	CA004	2024-01-03	2	S003	R003
	C004	CA007	2024-01-03	2	S006	R004
	C004	PA004	2024-01-03	2	S006	R004
	C005	CA004	2024-01-04	2	S008	R005
	C005	PA005	2024-01-04	2	S004	R006
	C005	PA009	2024-01-04	2	S008	R005
	C006	CA007	2024-01-05	2	S006	R007

PAYMENT TABLE

```
SELECT *

FROM PAYMENT;
```

	transactionNo	paymentDate	paymentMethod
▶	R001	2024-01-05	PAYWAVE
	R002	2024-01-05	CASH
	R003	2024-01-05	PAYWAVE
	R004	2024-01-09	CASH
	R005	2024-01-09	PAYWAVE
	R006	2024-01-11	CASH
	R007	2024-01-11	PAYWAVE
	R008	2024-01-11	PAYWAVE
	R009	2024-01-16	CASH
	R010	2024-01-16	PAYWAVE
	R011	2024-01-18	CASH
*	NULL	NULL	NULL

DESSERT TABLE

```
SELECT *

FROM DESSERT;
```

	dessertID	flavourDessert	dessertName	dessertPrice
▶	CA001	MATCHA	JAPANESE MATCHA CAKE	79.90
	CA002	BLUEBERRY	NIGHTSKY CAKE	80.50
	CA003	HAZELNUT COFFEE	ROASTY CAKE	59.90
	CA004	CHEESE	ALMOND CHEESECAKE	65.20
	CA005	ROCKY ROAD	FRUITY BITES	78.60
	CA006	BUTTERCHIPS	COOKIES BUTTER CHIP	102.20
	CA007	VANILLA	VANILLA CREAM CAKE	90.50
	CA008	BUTTER	CREAM SALTBUTTER CAKE	102.70
	CA009	KIWI	SWEET SOUR KIWI CAKE	85.20
	CA010	BANANA	FLUFFY BANANA CAKE	50.80
	PA001	GREEN TEA	SOFTWAFFLE	20.20

PA002	STRAWBERRY	SPEAK NOW ICE CREAM	10.70
PA003	PISTACHIO	CROMBOLONI	25.80
PA004	CARAMEL	CINAMON GIRL PIE	29.80
PA005	BLUEBERRY	SWEET SERENDIPITY TRE...	15.70
PA006	CREAM CHEESE	BOMBOLONI	30.20
PA007	BANANA	CREPE BANANA	39.90
PA008	CHOCOLATE	WAFFLE	12.50
PA009	MILK	MLIK DONUT	5.00
PA010	RASPBERRY	PIE RASPBERRY	19.90
NULL	NULL	NULL	NULL

CAKE TABLE

```
SELECT *

FROM CAKE;
```

dessertID	cakeWeight	cakeShape
CA001	2.5	RECTANGLE
CA002	1.25	ROUND
CA003	1.05	SQUARE
CA004	1.25	ROUND
CA005	2	ROUND
CA006	2	ROUND
CA007	2.5	ROUND
CA008	1.25	ROUND
CA009	2	SQUARE
CA010	2.5	SQUARE
NULL	NULL	NULL

PASTRY TABLE

```
SELECT *

FROM PASTRY;
```

dessertID	addTopping	quantityPerBox
PA001	CHOCOLATE RICE	6
PA002	CREAM CHEESE	4
PA003	WHIPPING CREAM	3
PA004	RAINBOW RICE CHOCOLATE	5
PA005	ALMOND	5
PA006	GRATED CHEESE	6
PA007	MIX FRUIT	4
PA008	NUTELLA	3
PA009	CORNFLAKES	2
PA010	SUGAR	4
NULL	NULL	NULL

7.2.2 DESCRIBES THE QUERIES

7.2.2.1 RETREVING DATA FROM MULTIPLE TABLE

1. 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
FROM dessert d
JOIN pastry p ON d.dessertID = p.dessertID
```

	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

2. Show the dessert ID, weight of cake and shape of cake for each cake, including information about any associated pastries.

```
SELECT c.dessertID, c.cakeWeight, c.cakeShape
FROM cake c
LEFT JOIN pastry p
ON c.dessertID = p.dessertID;
```

	dessertID	cakeWeight	cakeShape
▶	CA001	2.5	RECTANGLE
	CA002	1.25	ROUND
	CA003	1.05	SQUARE
	CA004	1.25	ROUND
	CA005	2	ROUND
	CA006	2	ROUND
	CA007	2.5	ROUND
	CA008	1.25	ROUND
	CA009	2	SQUARE
	CA010	2.5	SQUARE

3. Retrieve the dessert names, pastry types, and quantity per box for each dessert that has pastries, include dessert without pastries in the result.

```
SELECT d.dessertName, p.addTopping AS Topping,
p.quantityPerBox

FROM dessert d

RIGHT JOIN pastry p

ON d.dessertID = p.dessertID;
```

dessertName	Topping	quantityPerBox
SOFTWAFFLE	CHOCOLATE RICE	6
SPEAK NOW ICE CREAM	CREAM CHEESE	4
CROMBOLONI	WHIPPING CREAM	3
CINAMON GIRL PIE	RAINBOW RICE CHOCOLATE	5
SWEET SERENDIPITY TREATS	ALMOND	5
BOMBOLONI	GRATED CHEESE	6
CREPE BANANA	MIX FRUIT	4
WAFFLE	NUTELLA	3
MLIK DONUT	CORNFLAKES	2
PIE RASPBERRY	SUGAR	4

4. Display the names of staff members, dessert ID and dessert name for each dessert they have served.

```
SELECT s.staffName, oi.dessertID, d.dessertName

FROM staff s

INNER JOIN order_items oi

ON s.staffID = oi.staffID

INNER JOIN dessert d

ON oi.dessertID = d.dessertID;
```

staffName	dessertID	dessertName
NAZREEN AIDIL	CA001	JAPANESE MATCHA CAKE
NAZREEN AIDIL	PA001	SOFTWAFFLE
SHAZWANA IMANI	CA002	NIGHTSKY CAKE
SHAZWANA IMANI	CA004	ALMOND CHEESECAKE
AHMAD ZULFIKRI	PA005	SWEET SERENDIPITY TREATS
AHMAD ZULFIKRI	PA003	CROMBOLONI
AHMAD ZULFIKRI	PA002	SPEAK NOW ICE CREAM
NURSHAZ IMANI	CA001	JAPANESE MATCHA CAKE
NURSHAZ IMANI	PA001	SOFTWAFFLE
JEMAH JAMAL	CA007	VANILLA CREAM CAKE
JEMAH JAMAL	PA004	CINAMON GIRL PIE

5. Display all combinations of dessert names and pastry types.

```
SELECT d.dessertName, p.addTopping
FROM dessert d
CROSS JOIN pastry p;
```

dessertName	addTopping
JAPANESE MATCHA CAKE	CHOCOLATE RICE
JAPANESE MATCHA CAKE	CREAM CHEESE
JAPANESE MATCHA CAKE	WHIPPING CREAM
JAPANESE MATCHA CAKE	RAINBOW RICE CHOCOLATE
JAPANESE MATCHA CAKE	ALMOND
JAPANESE MATCHA CAKE	GRATED CHEESE
JAPANESE MATCHA CAKE	MIX FRUIT
JAPANESE MATCHA CAKE	NUTELLA
JAPANESE MATCHA CAKE	CORNFLAKES
JAPANESE MATCHA CAKE	SUGAR
NIGHTSKY CAKE	CHOCOLATE RICE
NIGHTSKY CAKE	CREAM CHEESE
NIGHTSKY CAKE	WHIPPING CREAM
NIGHTSKY CAKE	RAINBOW RICE CHOCOLATE
NIGHTSKY CAKE	ALMOND
NIGHTSKY CAKE	GRATED CHEESE

7.2.2.2 SIMPLE SQL QUERIES

1. Display order details for dessert ID ‘CA004’ and ‘PA006’. The data that must be shown are dessert ID, transaction number that rename it as order ID, flavour dessert, dessert name, quantity, and dessert price that rename it as price.

```
SELECT oi.dessertID, oi.transactionNo AS orderID,
d.flavourDessert, d.dessertName, oi.quantity,
d.dessertPrice AS price
FROM order_items oi
JOIN dessert d
ON oi.dessertID = d.dessertItemsID
WHERE oi.dessertID IN ('CA004', 'PA006');
```

dessertID	orderID	flavourDessert	dessertName	quantity	price
CA004	R003	CHEESE	ALMOND CHEESECAKE	2	65.20
CA004	R005	CHEESE	ALMOND CHEESECAKE	2	65.20
CA004	R009	CHEESE	ALMOND CHEESECAKE	1	65.20
PA001	R002	GREEN TEA	SOFTWAFFLE	1	20.20
PA001	R008	GREEN TEA	SOFTWAFFLE	2	20.20
PA001	R011	GREEN TEA	SOFTWAFFLE	3	20.20

2. Display the order based on order date which is ‘2024-01-05. The data must be shown is transaction number that renamed it as order ID, dessert ID, flavour name, dessert name, quantity, dessert price and order date.

```
SELECT oi.transactionNo AS orderID, oi.dessertID,
d.flavourDessert, d.dessertName, oi.quantity,
d.dessertPrice, oi.orderDate
FROM order_items oi
JOIN dessert d ON oi.dessertID = d.dessertID
WHERE oi.orderDate = '2024-01-09';
```

orderitemsID	dessertID	flavourDessert	dessertName	quantity	dessertPrice	orderDate
R010	PA002	STRAWBERRY	SPEAK NOW ICE CREAM	2	10.70	2024-01-09
R011	PA001	GREEN TEA	FTWAFFLE	3	20.20	2024-01-09

3. Display the orders where the payment method is ‘CASH’ and staff ID is ‘S004’. The data that needs to be displayed are dessert ID, transaction number that renamed it as order ID, flavour of dessert, name of dessert, quantity, and dessert price that rename it as price.

```
SELECT oi.dessertID, oi.transactionNo AS orderID,
d.flavourDessert, d.dessertName, oi.quantity,
d.dessertPrice AS price
FROM order_items oi
JOIN dessert d ON oi.dessertID = d.dessertID
JOIN payment p ON oi.transactionNo = p.transactionNo
WHERE p.paymentMethod = 'CASH' AND oi.staffID = 'S004';
```

dessertID	orderID	flavourDessert	dessertName	quantity	price
PA005	R006	BLUEBERRY	SWEET SERENDIPITY TREATS	2	15.70
PA003	R009	PISTACHIO	CROMBOLONI	2	25.80

4. Display the information of order item that only customer who ordered for cake and quantity less than TWO. The data that must show are dessert ID, dessert name, 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

5. Display staff ID, staff name, staff position, and staff phone number that who took customer's order between '2024-01-07' and '2024-01-11'.

```
SELECT DISTINCT s.staffID, s.staffName, s.staffPosition,
s.staffNoPhone
FROM staff s
JOIN order_items oi
ON s.staffID = oi.staffID
WHERE oi.orderDate BETWEEN '2024-01-07' AND '2024-01-11';
```

	staffID	staffName	staffPosition	staffNoPhone
▶	S004	AHMAD ZULFIKRI	CREW	018-2234569
	S005	NURSHAZ IMANI	CREW	012-8942645
	S010	AIDIL ADHA	INTERN	012-9112345

7.2.2.3 COLUMN FUNCTION AND GROUPING

1. Display all customer details that make an order on ‘2024-01-09’ by sorting custmer ID into ascending.

```
SELECT *

FROM customer

WHERE custID IN (

      SELECT DISTINCT o.custID

      FROM order_items o

      WHERE o.orderDate = '2024-01-09'

)

      ORDER BY custID ASC;
```

custID	custName	custEmail	custState	custPostcode	custAddress	custPhoneNo	staffID
C009	AHMAD AMIRUL NAZIM	mirulAhmd@gmail.com	TERENGGANU	20010	5-4-3, KAMPUNG SELAMAT 2	018-2348560	S007
C010	SITI FATIMAH ALIAS	aliasSitiFat@gmail.com	SELANGOR	40095	90A, TAMAN AMARAH MAWAR	012-9854632	S006
NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

2. Calculate total sales based on dessert ID and display the dessert ID, dessert name and total sales for each dessert ID.

```
SELECT d.dessertID, d.dessertName, SUM(oi.quantity * de.dessertPrice) AS totalSales

FROM dessert d

JOIN order_items oi

ON d.dessertID = oi.dessertID

JOIN dessert de

ON oi.dessertID = d.dessertID

GROUP BY d.dessertID, d.dessertName
```

	dessertID	dessertName	totalSales
▶	CA001	JAPANESE MATCHA CAKE	4020.80
	CA002	NIGHTSKY CAKE	2010.40
	CA004	ALMOND CHEESECAKE	5026.00
	CA007	VANILLA CREAM CAKE	4020.80
	CA010	FLUFFY BANANA CAKE	1005.20
	PA001	SOFTWAFFLE	6031.20
	PA002	SPEAK NOW ICE CREAM	2010.40
	PA003	CROMBOLONI	2010.40
	PA004	CINAMON GIRL PIE	2010.40
	PA005	SWEET SERENDIPITY TREATS	2010.40
	PA009	MLIK DONUT	2010.40
	PA010	PIE RASPBERRY	1005.20

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

```
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;
```

	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

7.2.2.4 SUBQUERIES

1. Find the staff members that who got the most order in database.

```
SELECT s.staffID, s.staffName, COUNT(oi.custID) AS
numOrders

FROM staff s

JOIN order_items oi

ON s.staffID = oi.staffID

GROUP BY s.staffID, s.staffName

HAVING numOrders = (

      SELECT COUNT(custID)

      FROM order_items

      GROUP BY staffID

      LIMIT 1

);
```

	staffID	staffName	numOrders
▶	S002	NAZREEN AIDIL	2
	S003	SHAZWANA IMANI	2
	S005	NURSHAZ IMANI	2
	S008	RAFIZI HAKIM	2
	S010	AIDIL ADHA	2

2. Identify the five most popular dessert flavours and 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
VANILLA	4	2	362.00
MATCHA	4	2	319.60
BLUEBERRY	4	2	192.40

3. Each of the staff receives 5% commission for every sales they handle. Calculate the commission each staff earns and sort the commission in descending order for each date.

```
SELECT s.staffID, s.staffName, s.staffPosition,
oi.orderDate, ROUND(SUM(d.dessertPrice * oi.quantity) *
0.05, 2) AS commissionEarned

FROM staff s

LEFT JOIN order_items oi

ON s.staffID = oi.staffID

LEFT JOIN dessert d

ON oi.dessertID = d.dessertID

WHERE oi.transactionNo IN (

    SELECT DISTINCT transactionNo

    FROM payment

) OR oi.transactionNo IS NULL

GROUP BY s.staffID, s.staffName, s.staffPosition,
oi.orderDate

ORDER BY oi.orderDate ASC, commissionEarned DESC;
```

staffID	staffName	staffPosition	orderDate	commissionEarned
S001	AMIRAH NAJIRAH	MANAGER	NULL	NULL
S007	HARIS ANNUAR	CREW	NULL	NULL
S002	NAZREEN AIDIL	CREW	2024-01-02	9.00
S003	SHAZWANA IMANI	CREW	2024-01-03	14.57
S006	JEMAH JAMAL	ASSISTANT MANAGER	2024-01-03	12.03
S008	RAFIZI HAKIM	CREW	2024-01-04	7.02
S004	AHMAD ZULFIKRI	CREW	2024-01-04	1.57
S006	JEMAH JAMAL	ASSISTANT MANAGER	2024-01-05	9.05
S009	SITI SURAYA	INTERN	2024-01-05	1.00
S005	NURSHAZ IMANI	CREW	2024-01-06	7.99
S006	JEMAH JAMAL	ASSISTANT MANAGER	2024-01-06	2.02
S010	AIDIL ADHA	INTERN	2024-01-08	5.80
S004	AHMAD ZULFIKRI	CREW	2024-01-08	2.58
S005	NURSHAZ IMANI	CREW	2024-01-09	3.03
S004	AHMAD ZULFIKRI	CREW	2024-01-09	1.07

8.0 CONCLUSION

The database system used by Dato's Bakery Shop has a complex relational structure with tables specially created for different preparing process. These tables, which contain components such as orders, customers, and cakes, are linked together by means of foreign key constrains, guaranteeing the accuracy and coherence of the data relationships. By linking staff and customer tables, the staff ID property acts as a crucial connection that promotes accountability and increases organizational transparency.

In addition, the system presents its features with well selected sample data in each table, offering useful insight into how it works. By strategically using indexes on columns, further efficiency is added, speeding up query processing and enhancing the overall efficacy of the bakery's operation. Dato Bakery Shop is ideally positioned to handle the challenges of a cutthroat industry thanks to its relational strategy, which highlights a dedication to data integrity, accountability, and maximized performance.

9.0 REFERENCES

Book:

- J. Hernandez M. (2014). Database Design for Mere Mortals 3rd ed.: A Hands-On Guide to Relational Database Design. Addison-Wesley.
- R. Groff J., N. Weinberg P, & J. Oppel A. (2010). The Complete Reference SQL 3rd ed. McGraw Hill Companies.
- Deutsch D., Shaw P., Melton J. & Gallagher L. (1991). Database Management Systems: Understanding and Applying Database Technology. Butterworth-Heinemann Ltd.
- Coronel C. & Morris S. (2016). Database System: Design, Implementation, and Management. 12th. ed. Course Technology, CENGAGE Learning.

Interview:

- Sir Nurshaz Naziel bin Ammar, personal communication, October 1st, 2023.