FACULTY OF COMPUTING
UTM Johor Bahru

SECD2523 - DATABASE SEMESTER 1 2023/2024

Phase 3 (P3) –

Database Logical Design & SQL

Group Name : Teh O Limau Ais

Members:

LIM SI NI A22EC0070

CHAN QING YEE A22EC0040

ONG KAI XUEN A22EC0100

OOI WEI SIAN A22EC0102

Stakeholder: getmehired.io

Table of Contents

1.0 Introduction

GetMeHired is a job search assistance platform that provides a variety of services to help job seekers find their ideal job. One of the services offered by GetMeHired is CV template distribution. However, the current process of manually distributing CV templates is time-consuming, error-prone and inefficient. Additionally, GetMeHired's current customer service system often has a late response in helping customers resolve customer issues. These delays can be caused by various factors such as a high volume of customer requests, limited staff availability or inefficient communication processes. As a result, customers may have to wait extended periods for assistance, which will lead the customer to feel frustrated and dissatisfied.

To address these problems, we propose the development of an automated CV template distribution system with an integrated AI-powered chatbot. This system will automatically send the CV template to customers upon payment. Through this method, prompt delivery will be guaranteed and manual intervention will be eliminated. The AI-powered chatbot will provide 24/7 customer support, which allows customers to receive immediate assistance whenever they need it. This AI chatbot can handle common customer inquiries and provide personalized responses, which reduces the burden on human support staff and improves the overall customer experience.

The automated CV template distribution system with an AI-powered chatbot will rely on the database to store and retrieve information about customers, CV templates, FAQ info, payment info and so on. The system will use this information to automatically deliver CV templates and also provide prompt customer support.

By implementing an automated CV template distribution system with an AI-powered chatbot, GetMeHired can improve its efficiency, reduce errors made by humans, enhance satisfaction and provide a responsive customer service experience. The system will also help GetMeHired to maintain its competitive edge in the job search assistance platform.

2.0 Overview of project

In Phase 3 of the project, we focus on logical database design and SQL implementation. Before going to the actual database design process, it is important to have a well understanding of the business rules. In this phase, we updated the business rules, and refining both the conceptual ERD and the enhanced ERD.

Once the conceptual and enhanced ERDs are updated, the project progresses to the logical database design stage. Here, we design the logical ERD to ensure a well-structured representation of data entities and their relationships. Simultaneously, the data dictionary also undergoes an update. We also applied the normalization to the logical design. This process refines the database structure. After performing normalization, we produce the relational database schema based on the normalization.

Finally, we produce the result by writing SQL statements. Data Definition Language (DDL) and Data Manipulation Language (DML) are being used in the SQL. DDL statements are used to construct tables, define indexes, and enforce constraints, while DML statements facilitate the manipulation and retrieval of data.

3.0 Database conceptual design

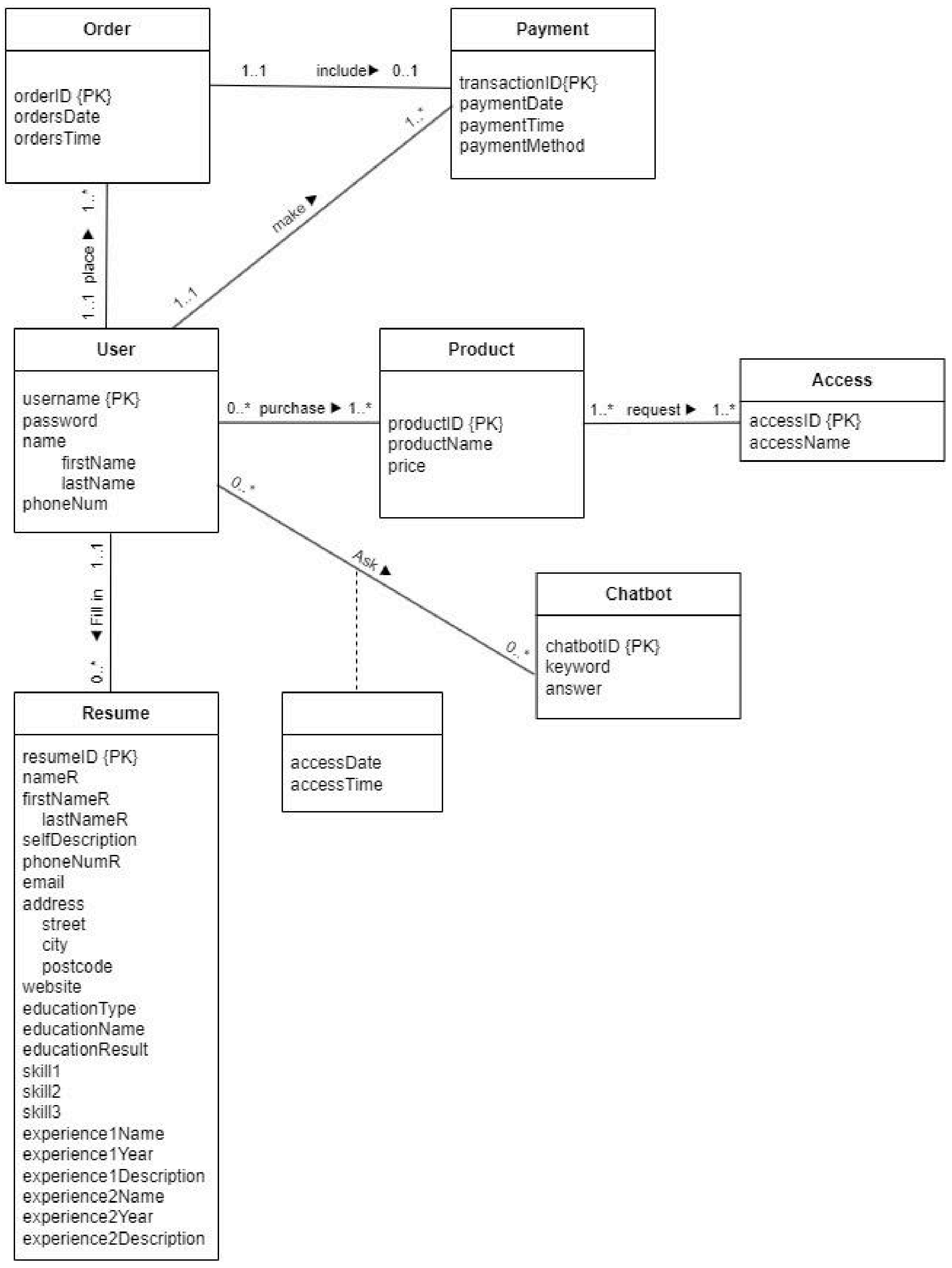
3.1 Updated business rule

1. The order can be placed by only one user
2. One user can place one or more orders
3. The payment can be made by only one user
4. One user can place one or more payments
5. One order can include a maximum one payment
6. One payment is included in only one order
7. The product can be purchased by zero or more users
8. The user can purchase one or more product
9. The product can request one or more access
10. The access can be requested by one or more product
11. The user can ask zero or more chatbots
12. The chatbot can be asked by zero or many users
13. The user can fill in zero or more resumes
14. The resume template can be filled in only one user

3.2 Conceptual ERD

3.2.1 Conceptual ERD

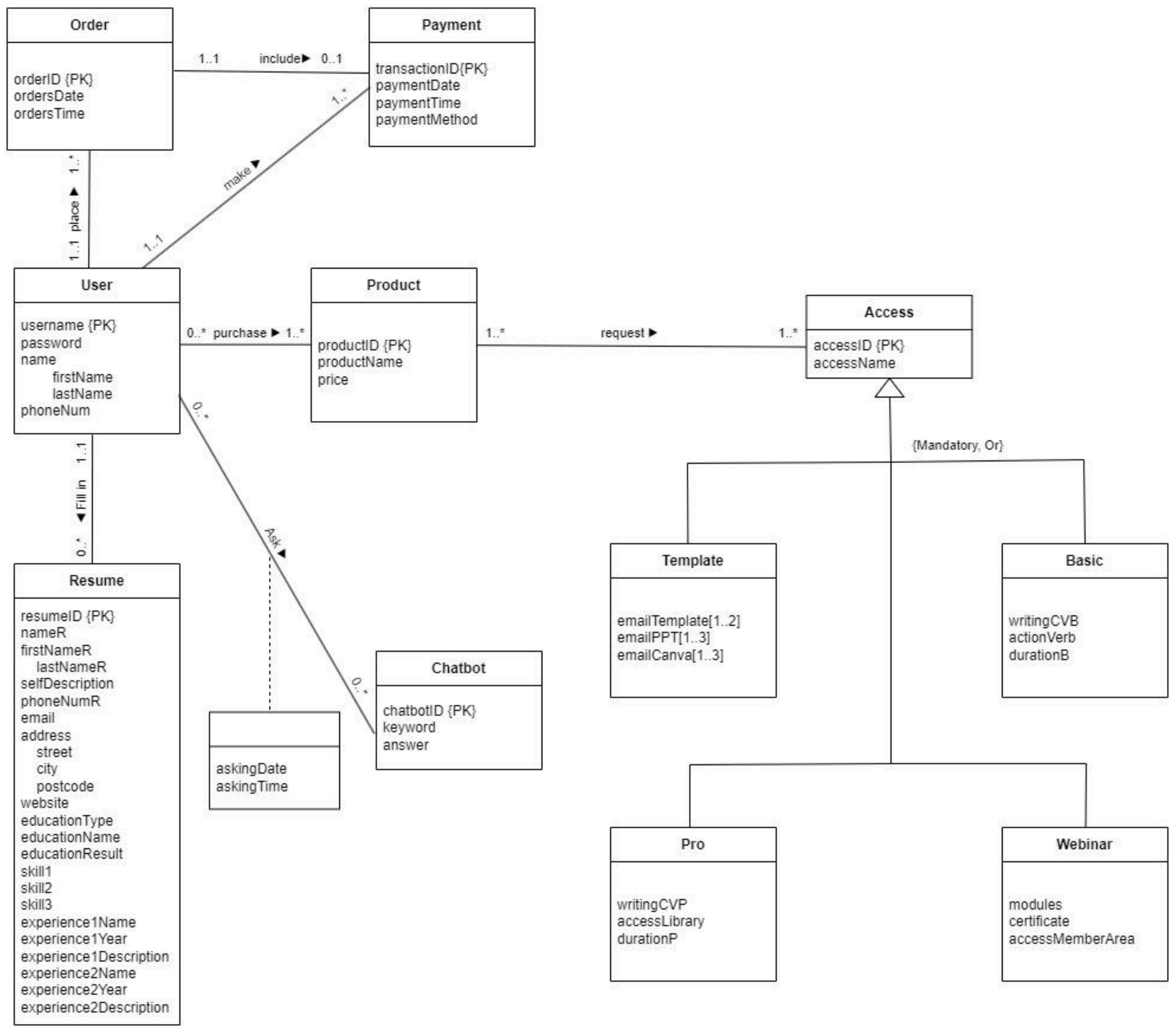
Drawio Link:
https://drive.google.com/file/d/1rWoqItRKIy7b9ICvu1bQ3ax5c2DDOQ1X/view?usp=sharing



3.2.2 Enhanced ERD (EERD)

Drawio Link:

https://drive.google.com/file/d/1rWoqItRKIy7b9ICvu1bQ3ax5c2DDOQ1X/view?usp=sharing



4.0 DB logical design

4.1 Logical ERD

Derive Relations

1. Strong entity types for User and Resume
    - For composite attributes, includes only constituent simple attributes

2. Weak Entity
    - Create a relation that includes all simple attributes of the entity and derive fully primary key from owner entity

3. One-to-many (1:*) binary relationship types
    - exists between User (Parent) and Order (Child)
    - exists between User (Parent) and Payment (Child)
    - exists between User (Parent) and Resume (Child)
    - Copy the primary key of parent entity to child entity as a foreign key

4. One-to-one (1:1) binary relationship types
    - exits between Order and Payment
    - Mandatory participation on Order (Parent) side and optional participation on Payment (Child) side
    - Copy the primary key of parent entity to child entity as a foreign key

5. Many-to-many (*:*) binary relationship types
    - exists between User and Product
    - exists between Product and Access
    - exists between User and Chatbot
    - Transform *;* relationship by creating two 1:* relationships
    - create a new relation and copy the primary keys of both entities into it as foreign keys

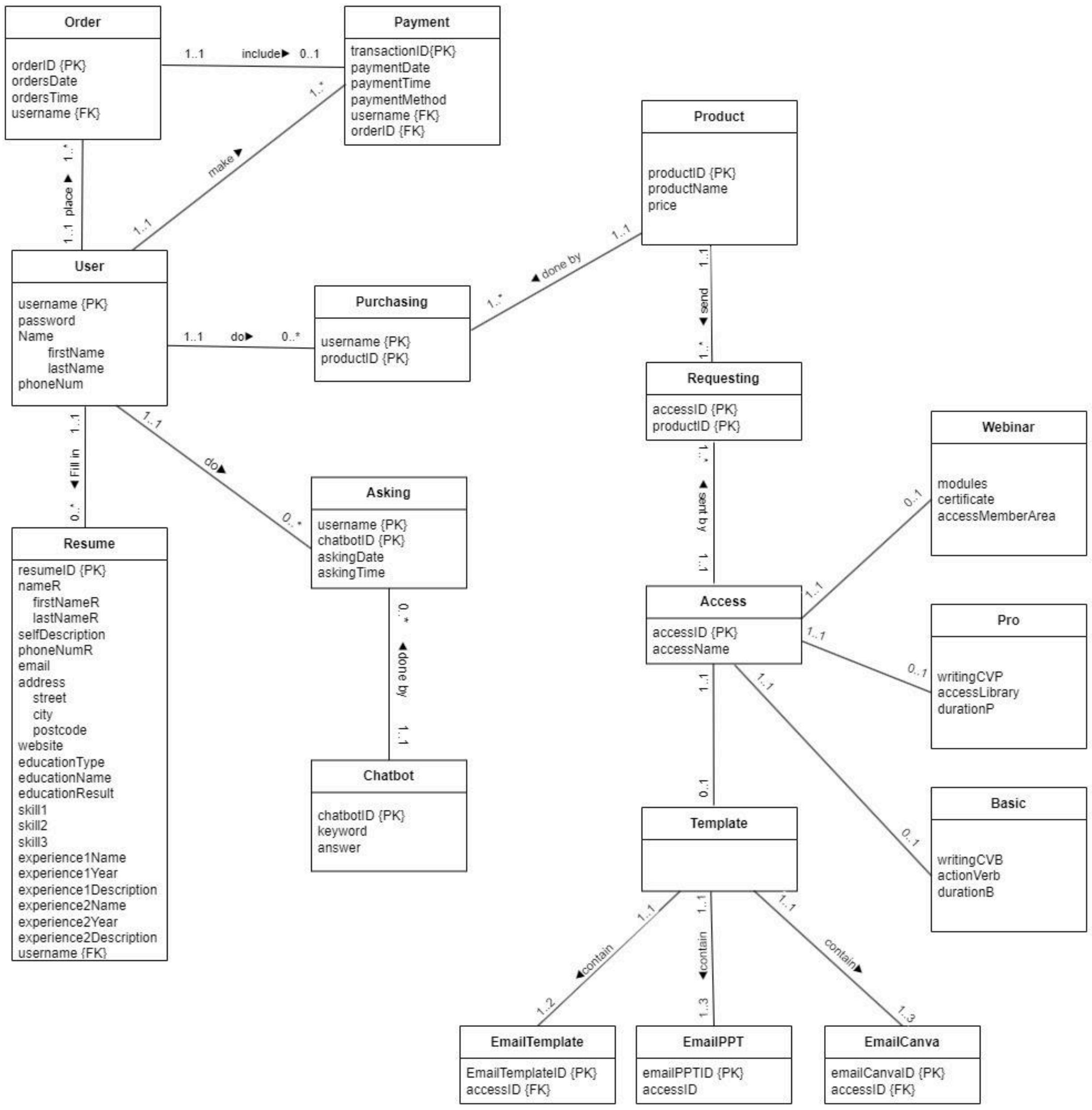
6. Superclass/subclass relationship types
    - Since the relationship is {Mandatory, Or}, create one relation for each combined superclass/subclass

7. Multi-valued attribute
    - There are 3 multi-valued attributes in Template entity
    - Create a relation to represent the multi-valued attribute and post a copy of primary key of the owner entity into the new relation to act as a foreign key

Logical ERD Diagram

Drawio Link:

https://drive.google.com/file/d/1rWoqItRKIy7b9ICvu1bQ3ax5c2DDOQ1X/view?usp=sharing



4.2 Updated Data Dictionary

4.2.1 Description of Entity

Entity	**Description**	**Occurrence**
User	Hold customer's information	-User login the getmehired system -User access their account -User provide information to system -User ask question to system -User purchase CV template -User get the CV template package
Order	Hold information of customer order	-Order has placed by user after they made payment
Resume	Hold information of customer's resume details	-User fill in the resume details after they made payment
Payment	Hold information of user payment	-Each payment made by one customer
Product	Hold customer purchased product information	-Customer purchase the CV template package, the product is generated
Access	Hold the access information when customer made payment	-Customer access the CV template after they made payment
Chatbot	Hold the information of user queries	-Chatbot pop out the prepared answers after user send queries questions
Purchasing	Hold the purchasing information when customer made payment	- Customer purchase the CV template package, the purchasing details is generated
Asking	Hold the username and time information when user makes query	- User makes query to the chatbot
Requesting	Hold the accessID and productID when customer make request for CV template	- User makes request after purchasing the product

Webinar	Hold the information related to webinar events	-User access to webinar package after making the purchase
Pro	Hold the information of the package product that include action verbs and a full writing CV	-User access to Pro package after making the purchase
Basic	Hold the information of the package product that include full writingCV and action verbs	-User access to Basic package after making the purchase
Template	Hold the information related to template events	-User access to template package after making the purchase
EmailTemplate	Hold the information of template	-User gets the CV Template after making request
EmailPPT	Hold the information of PPT template	-User gets the PPT CV Template after making request
EmailCanva	Hold the information of Canva template	-User gets the Canva CV Template after making request

4.2.2 Description of Relationship

Entity	Multiplicity	Relationship	Multiplicity	Entity
User	1..1	place	1..*	Order
	1..1	make	1..*	Payment
	1..1	do	0..*	Purchasing
	1..1	do	0..*	Asking
	1..1	Fill in	0..*	Resume
Order	1..1	include	0..1	Payment
Product	1..1	done by	1..*	Purchasing
	1..1	send	1..*	Requesting
Chatbot	1..1	done by	0..*	Asking
Access	1..1	sent by	1..*	Requesting
	1..1		0..1	Webinar
	1..1		0..1	Pro
	1..1		0..1	Basic
	1..1		0..1	Template
Template	1..1	contain	1..2	EmailTemplate
	1..1	contain	1..3	EmailPPT
	1..1	contain	1..3	EmailCanva

4.2.3 Description of Attributes

Entity	Attributes	Description	Data Type	Null	Multi-valued
Product	productID	Uniquely identify a product ID (PK)	VARCHAR2(10)	No	No
	price	product price	NUMBER(5,2)	No	No
	productName	name of the product	VARCHAR2(15)	No	No
Access	accessID	Uniquely identify an access ID (PK)	VARCHAR2(10)	No	No
	accessName	Name of the access	VARCHAR2(30)	No	No
Template	accessID	Primary key that reference to the accessID of Access entity	VARCHAR2(10)	No	No
Basic	accessID	Primary key that reference to the accessID of Access entity	VARCHAR2(10)	No	No
	writingCVB	trigger the CV writing service	VARCHAR2(30)	No	No
	actionVerb	powerful action verbs on cv	VARCHAR2(20)	No	No
	durationB	duration of basic CV writing service	NUMBER(2)	No	No
Pro	accessID	Primary key that reference to the accessID of Access entity	VARCHAR2(10)	No	No
	writingCVP	trigger the CV writing service	VARCHAR2(30)	No	No
	accessLibrary	full access library of CV service	VARCHAR2(10)	No	No
	durationP	duration of pro CV writing service	NUMBER(2)	No	No
Webinar	accessID	Primary key that reference to the accessID of Access entity	VARCHAR2(10)	No	No
	modules	cover 4 modules of webinar	VARCHAR2(20)	No	No
	certificate	certificate of completion of webinar	VARCHAR2(20)	No	No
	accessMemberArea	full access member for whole CV service	VARCHAR2(20)	No	No
User	username	Uniquely identify a username (PK)	VARCHAR2(10)	No	No
	password	user account login password	VARCHAR2(30)	No	No
	Name firstName	Name of customer in system First name of customer in system	VARCHAR2(15)	No	

	lastName	Last name of customer in system	VARCHAR2(15)	No	
	phoneNum	phone number of customer	VARCHAR2(20)	No	No
Chatbot	chatbotID	Uniquely identify a chatbot (PK)	VARCHAR2(10)	No	No
	keyword	a keyword that triggered the answer	VARCHAR2(30)	No	No
	answer	corresponding answer is pop out after the keyword is triggered	VARCHAR2(80)	No	No
Order	orderID	Uniquely identify an order (PK)	VARCHAR2(10)	No	No
	username	Foreign key of User which uniquely identify an username (FK)	VARCHAR2(10)	No	No
	ordersDate	date of order	DATE	No	No
	ordersTime	time of order	TIMESTAMP	No	No
Payment	transactionID	Uniquely identify a transaction ID (PK)	VARCHAR2(10)	No	No
	paymentDate	date of order	DATE	No	No
	paymentTime	time of order	TIMESTAMP	No	No
	paymentMethod	user online payment method	VARCHAR2(20)	No	No
	username	Foreign key of User which uniquely identify an username (FK)	VARCHAR2(10)	No	No
	orderID	Foreign key of Order which uniquely identify an order id (FK)	VARCHAR2(10)	No	No
Resume	resumeID	Uniquely identify a resume id (PK)	VARCHAR2(10)	No	No
	nameR firstNameR lastNameR	 First name of customer filled in resume Last name of customer filled in resume	 VARCHAR2(15) VARCHAR2(15)	No No No	No No No
	selfDescription	Self description of customer filled in resume	VARCHAR2(100)	No	No
	phoneNumR	phone number of customer filled in resume	VARCHAR2(20)	No	No
	email	email of customer filled in resume	VARCHAR2(30)	No	No

	address
	street
		address of customer filled in resume
		street of customer filled in resume
			VARCHAR2(30)
			VARCHAR2(15)
				No
				No
					No
					No

	city postcode	city of customer filled in resume postcode of customer filled in resume	VARCHAR2(15) VARCHAR2(15)	No No	No No
	website	customer's profile website	VARCHAR2(30)	No	No
	educationType educationName educationResult	customer's education type customer's education name customer's education result	VARCHAR2(30) VARCHAR2(30) VARCHAR2(30)	No No No	No No No
	skill1 skill2 skill3	customer's first skill customer's second skill customer's third skill	VARCHAR2(30) VARCHAR2(30) VARCHAR2(30)	No Yes Yes	No No No

	experience1Name
	experience1Year
	experience1Description
	experience2Name
	experience2Year
	experience2Description
		customer's first experience name
		customer's first experience year
		customer's first experience description
		customer's second experience name
		customer's second experience year
		customer's second experience description
			VARCHAR2(30)
			NUMBER(4)
			VARCHAR2(30)
			VARCHAR2(30)
			NUMBER(4)
			VARCHAR2(30)
				No
				No
				No
				Yes
				Yes
				Yes
					No
					No
					No
					No
					No
					No

	username	Foreign key of User which uniquely identify an username (FK)	VARCHAR2(10)	No	No
Asking	username	Foreign key of User which uniquely identify an username (FK), combine with Asking(chatbotID) as composite primary key	VARCHAR2(10)	No	No
	chatbotID	Foreign key of Chatbot which uniquely identify a chatbot ID (FK), combine with Asking(username) as composite primary key	VARCHAR2(30)	No	No
	askingDate	date of order	DATE	No	No
	askingTime	time of order	TIMESTAMP	No	No
EmailTemplate	EmailTemplateID	Uniquely identify an email template ID (PK)	VARCHAR2(10)	No	No
	accessID	Foreign key of Template which uniquely identify an access ID (FK)	VARCHAR2(10)	No	No
EmailPPT	EmailPPTID	Uniquely identify an email PPT ID (PK)	VARCHAR2(10)	No	No
	accessID	Foreign key of Template which uniquely identify an access ID (FK)	VARCHAR2(10)	No	No

EmailCanva	emailCanvaID	Uniquely identify an email PPT ID (PK)	VARCHAR2(10)	No	No
	accessID	Foreign key of Template which uniquely identify an access ID (FK)	VARCHAR2(10)	No	No
Purchasing	username	Foreign key of User which uniquely identify a username (FK), combine with Purchasing(productID) as composite primary key	VARCHAR2(10)	No	No
	productID	Foreign key of Product which uniquely identify a product ID (FK), combine with Purchasing(username) as composite primary key	VARCHAR2(10)	No	No
Requesting	productID	Foreign key of Product which uniquely identify a product ID (FK), combine with Requesting(accessID) as composite primary key	VARCHAR2(10)	No	No
	accessID	Foreign key of Template which uniquely identify an access ID (FK), combine with Requesting(productID) as composite primary key	VARCHAR2(10)	No	No

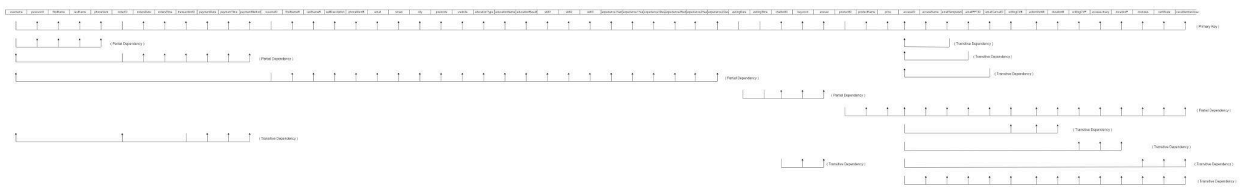
4.3 Normalization

Since we had declared a unique ID for each entity as their primary key, therefore there only exists full functional dependency in our database design. Since it does not exist partial dependency and transitive dependency so that the relation schema for 2NF and 3NF remain same with 1NF.

Dependency Diagram

Drawio Link:

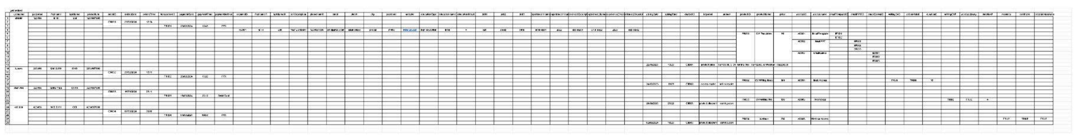
https://drive.google.com/file/d/1wpk4RSg6ZnQ4QukzNCcQJ8-kqRs1fDhK/view?usp=sharing



Google Sheets Link [UNF until 3NF]:

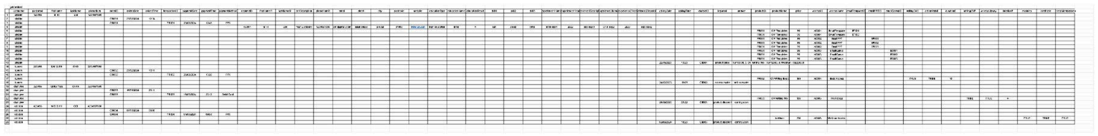
https://docs.google.com/spreadsheets/d/1IzqfSs0SBVQmozqSq968LZQu_2Nl1eRSrEN2AgtSPO0/edit?usp=sharing

UNF



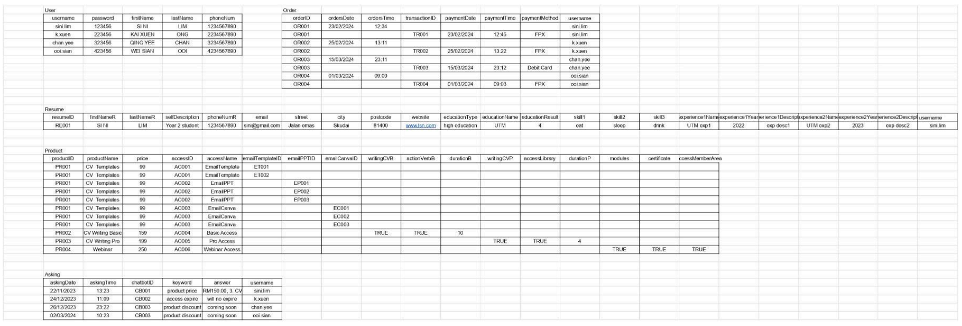
1NF

- Eliminate repeating groups by creating separate tables for them.



2NF

- Removing Partial Dependency



User

username	password	firstName	lastName	phoneNum
sini.lim	123456	SI NI	LIM	1234567890
k.xuen	223456	KAI XUEN	ONG	2234567890
chan.yee	323456	QING YEE	CHAN	3234567890
ooi.sian	423456	WEI SIAN	OOI	4234567890

Order

orderID	ordersDate	ordersTime	transactionID	paymentDate	paymentTime	paymentMethod	username
OR001	23/02/2024	12:34					sini.lim
OR001			TR001	23/02/2024	12:45	FPX	sini.lim
OR002	25/02/2024	13:11					k.xuen
OR002			TR002	25/02/2024	13:22	FPX	k.xuen
OR003	15/03/2024	23:11					chan.yee
OR003			TR003	15/03/2024	23:12	Debit Card	chan.yee
OR004	01/03/2024	09:00					ooi.sian
OR004			TR004	01/03/2024	09:03	FPX	ooi.sian

Resume

resumeID	firstNameR	lastNameR	selfDescription	phoneNumR	email	street	city	postcode	website	educationType	educationName	educationResult	skill1	skill2	skill3	xperience1Nam	experience1Year	erience1Descript	xperience2Nam	experience2Year	erience2Descript	username
RE001	SI NI	LIM	Year 2 student	1234567890	sini@gmail.com	Jalan emas	Skudai	81400	www.lsn.com	high education	UTM	4	eat	sleep	drink	UTM exp1	2022	exp desc1	UTM exp2	2023	exp desc2	sini.lim

Product

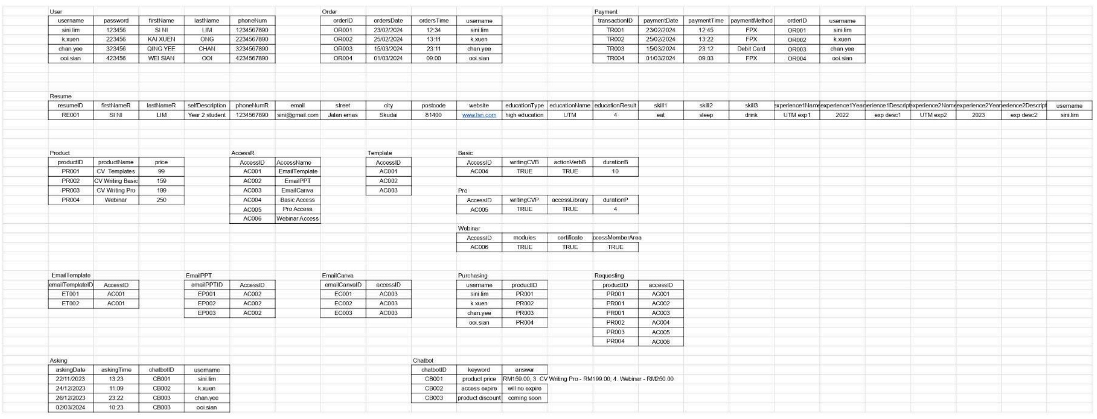
productID	productName	price	accessID	accessName	emailTemplateID	emailPPTID	emailCanvaID	writingCVB	actionVerbB	durationB	writingCVP	accessLibrary	durationP	modules	certificate	ccessMemberArea
PR001	CV Templates	99	AC001	EmailTemplate	ET001											
PR001	CV Templates	99	AC001	EmailTemplate	ET002											
PR001	CV Templates	99	AC002	EmailPPT		EP001										
PR001	CV Templates	99	AC002	EmailPPT		EP002										
PR001	CV Templates	99	AC002	EmailPPT		EP003										
PR001	CV Templates	99	AC003	EmailCanva			EC001									
PR001	CV Templates	99	AC003	EmailCanva			EC002									
PR001	CV Templates	99	AC003	EmailCanva			EC003									
PR002	CV Writing Basic	159	AC004	Basic Access				TRUE	TRUE	10						
PR003	CV Writing Pro	199	AC005	Pro Access							TRUE	TRUE	4			
PR004	Webinar	250	AC006	Webinar Access										TRUE	TRUE	TRUE

Asking

askingDate	askingTime	chatbotID	keyword	answer	username
22/11/2023	13:23	CB001	product price	RM159.00, 3. CV	sini.lim
24/12/2023	11:09	CB002	access expire	will no expire	k.xuen
26/12/2023	23:22	CB003	product discount	coming soon	chan.yee
02/03/2024	10:23	CB003	product discount	coming soon	ooi.sian

3NF

- Removing Transitive Dependency



User

username	password	firstName	lastName	phoneNum
sini.lim	123456	SI NI	LIM	1234567890
k.xuen	223456	KAI XUEN	ONG	2234567890
chan.yee	323456	QING YEE	CHAN	3234567890
ooi.sian	423456	WEI SIAN	OOI	4234567890

Order

orderID	ordersDate	ordersTime	username
OR001	23/02/2024	12:34	sini.lim
OR002	25/02/2024	13:11	k.xuen
OR003	15/03/2024	23:11	chan.yee
OR004	01/03/2024	09:00	ooi.sian

Payment

transactionID	paymentDate	paymentTime	paymentMethod	orderID	username
TR001	23/02/2024	12:45	FPX	OR001	sini.lim
TR002	25/02/2024	13:22	FPX	OR002	k.xuen
TR003	15/03/2024	23:12	Debit Card	OR003	chan.yee
TR004	01/03/2024	09:03	FPX	OR004	ooi.sian

Resume

resumeID	firstNameR	lastNameR	selfDescription	phoneNumR	email	street	city	postcode	website	educationType	educationName	educationResult	skill1	skill2	skill3	xperience1Nam	experience1Year	erience1Descript	xperience2Nam	experience2Year	erience2Descript	username
RE001	SI NI	LIM	Year 2 student	1234567890	sini@gmail.com	Jalan emas	Skudai	81400	www.lsn.com	high education	UTM	4	eat	sleep	drink	UTM exp1	2022	exp desc1	UTM exp2	2023	exp desc2	sini.lim

Product

productID	productName	price
PR001	CV Templates	99
PR002	CV Writing Basic	159
PR003	CV Writing Pro	199
PR004	Webinar	250

AccessR

AccessID	AccessName
AC001	EmailTemplate
AC002	EmailPPT
AC003	EmailCanva
AC004	Basic Access
AC005	Pro Access
AC006	Webinar Access

Template

AccessID
AC001
AC002
AC003

Basic

AccessID	writingCVB	actionVerbB	durationB
AC004	TRUE	TRUE	10

Pro

AccessID	writingCVP	accessLibrary	durationP
AC005	TRUE	TRUE	4

Webinar

AccessID	modules	certificate	ccessMemberArea
AC006	TRUE	TRUE	TRUE

EmailTemplate

emailTemplateID	AccessID
ET001	AC001
ET002	AC001

EmailPPT

emailPPTID	AccessID
EP001	AC002
EP002	AC002
EP003	AC002

EmailCanva

emailCanvaID	accessID
EC001	AC003
EC002	AC003
EC003	AC003

Purchasing

username	productID
sini.lim	PR001
k.xuen	PR002
chan.yee	PR003
ooi.sian	PR004

Requesting

productID	accessID
PR001	AC001
PR001	AC002
PR001	AC003
PR002	AC004
PR003	AC005
PR004	AC006

Asking

askingDate	askingTime	chatbotID	username
22/11/2023	13:23	CB001	sini.lim
24/12/2023	11:09	CB002	k.xuen
26/12/2023	23:22	CB003	chan.yee
02/03/2024	10:23	CB003	ooi.sian

Chatbot

chatbotID	keyword	answer
CB001	product price	RM159.00, 3. CV Writing Pro - RM199.00, 4. Webinar - RM250.00
CB002	access expire	will no expire
CB003	product discount	coming soon

5.0 Relational DB Schemas (after normalization)

1. User(<u>username</u>, password, firstName, lastName, phoneNum)
   PK: User(username)

2. Order(<u>orderID</u>, ordersDate, ordersTime, username)
   PK: Order(orderID)
   FK: username reference User(username)

3. Payment(<u>transactionID</u>, paymentDate, paymentTime, paymentMethod, username, orderID)
   PK: Payment(transactionID)
   FK: orderID reference Order(orderID)
   username reference User(username)

4. Purchasing(<u>username, productID</u>)
   PK: Purchasing(username, productID)
   FK: Product reference Order(productID)
   username reference User(username)

5. Product(<u>productID</u>, productName, price)
   PK: Product(productID)

6. Resume(<u>resumeID</u>, firstNameR, lastNameR, selfDescription, phoneNum, email, address, street, city, postcode, website, educationType, educationName, educationResult, skill1, skill2, skill3, experience1Name, experience1Year, experience1Description, experience2Name, experience2Year, experience2Description, username)
   PK: Resume(resumeID)
   FK: username reference User(username)

7. Requesting(<u>productID, accessID</u>)
   PK: Requesting(productID, accessID)
   FK: productID reference Product(productID)
   accessID reference Template(accessID) or Basic(accessID) or Pro(accessID) or Webinar(accessID)

8. Access(accessID, accessName)
   PK: Access(accessID)

9. Template(accessID)
   PK: Template(accessID)
   FK: accessID references Access(accessID)

10. EmailTemplate(EmailTemplateID, accessID)
    PK: EmailTemplate(EmailTemplateID)
    FK: accessID references Template(accessID)

11. EmailPPT(EmailPPTID, accessID)
    PK: EmailPPT(EmailPPTID)
    FK: accessID references Template(accessID)

12. EmailCanva(emailCanvaID, accessID)
    PK: EmailCanva(emailCanvaID)
    FK: accessID references Template(accessID)

13. Basic(accessID, writingCVB, actionVerb, durationB)
    PK: Basic(accessID)
    FK: accessID references Access(accessID)

14. Pro(accessID, writingCVP, accessLibrary, durationP)
    PK: Pro(accessID)
    FK: accessID references Access(accessID)

15. Webinar(accessID, modules, certificate, accessMemberArea)
    PK: Webinar(accessID)
    FK: accessID references Access(accessID)

16. Asking(username, chatbotID, askingDate, askingTime)
    PK: Asking(username, chatbotID)
    FK: User(username)
    Chatbot(chatbotID)

17. Chatbot(chatbotID, keyword, answer)

    PK: Chatbot(chatbotID)

6.0 SQL Statements (DDL & DML)

6.1 Interface Design (Figma)

https://www.figma.com/file/p2qvXUXbZCsjkGOGYPC7EJ/SAD-Phase-4?type=design&node-id=0%3A1&mode=design&t=A4AaHLsU6dHMz8f2-1

6.2 DDL - Create all the table

```
CREATE TABLE Product(
productID VARCHAR2(10) PRIMARY KEY,
price NUMBER(5,2) NOT NULL,
productName VARCHAR2(15) NOT NULL
);

CREATE TABLE AccessR (
accessID VARCHAR2(10) PRIMARY KEY,
accessName VARCHAR2(30) NOT NULL
);

CREATE TABLE Template(
accessID VARCHAR2(10) PRIMARY KEY,
CONSTRAINT fk_access_template FOREIGN KEY (accessID) REFERENCES
AccessR(accessID)
);

CREATE TABLE Basic (
accessID VARCHAR2(10) PRIMARY KEY,
writingCVB VARCHAR2(30) NOT NULL,
actionVerb VARCHAR2(20) NOT NULL,
durationB NUMBER(2) NOT NULL,
CONSTRAINT fk_access_basic FOREIGN KEY (accessID) REFERENCES
AccessR(accessID)
);

CREATE TABLE Pro (
accessID VARCHAR2(10) PRIMARY KEY,
writingCVP VARCHAR2(30) NOT NULL,
accessLibrary VARCHAR2(10) NOT NULL,
durationP NUMBER(2) NOT NULL,
CONSTRAINT fk_access_pro FOREIGN KEY (accessID) REFERENCES
AccessR(accessID)
);

CREATE TABLE Webinar (
accessID VARCHAR2(10) PRIMARY KEY,
modules VARCHAR2(20) NOT NULL,
certificate VARCHAR2(20) NOT NULL,
accessMemberArea VARCHAR2(20) NOT NULL,
CONSTRAINT fk_access_webinar FOREIGN KEY (accessID) REFERENCES
AccessR(accessID)
);
```

```
CREATE TABLE Users(
username VARCHAR2(10) PRIMARY KEY,
password VARCHAR2(30) NOT NULL,
firstName VARCHAR2(15) NOT NULL,
lastName VARCHAR2(15) NOT NULL,
phoneNum VARCHAR2(20) NOT NULL
);

CREATE TABLE Chatbot (
chatbotID VARCHAR2(10) PRIMARY KEY,
keyword VARCHAR2(30) NOT NULL,
answer VARCHAR2(200) NOT NULL
);


CREATE TABLE Orders (
orderID VARCHAR2(10) PRIMARY KEY,
username VARCHAR2(10) NOT NULL,
ordersDate DATE NOT NULL,
ordersTime TIMESTAMP NOT NULL,
CONSTRAINT fk_order_username FOREIGN KEY (username) REFERENCES
Users(username)
);


CREATE TABLE Payment (
transactionID VARCHAR2(10) PRIMARY KEY,
paymentDate DATE NOT NULL,
paymentTime TIMESTAMP NOT NULL,
paymentMethod VARCHAR2(20) NOT NULL,
username VARCHAR2(10) NOT NULL,
orderID VARCHAR2(10) NOT NULL,
CONSTRAINT fk_pay_username FOREIGN KEY (username) REFERENCES
Users(username),
CONSTRAINT fk_pay_order FOREIGN KEY (orderID) REFERENCES
Orders(orderID)

);

CREATE TABLE Resume (
resumeID VARCHAR2(10) PRIMARY KEY,
firstNameR VARCHAR2(15) NOT NULL,
lastNameR VARCHAR2(15) NOT NULL,
selfDescription VARCHAR2(100) NOT NULL,
phoneNumR VARCHAR2(20) NOT NULL,
email VARCHAR2(30) NOT NULL,
street VARCHAR2(15) NOT NULL,
city VARCHAR2(15) NOT NULL,
postcode VARCHAR2(15) NOT NULL,
website VARCHAR2(30) NOT NULL,
educationType VARCHAR2(30) NOT NULL,
educationName VARCHAR2(30) NOT NULL,
educationResult VARCHAR2(30) NOT NULL,
skill1 VARCHAR2(30) NOT NULL,
skill2 VARCHAR2(30),
skill3 VARCHAR2(30),
experience1Name VARCHAR2(30) NOT NULL,
experience1Year NUMBER(4) NOT NULL,
experience1Description VARCHAR2(30) NOT NULL,
experience2Name VARCHAR2(30),
experience2Year NUMBER(4),
experience2Description VARCHAR2(30),
username VARCHAR2(10) NOT NULL,
CONSTRAINT fk_res_username FOREIGN KEY (username) REFERENCES
Users(username)
);

CREATE TABLE Asking (
username VARCHAR2(10) NOT NULL,
chatbotID VARCHAR2(10) NOT NULL,
askingDate DATE NOT NULL,
askingTime TIMESTAMP NOT NULL,
CONSTRAINT pk_asking PRIMARY KEY (username, chatbotID),
CONSTRAINT fk_ask_username FOREIGN KEY (username) REFERENCES
Users(username),
CONSTRAINT fk_ask_chatbot FOREIGN KEY (chatbotID) REFERENCES
Chatbot(chatbotID)
);

CREATE TABLE EmailTemplate (
EmailTemplateID VARCHAR2(10) PRIMARY KEY,
accessID VARCHAR2(10) NOT NULL,
CONSTRAINT fk_temp_access FOREIGN KEY (accessID) REFERENCES
Template(accessID)
);

CREATE TABLE EmailPPT (
EmailPPTID VARCHAR2(10) PRIMARY KEY,
accessID VARCHAR2(10) NOT NULL,
CONSTRAINT fk_ppt_access FOREIGN KEY (accessID) REFERENCES
Template(accessID)
);

CREATE TABLE EmailCanva (
emailCanvaID VARCHAR2(10) PRIMARY KEY,
accessID VARCHAR2(10) NOT NULL,
CONSTRAINT fk_canva_access FOREIGN KEY (accessID) REFERENCES
Template(accessID)
);

CREATE TABLE Purchasing (
username VARCHAR2(10) NOT NULL,
productID VARCHAR2(10) NOT NULL,
CONSTRAINT pk_purchasing PRIMARY KEY (username, productID),
CONSTRAINT fk_pur_username FOREIGN KEY (username) REFERENCES
Users(username),
CONSTRAINT fk_pur_product FOREIGN KEY (productID) REFERENCES Product(productID)
);

CREATE TABLE Requesting (
productID VARCHAR2(10) NOT NULL,
accessID VARCHAR2(10) NOT NULL,
CONSTRAINT pk_requesting PRIMARY KEY (productID, accessID),
CONSTRAINT fk_req_product FOREIGN KEY (productID) REFERENCES Product(productID),
CONSTRAINT fk_req_access FOREIGN KEY (accessID) REFERENCES AccessR(accessID)
);
```

6.3 DML 1 - Insert all the table

```
/* Insert data into Product table */
INSERT INTO Product (productID, productName, price)
VALUES('PR001', 'CV Templates', 99);

INSERT INTO Product (productID, productName, price)
VALUES('PR002', 'CVWriting Basic', 159);

INSERT INTO Product (productID, productName, price)
VALUES('PR003', 'CVWriting Pro', 199);

INSERT INTO Product (productID, productName, price)
VALUES('PR004', 'Webinar', 250);

/* Insert data into AccessR table*/

INSERT INTO AccessR (AccessID, AccessName)
VALUES('AC001','EmailTemplate');

INSERT INTO AccessR (AccessID, AccessName)
VALUES ('AC002', 'EmailPPT');

INSERT INTO AccessR (AccessID, AccessName)
VALUES ('AC003', 'EmailCanva');

INSERT INTO AccessR (AccessID, AccessName)
VALUES ('AC004', 'Basic Access');

INSERT INTO AccessR (AccessID, AccessName)
VALUES ('AC005', 'Pro Access');

INSERT INTO AccessR (AccessID, AccessName)
VALUES ('AC006', 'Webinar Access');

/* Insert data into Template table */
INSERT INTO Template (accessID)
VALUES('AC001');

INSERT INTO Template (accessID)
VALUES('AC002');

INSERT INTO Template (accessID)
VALUES('AC003');

/* Insert data into Basic table */
INSERT INTO Basic (accessID, writingCVB, actionVerb, durationB)
VALUES('AC004', 'TRUE', 'TRUE', 10);

/* Insert data into Pro table */
INSERT INTO Pro (accessID, writingCVP, accessLibrary, durationP)
VALUES('AC005', 'TRUE', 'TRUE', 4);

/* Insert data into Webinar table */
```

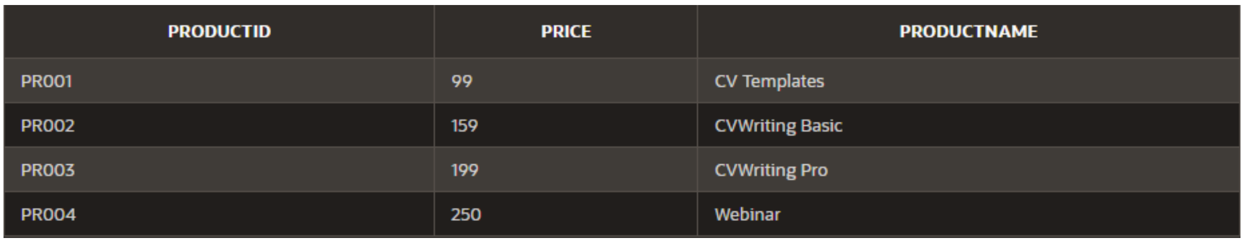
```
INSERT INTO Webinar (accessID, modules, certificate, accessMemberArea)
VALUES('AC006', 'TRUE', 'TRUE', 'TRUE');

/* Insert data into User table */
INSERT INTO Users (username, password, firstName, lastName, phoneNum)
VALUES('sini.lim', '123456', 'SI NI', 'LIM', '1234567890');

INSERT INTO Users (username, password, firstName, lastName, phoneNum)
VALUES('k.xuen', '223456', 'KAI XUEN', 'ONG', '2234567890');

INSERT INTO Users (username, password, firstName, lastName, phoneNum)
VALUES('chan.yee', '323456', 'QING YEE', 'CHAN', '3234567890');

INSERT INTO Users (username, password, firstName, lastName, phoneNum)
VALUES('ooi.sian', '423456', 'WEI SIAN', 'OOI', '4234567890');

/* Insert data into Orders table */

INSERT INTO Orders (orderID, username, ordersDate, ordersTime)
VALUES('OR001', 'sini.lim', TO_DATE('23-FEB-2024', 'DD-MON-YYYY'),
TO_DATE('12:34', 'HH24:MI'));

INSERT INTO Orders (orderID, username, ordersDate, ordersTime)
VALUES('OR002', 'k.xuen', TO_DATE('25-FEB-2024', 'DD-MON-YYYY'),
TO_DATE('13:11', 'HH24:MI'));

INSERT INTO Orders (orderID, username, ordersDate, ordersTime)
VALUES('OR003', 'chan.yee', TO_DATE('15-MAR-2024', 'DD-MON-YYYY'),
TO_DATE('23:11', 'HH24:MI'));

INSERT INTO Orders (orderID, username, ordersDate, ordersTime)
VALUES('OR004', 'ooi.sian', TO_DATE('01-MAR-2024', 'DD-MON-YYYY'),
TO_DATE('09:00', 'HH24:MI'));

/* Insert data into Payment table */

INSERT INTO Payment (transactionID, paymentDate, paymentTime, paymentMethod,
username, orderID)
VALUES('TR001', TO_DATE('23-FEB-2024', 'DD-MON-YYYY'), TO_DATE('12:45',
'HH24:MI'), 'FPX', 'sini.lim', 'OR001');

INSERT INTO Payment (transactionID, paymentDate, paymentTime, paymentMethod,
username, orderID)
VALUES('TR002', TO_DATE('25-FEB-2024', 'DD-MON-YYYY'), TO_DATE('13:22',
'HH24:MI'), 'FPX', 'k.xuen', 'OR002');

INSERT INTO Payment (transactionID, paymentDate, paymentTime, paymentMethod,
username, orderID)
VALUES('TR003', TO_DATE('15-MAR-2024', 'DD-MON-YYYY'), TO_DATE('23:12',
'HH24:MI'), 'Debit Card', 'chan.yee', 'OR003');

INSERT INTO Payment (transactionID, paymentDate, paymentTime, paymentMethod,
username, orderID)
```

```
VALUES('TR004', TO_DATE('01-MAR-2024', 'DD-MON-YYYY'), TO_DATE('09:03',
'HH24:MI'), 'FPX', 'ooi.sian', 'OR004');

/* Insert data into Resume table */
INSERT INTO Resume (resumeID, firstNameR, lastNameR, selfDescription,
phoneNumR, email, street, city, postcode, website, educationType, educationName,
educationResult, skill1, experience1Name, experience1Year, experience1Description,
username)
VALUES('RE001', 'SI NI', 'LIM', 'Year 2 student', '1234567890', 'sini@gmail.com',
'Jalan emas', 'Skudai', '81400', 'www.lsn.com', 'high education', 'UTM', '4', 'eat sleep
drink UTM', 'UTM exp1', '2022', 'exp desc1', 'sini.lim');

/* Insert data into EmailTemplate table */
INSERT INTO EmailTemplate (EmailTemplateID, accessID)
VALUES('ET001', 'AC001');

INSERT INTO EmailTemplate (EmailTemplateID, accessID)
VALUES('ET002', 'AC001');

/* Insert data into EmailPPT table */
INSERT INTO EmailPPT (EmailPPTID, accessID)
VALUES('EP001', 'AC002');

INSERT INTO EmailPPT (EmailPPTID, accessID)
VALUES('EP002', 'AC002');

INSERT INTO EmailPPT (EmailPPTID, accessID)
VALUES('EP003', 'AC002');

/* Insert data into EmailCanva table */
INSERT INTO EmailCanva (emailCanvaID, accessID)
VALUES('EC001', 'AC003');

INSERT INTO EmailCanva (emailCanvaID, accessID)
VALUES('EC002', 'AC003');

INSERT INTO EmailCanva (emailCanvaID, accessID)
VALUES('EC003', 'AC003');

/* Insert data into Purchasing table */
INSERT INTO Purchasing (username, productID)
VALUES('sini.lim', 'PR001');

INSERT INTO Purchasing (username, productID)
VALUES('k.xuen', 'PR002');

INSERT INTO Purchasing (username, productID)
VALUES('chan.yee', 'PR003');

INSERT INTO Purchasing (username, productID)
VALUES('ooi.sian', 'PR004');

/* Insert data into Requesting table */
```

```
INSERT INTO Requesting (productID, accessID)

VALUES('PR001', 'AC001');

INSERT INTO Requesting (productID, accessID)
VALUES('PR001', 'AC002');

INSERT INTO Requesting (productID, accessID)
VALUES('PR001', 'AC003');

INSERT INTO Requesting (productID, accessID)
VALUES('PR002', 'AC004');

INSERT INTO Requesting (productID, accessID)
VALUES('PR003', 'AC005');

INSERT INTO Requesting (productID, accessID)
VALUES('PR004', 'AC006');

/* Insert data into Chatbot table */

INSERT INTO Chatbot (chatbotID, keyword, answer)
VALUES('CB001', 'product price', '1. CV templates - RM99.00, 2. CV Writing Basic -
RM159.00, 3. CV Writing Pro - RM199.00, 4. Webinar - RM250.00');

INSERT INTO Chatbot (chatbotID, keyword, answer)
VALUES('CB002', 'access expire', 'will no expire');

INSERT INTO Chatbot (chatbotID, keyword, answer)
VALUES('CB003', 'product discount', 'coming soon');

/* Insert data into Asking table */

INSERT INTO Asking (username, chatbotID, askingDate, askingTime)
VALUES('sini.lim', 'CB001', TO_DATE('22-NOV-2023', 'DD-MON-YYYY'),
TO_DATE('13:23', 'HH24:MI'));

INSERT INTO Asking (username, chatbotID, askingDate, askingTime)
VALUES('k.xuen', 'CB002', TO_DATE('24-DEC-2023', 'DD-MON-YYYY'),
TO_DATE('11:09', 'HH24:MI'));

INSERT INTO Asking (username, chatbotID, askingDate, askingTime)
VALUES('chan.yee', 'CB003', TO_DATE('26-DEC-2023', 'DD-MON-YYYY'),
TO_DATE('23:22', 'HH24:MI'));

INSERT INTO Asking (username, chatbotID, askingDate, askingTime)
VALUES('ooi.sian', 'CB003', TO_DATE('02-MAR-2024', 'DD-MON-YYYY'),
TO_DATE('10:23', 'HH24:MI'));
```

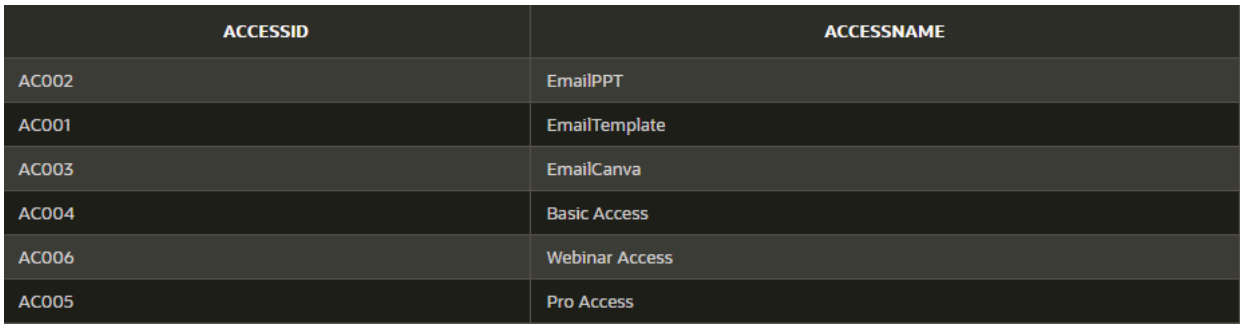
6.4 DML 2 - Display all the table

SELECT * FROM Product;



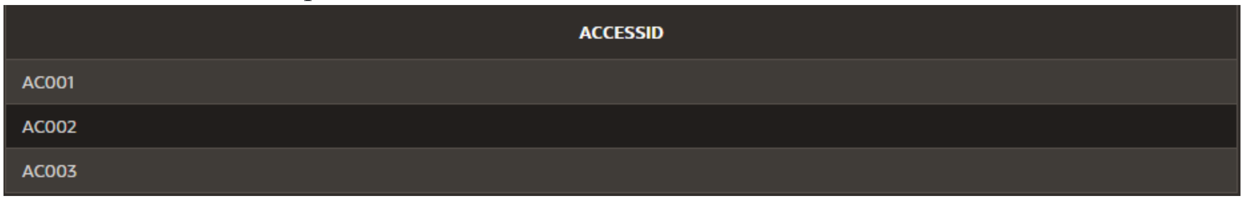
PRODUCTID	PRICE	PRODUCTNAME
PR001	99	CV Templates
PR002	159	CVWriting Basic
PR003	199	CVWriting Pro
PR004	250	Webinar

SELECT * FROM AccessR;



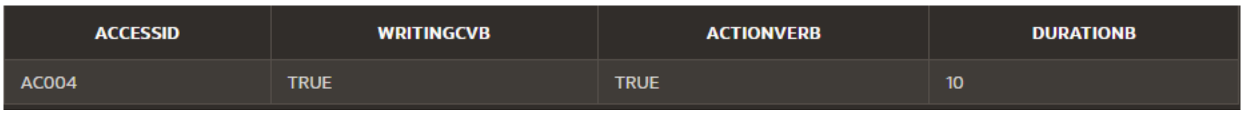
ACCESSID	ACCESSNAME
AC002	EmailPPT
AC001	EmailTemplate
AC003	EmailCanva
AC004	Basic Access
AC006	Webinar Access
AC005	Pro Access

SELECT * FROM Template;



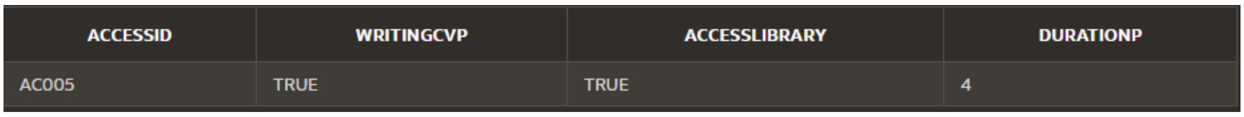
ACCESSID
AC001
AC002
AC003

SELECT * FROM Basic;



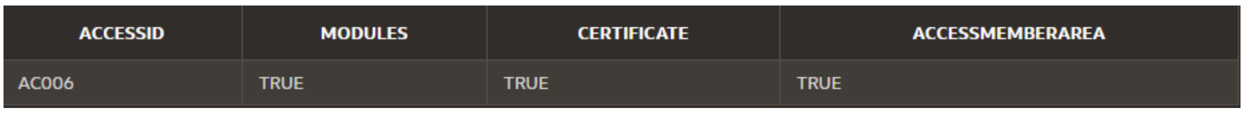
ACCESSID	WRITINGCVB	ACTIONVERB	DURATIONB
AC004	TRUE	TRUE	10

SELECT * FROM Pro;



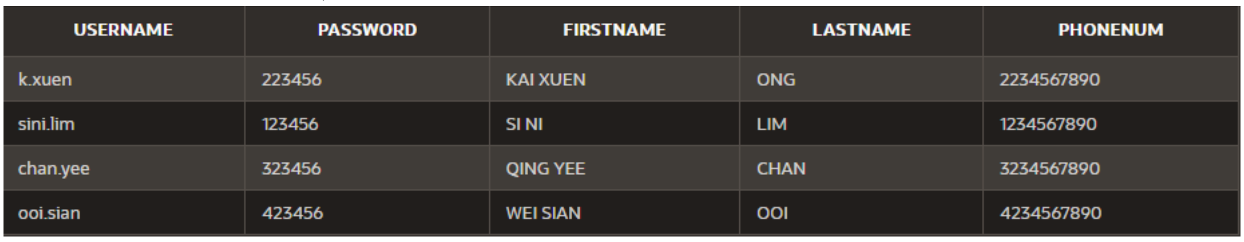
ACCESSID	WRITINGCVP	ACCESSLIBRARY	DURATIONP
AC005	TRUE	TRUE	4

SELECT * FROM Webinar;



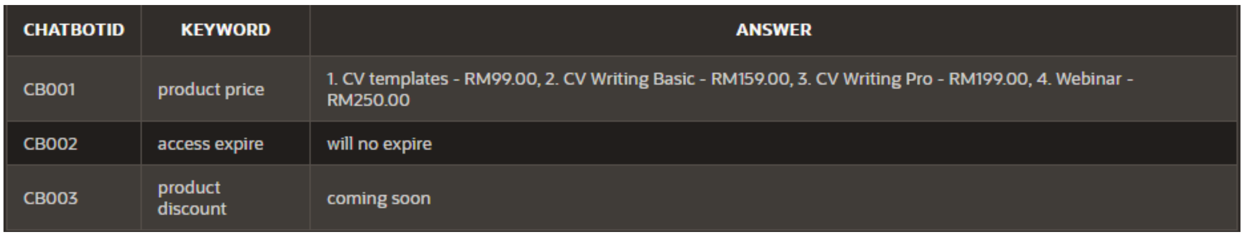
ACCESSID	MODULES	CERTIFICATE	ACCESSMEMBERAREA
AC006	TRUE	TRUE	TRUE

SELECT * FROM Users;



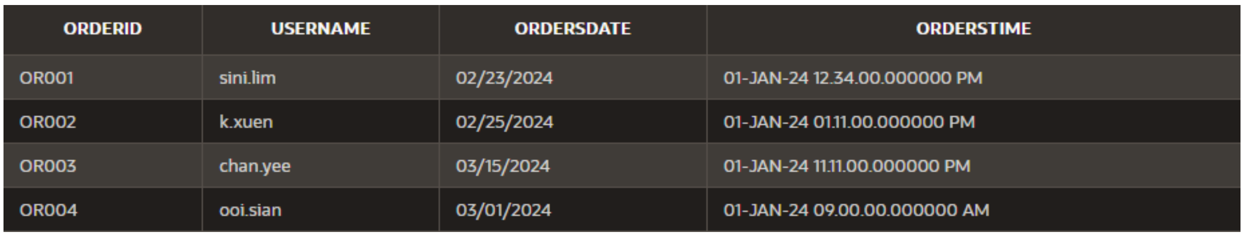
USERNAME	PASSWORD	FIRSTNAME	LASTNAME	PHONENUM
k.xuen	223456	KAI XUEN	ONG	2234567890
sini.lim	123456	SI NI	LIM	1234567890
chan.yee	323456	QING YEE	CHAN	3234567890
ooi.sian	423456	WEI SIAN	OOI	4234567890

SELECT * FROM Chatbot;



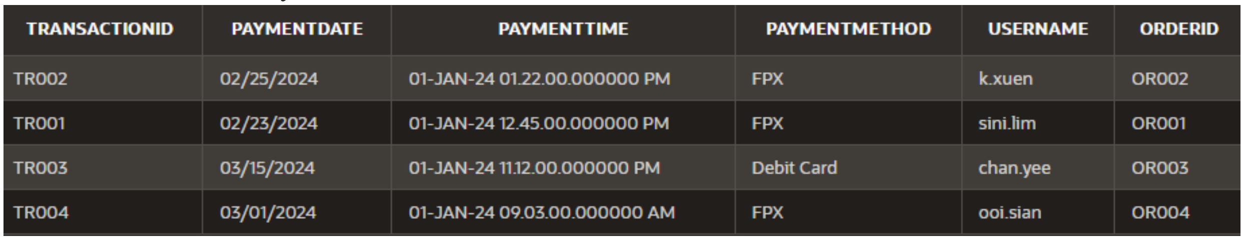
CHATBOTID	KEYWORD	ANSWER
CB001	product price	1. CV templates - RM99.00, 2. CV Writing Basic - RM159.00, 3. CV Writing Pro - RM199.00, 4. Webinar - RM250.00
CB002	access expire	will no expire
CB003	product discount	coming soon

SELECT * FROM Orders;



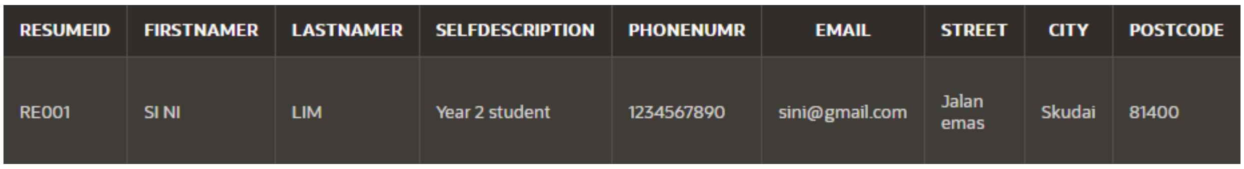
ORDERID	USERNAME	ORDERSDATE	ORDERSTIME
OR001	sini.lim	02/23/2024	01-JAN-24 12.34.00.000000 PM
OR002	k.xuen	02/25/2024	01-JAN-24 01.11.00.000000 PM
OR003	chan.yee	03/15/2024	01-JAN-24 11.11.00.000000 PM
OR004	ooi.sian	03/01/2024	01-JAN-24 09.00.00.000000 AM

SELECT * FROM Payment;



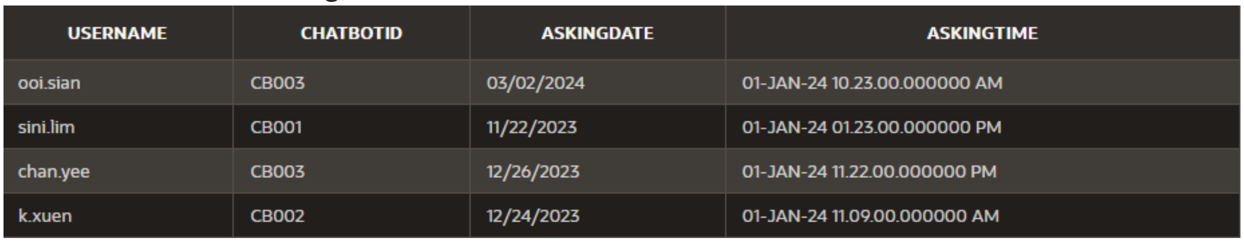
TRANSACTIONID	PAYMENTDATE	PAYMENTTIME	PAYMENTMETHOD	USERNAME	ORDERID
TR002	02/25/2024	01-JAN-24 01.22.00.000000 PM	FPX	k.xuen	OR002
TR001	02/23/2024	01-JAN-24 12.45.00.000000 PM	FPX	sini.lim	OR001
TR003	03/15/2024	01-JAN-24 11.12.00.000000 PM	Debit Card	chan.yee	OR003
TR004	03/01/2024	01-JAN-24 09.03.00.000000 AM	FPX	ooi.sian	OR004

SELECT * FROM Resume;



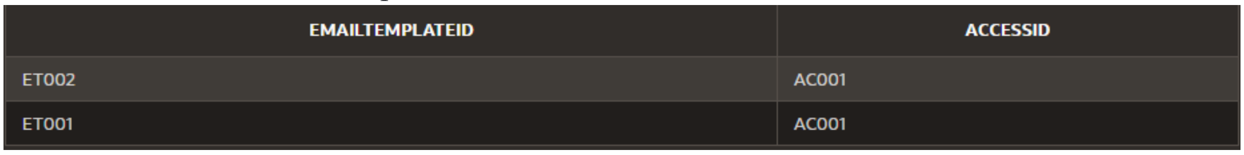
RESUMEID	FIRSTNAMER	LASTNAMER	SELFDESCRIPTION	PHONENUMR	EMAIL	STREET	CITY	POSTCODE
RE001	SI NI	LIM	Year 2 student	1234567890	sini@gmail.com	Jalan emas	Skudai	81400

SELECT * FROM Asking;



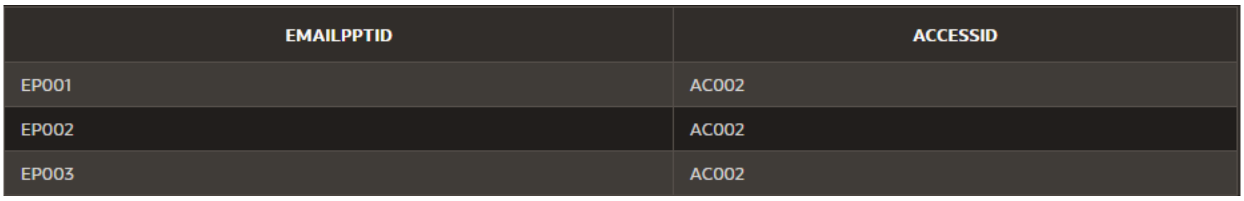
USERNAME	CHATBOTID	ASKINGDATE	ASKINGTIME
ooi.sian	CB003	03/02/2024	01-JAN-24 10.23.00.000000 AM
sini.lim	CB001	11/22/2023	01-JAN-24 01.23.00.000000 PM
chan.yee	CB003	12/26/2023	01-JAN-24 11.22.00.000000 PM
k.xuen	CB002	12/24/2023	01-JAN-24 11.09.00.000000 AM

SELECT * FROM EmailTemplate;



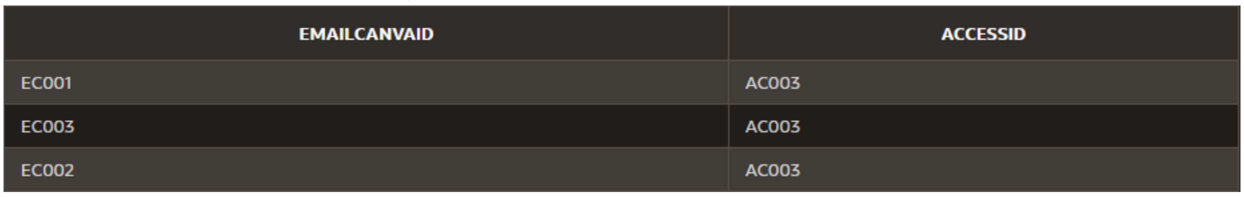
EMAILTEMPLATEID	ACCESSID
ET002	AC001
ET001	AC001

SELECT * FROM EmailPPT;



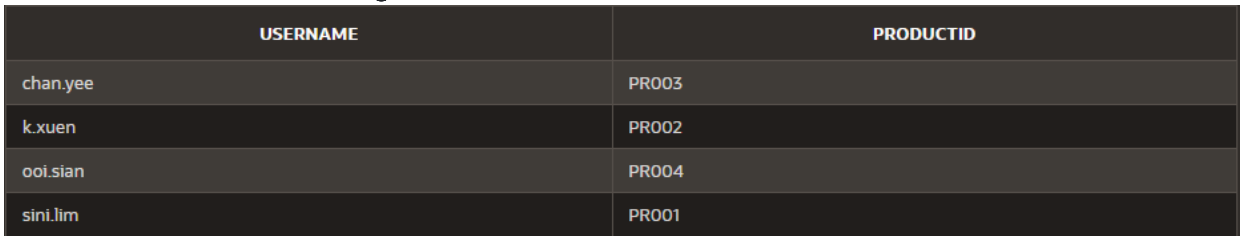
EMAILPPTID	ACCESSID
EP001	AC002
EP002	AC002
EP003	AC002

SELECT * FROM EmailCanva;



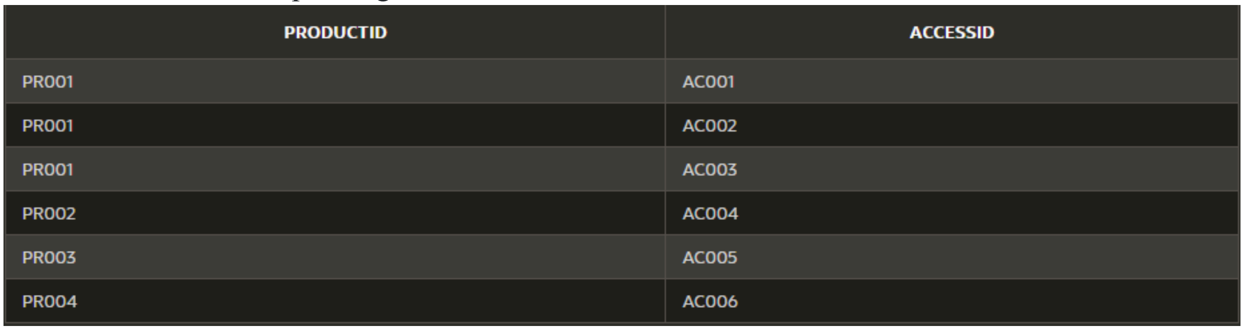
EMAILCANVAID	ACCESSID
EC001	AC003
EC003	AC003
EC002	AC003

SELECT * FROM Purchasing;



USERNAME	PRODUCTID
chan.yee	PR003
k.xuen	PR002
ooi.sian	PR004
sini.lim	PR001

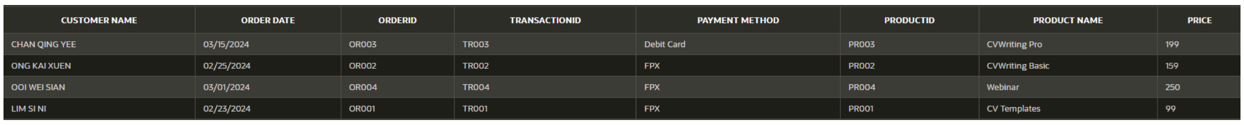
SELECT * FROM Requesting;



PRODUCTID	ACCESSID
PR001	AC001
PR001	AC002
PR001	AC003
PR002	AC004
PR003	AC005
PR004	AC006

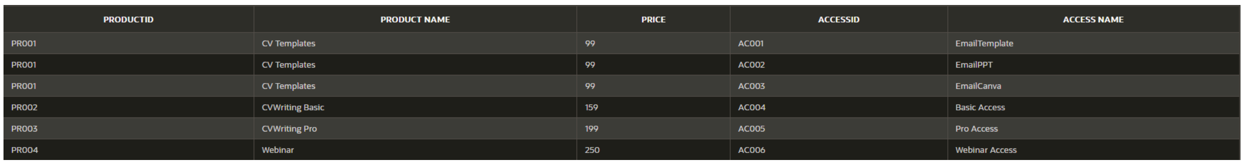
6.5 DML 3 - Display the interface

```
/*Purchasing Interface*/
SELECT u.lastName || ' ' || u.firstName "CUSTOMER NAME", o.ordersDate "ORDER DATE", o.orderID, pay.transactionID, pay.paymentMethod "PAYMENT METHOD", prod.productID, prod.productName "PRODUCT NAME", prod.price
FROM Users u JOIN Orders o
ON u.username = o.username
JOIN Payment pay
ON o.orderID = pay.orderID
JOIN Purchasing pur
ON pur.username = u.username
JOIN Product prod
ON prod.productID = pur.productID
ORDER BY u.username;
```



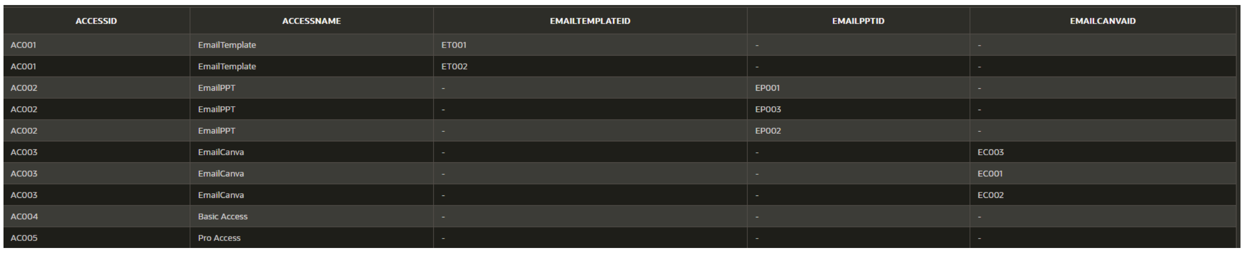
CUSTOMER NAME	ORDER DATE	ORDERID	TRANSACTIONID	PAYMENT METHOD	PRODUCTID	PRODUCT NAME	PRICE
CHAN QING YEE	03/15/2024	OR003	TR003	Debit Card	PR003	CVWriting Pro	199
ONG KAI XUEN	02/25/2024	OR002	TR002	FPX	PR002	CVWriting Basic	159
OOI WEI SIAN	03/01/2024	OR004	TR004	FPX	PR004	Webinar	250
LIM SI NI	02/23/2024	OR001	TR001	FPX	PR001	CV Templates	99

```
/*Product Introduce Interface*/
SELECT prod.productID, prod.productName "PRODUCT NAME", prod.price, acc.accessID, acc.accessName "ACCESS NAME"
FROM Product prod JOIN Requesting req
ON prod.productID = req.productID
JOIN AccessR acc
ON acc.accessID = req.accessID;
```



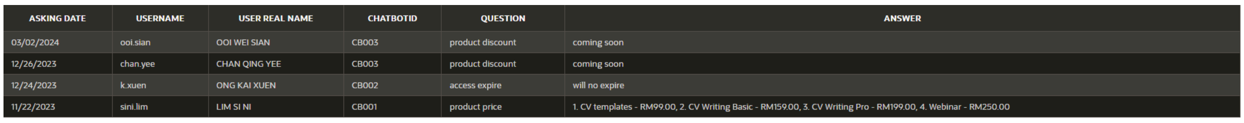
PRODUCTID	PRODUCT NAME	PRICE	ACCESSID	ACCESS NAME
PR001	CV Templates	99	AC001	EmailTemplate
PR001	CV Templates	99	AC002	EmailPPT
PR001	CV Templates	99	AC003	EmailCanva
PR002	CVWriting Basic	159	AC004	Basic Access
PR003	CVWriting Pro	199	AC005	Pro Access
PR004	Webinar	250	AC006	Webinar Access

```
/*Describe Product Access*/
SELECT acc.accessID, acc.accessName, eT.emailTemplateID, eP.emailPPTID, eC.emailCanvaID
FROM AccessR acc LEFT OUTER JOIN EmailTemplate eT
ON eT.accessID = acc.accessID
LEFT OUTER JOIN EmailPPT eP
ON eP.accessID = acc.accessID
LEFT OUTER JOIN EmailCanva eC
ON eC.accessID = acc.accessID
ORDER BY acc.accessID;
```



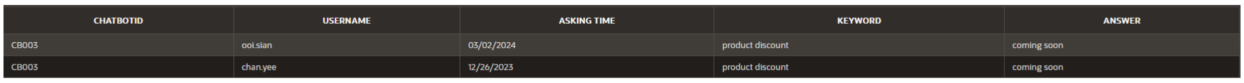
ACCESSID	ACCESSNAME	EMAILTEMPLATEID	EMAILPPTID	EMAILCANVAID
AC001	EmailTemplate	ET001	-	-
AC001	EmailTemplate	ET002	-	-
AC002	EmailPPT	-	EP001	-
AC002	EmailPPT	-	EP003	-
AC002	EmailPPT	-	EP002	-
AC003	EmailCanva	-	-	EC003
AC003	EmailCanva	-	-	EC001
AC003	EmailCanva	-	-	EC002
AC004	Basic Access	-	-	-
AC005	Pro Access	-	-	-

```
/*Sending Enquiries Interface*/
SELECT ask.askingDate "ASKING DATE", u.username, u.lastName || ' ' || u.firstName
"USER REAL NAME", c.chatbotID, c.keyword "QUESTION", c.answer
FROM Users u JOIN Asking ask
ON u.username = ask.username
JOIN Chatbot c
ON c.chatbotID = ask.chatbotID
ORDER BY ask.askingDate DESC;
```



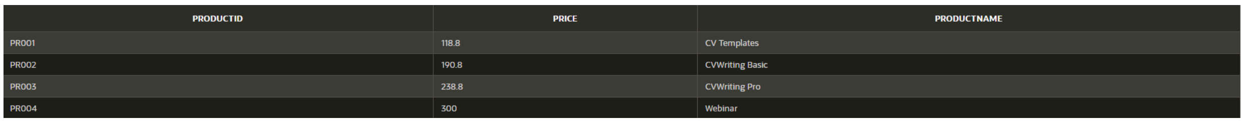
ASKING DATE	USERNAME	USER REAL NAME	CHATBOTID	QUESTION	ANSWER
03/02/2024	ooi.sian	OOI WEI SIAN	CB003	product discount	coming soon
12/26/2023	chan.yee	CHAN QING YEE	CB003	product discount	coming soon
12/24/2023	k.xuen	ONG KAI XUEN	CB002	access expire	will no expire
11/22/2023	sini.lim	LIM SI NI	CB001	product price	1. CV templates - RM99.00, 2. CV Writing Basic - RM159.00, 3. CV Writing Pro - RM199.00, 4. Webinar - RM250.00

```
/*Select user that ask for question chatbotID = 3*/
SELECT a.chatbotID, a.username, a.askingDate "ASKING TIME", c.keyword, c.answer
FROM Asking a JOIN Chatbot c
ON a.chatbotID = c.chatbotID
WHERE a.chatbotID = 'CB003';
```



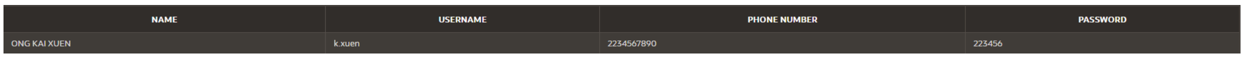
CHATBOTID	USERNAME	ASKING TIME	KEYWORD	ANSWER
CB003	ooi.sian	03/02/2024	product discount	coming soon
CB003	chan.yee	12/26/2023	product discount	coming soon

```
/*Increase the price of product by multiplying 1.2*/
UPDATE Product
SET price = price*1.2;
SELECT * FROM Product;
```



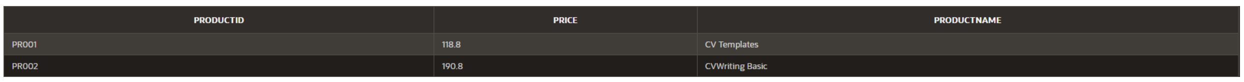
PRODUCTID	PRICE	PRODUCTNAME
PR001	118.8	CV Templates
PR002	190.8	CVWriting Basic
PR003	238.8	CVWriting Pro
PR004	300	Webinar

```
/*Select user with the last name that start with 'O'*/
SELECT lastName || ' ' || firstName AS NAME, username, phoneNum "PHONE
NUMBER", password
FROM Users
WHERE lastName LIKE 'O%' AND phoneNum = '2234567890';
```



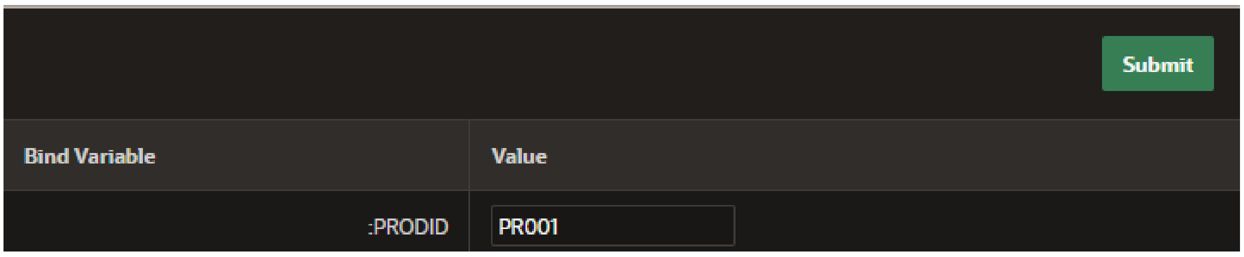
NAME	USERNAME	PHONE NUMBER	PASSWORD
ONG KAI XUEN	k.xuen	2234567890	223456

```
/*Select Product with the price between 100 until 200 */
SELECT *
FROM Product
WHERE price BETWEEN 100 AND 200;
```

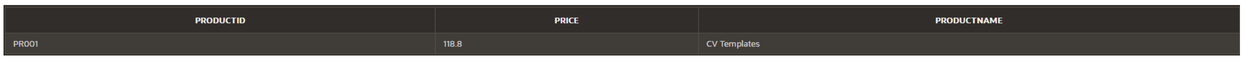


PRODUCTID	PRICE	PRODUCTNAME
PR001	118.8	CV Templates
PR002	190.8	CVWriting Basic

```
/*Allow user search the product details based on productID*/
SELECT *
FROM Product
WHERE productID = :prodID;
```



Bind Variable	Value
:PRODID	PR001



PRODUCTID	PRICE	PRODUCTNAME
PR001	118.8	CV Templates

7.0 Summary

Finally, our group has completed database design phase 3 for getMeHired website.
The third phase of the database design project constitutes a significant milestone as all the business rules are well updated. Recognizing the changes in business needs, the conceptual and improved ERDs are carefully redesigned ensuring the accurate representation of the organization's data requirements. The subsequent step of a logical database design is the creation of a complex logical ERD, the improvement of the data dictionary, and the application of normalization techniques for better data integrity and efficiency. This normalization process lays the foundation for the development of relational database schemas as it ensures that the data structure of the database is optimized and free of redundancies. During Phase 3's final steps, SQL statements are also well generated, combining DDL and DML to articulate the database blueprint.

Throughout this phase, we learned the importance of a well-organized structure in logical database design, employing normalization for data integrity. Additionally, we acquired practical skills in implementing conceptual and logical designs into SQL statements, we also acknowledged the essential link between theoretical database concepts and their real-world execution.