UNIVERSITI MALAYA

WIA2001 DATABASE PROJECT REPORT 1
(GROUP 9)

LECTURER: ASSOCIATE PROF. DR. FARIZA HANUM BINTI MD NASARUDDIN

NAME	MATRIC NO.
Lee Kai Yang	U2005398
Liew Rui Zhi	U2005270
Chow Shang Shyan	U2005283
Chin Chin Fang	U2005349
Wan Muhammad Azib	17207678

Table of content

1. Introduction

1.1 Introduction to business

Akenz Motor is a middle size motorcycle selling and repairing shop since 2010, located at Jertih, Terengganu. It aims to deliver quality sales and service to every customer. Akenz Motor only sells brand new motorcycles at affordable prices so that everyone can purchase their own motorcycle, be it for personal use, company use, lucky draw prices, or as a gift for your loved ones. Besides, Akenz Motor also provides top quality services for motorcycles which include changing engine and gear oil plus repairing. Moreover, Akenz Motor sells a small number of motorcycles and related accessories.

Our team has chosen Akenz Motor because the size of it is suitable for our database project. Middle size company means that we will not end up with a huge database which will be both hard to design and implement. On the other hand, the company does not have a solid database yet so we will be helping them to build one. This would greatly help them to manage their data more efficiently and to handle the increasing amount of data from time to time.

1.2 Objective

The main objective of Akenz Motor is to provide the best customer experience through products and services. Primarily, we choose Akenz Motor as it is not a multinational company or large scale enterprise in which the data flow is huge. Moreover, Akenz Motor is famous for its great customer experience and after-sale services. Akenz Motor helps us to understand more about the operation of a database in business which is definitely useful for us. We hope our database design can help in integrating the necessary data in the company and improving the business performances.

1.3 Business operation

The business operation in Akenz Motor starts from 9am until 6pm except on Friday. The staff will open the shop and carry out inventory checking on the stores such as motorcycles and spare parts every morning.

The main services that Akenz Motor offers are motorcycle sale, motorcycle services and repairs, spare parts sale, insurance and roadtax sale. To ensure the best presentation of the products, the staff will carry out cleaning on the stores and products. In conclusion, Akenz Motor offers in-store purchases, in-store pick-up, delivery and repair services for their customers everyday.

1.4 Issue in managing data

Akenz Motor faced several issues when managing their data. First of all, Akenz Motor manually checks its inventory to keep track of actual stock count. This work is both exhausting and error-prone. Moreover, as the business grows, there is more customer data to be stored, yet they haven't found a better way to store all of the information just yet. Which leads to the next point, Akenz Motor did not have a proper database management system, it is very hard for the company to ensure the data integrity and reduce data redundancy as more and more customer data are being inserted into the computer.

At the same time, we faced issues when managing the data, which is limited knowledge. We did not have experience in designing such complex databases for business, not even in doing tutorial exercises. Thus, we were trying our best to produce a usable and good database for our collaborator.

2. Business rule

These business rules of Akenz Motor are obtained from the manager, Mr. Liew Yoon Hee via Google Meet video call on 26 November 2021.

1. One supplier is responsible for one or more supplier orders.
2. Every supplier order of Akenz Motor is handled by only one supplier.
3. One or many products are present in one supplier order.
4. One supplier order made by Akenz Motor contains one or many products.
5. For every order made by a customer, there is at least a product in it.
6. A product will only be ordered by one and only one customer.
7. Every order of product must be made by a customer only.
8. Every customer can make one or more orders**.**
9. Each service requested is made by a customer only.
10. Every customer can walk into Akenz Motor and request for one or more services.
11. A payment is required for every order by the customer.
12. Every order made by a customer needs a payment either by cash or installment plan provided by Aeon Credit Service Malaysia Berhad.
13. A payment is required for every service requested by the customer.
14. A customer must pay for every service requested from Akenz Motor.
15. Each payment by the customer will generate a receipt to be given to the customer.
16. Every receipt will only be generated when a customer has made a payment.
17. Every receipt given to the customer is provided by only one Akenz Motor staff.
18. One staff can provide at least one receipt to the customer.
19. Every staff in the company holds a specific position.
20. A position in Akenz Motor is held by only one staff member.

3. User requirements

The database should consist of a receipt table which contains information about warranty and the staff who issues it. Next, the service needs to be in a different table from the order. Moreover, the motorcycle's engine and chassis number should be stored. Lastly, this database only consists of motorcycles as products, ignoring spare parts, insurance and key duplication business.

4. Entities, Attributes, Relationships, Cardinalities and Constraints

4.1 Entities with Attributes

From the above business rules and research on Akenz Motor, the following entities and attributes are identified.

1. SUPPLIER (**supplier_id**, supplier_name, address, pic_contact_no, pic_name)

   SUPPLIER table contains details of suppliers of Akenz Motor. It links to SUPPLIER_ORDER via supplier_id.

2. SUPPLIER_ORDER (**supplier_order_id**, supplier_id, engine_no, total_price)

   SUPPLIER_ORDER table contains order information of Akenz Motor to suppliers. It links to SUPPLIER via supplier_id and to PRODUCT via engine_no.

3. PRODUCT (**engine_no**, chassis_no, unit_cost, unit_price, brand, model, colour)

   PRODUCT table refers to the motorcycle obtained from suppliers and serves as the inventory of Akenz Motor until customers buy them. It links to SUPPLIER_ORDER and CUSTOMER_ORDER via engine_no.

4. CUSTOMER_ORDER (**customer_ic_no**, **engine_no**, date**,** payment_id, customer_name, customer_contact_no, customer_address, agreement_no, down_payment_amount, monthly_installment_amount)

   CUSTOMER_ORDER table refers to the order made by customers in the store. It links to PRODUCT via engine_no and to CASH_PAYMENT via payment_id.

5. SERVICE (**service_id**, description, customer_ic_no, payment_id)

   SERVICE table contains information on service requested by customers in the store. It links to CUSTOMER_ORDER via customer_ic_no and to CASH_PAYMENT via payment_id.

6. CASH_PAYMENT (**payment_id**, total_amount)

   CASH_PAYMENT table contains payment details of customers for motorcycles and services in Akenz Motor. It links to CUSTOMER_ORDER, SERVICE and RECEIPT via payment_id.

7. RECEIPT (**receipt_no**, timestamp, payment_id, staff_id, warranty)

   RECEIPT table contains information such as warranty and serves as proof to the customers that an order is successful. It links to CASH_PAYMENT via payment_id and to STAFF via staff_id.

8. STAFF (**staff_id**, name, ic_no, address, contact_no, position_name)

   STAFF table contains information and personal details of every staff in Akenz Motor. It links to RECEIPT and WORK_TIME via staff_id and to POSITION via position_name.

9. WORK_TIME (**staff_id**, date, check_in_time, check_out_time)

   WORK_TIME table contains information of daily check-in and check-out time of every staff member in Akenz Motor. It links to STAFF via staff_id.

10. POSITION (**position_name**, base_salary, overtime_rate)

    POSITION table contains roles or positions for staff member in Akenz Motor. It links to STAFF via position_name.

4.2 Relationships and Cardinalities

One-to-one relationship

- One customer order requires one and only one cash payment.
- One cash payment is required in one and only one customer order.
- One service requires one and only one cash payment.
- One cash payment is required in one and only one service.
- One cash payment generates one and only one receipt.
- One receipt is generated by one and only one cash payment.
- One staff has one and only one work time.
- One work time belongs to one and only one staff.
- One staff holds one and only one position.
- One position is held by one and only one staff.

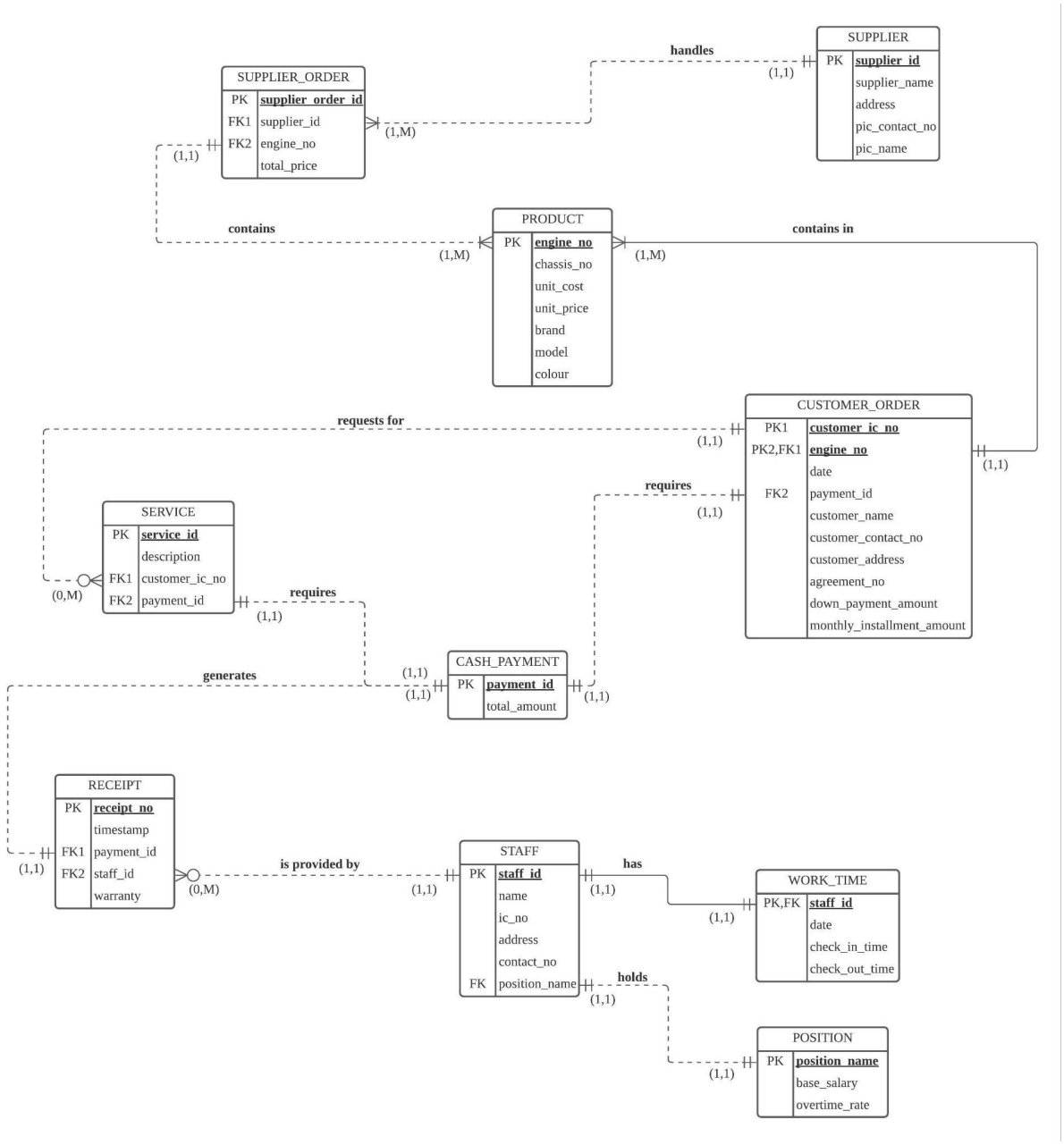
One-to-many relationship

- One supplier handles one or many supplier orders.
- One supplier order contains one or many products.
- One customer order contains one or many products.
- One customer order requests for zero or many services.
- One staff provides zero or many receipts.

Many-to-one relationship

- One or many supplier orders are handled by one and only one supplier.
- One or many products are contained in one and only one supplier order.
- One or many products are contained in one and only one customer order.
- Zero or many services are requested by one and only one customer order.
- Zero or many receipts are provided by one and only one staff.

5. Conceptual Entity Relationship Diagram



6. Normalization Process

*** All of the data in the table below are dummy data.**

Main purpose of normalization in the database is to evaluate and to correct the structure of the table in order to minimize data redundancy and data anomalies. It also ensures each table conforms to the concept of well-formed relations which are indicated below:

- Each table is free from insertion, update and deletion anomalies
- Each table represents a single subset
- No data item will be unnecessarily stored in more than one table
- All non-prime attributes in a table are dependent on the primary key

1st Normal Form (1NF)

- All the order in which data is stored does not matter
- Each cell is single-valued
- Entries in a column are same typed
- Each row is uniquely identified

For example for SUPPLIER_ORDER table, when UNF, there is more than one value in engine_no column which does not obey 1NF.

supplier_order_id	**supplier_id**	**engine_no**	**total_price**
A120976	5120976	FX730V XL1200C	100
B126549	5126549	52WVC10338 39KJG10995 40TCF22583	370
B129644	7129644	FJ1207	600

Since there are multiple engine_no for one supplier_order_id , hence the repeating groups should be eliminated. There are composite keys in this table which are supplier_order_id and also engine_no.

supplier_order_id	**supplier_id**	**engine_no**	**total_price**
A120976	5120976	FX730V	100
A120976	5120976	XL1200C	100
B126549	5126549	52WVC10338	370

B126549	5126549	39KJG10995	370
B126549	5126549	40TCF22583	370
B129644	7129644	FJ1207	600

At the end of the first normalization, the entities and attributes are below:
References: -> means "determines"

1. SUPPLIER (**supplier_id**, supplier_name, address, pic_contact_no, pic_name)
   Transitive Dependency:
   (pic_contact_no -> pic_name)

2. SUPPLIER_ORDER (**supplier_order_id**, supplier_id, engine_no, total_price)

3. PRODUCT (**engine_no**, chassis_no, unit_cost, unit_price, brand, model, colour)

4. CUSTOMER_ORDER (**customer_ic_no**, **engine_no**, date**,** payment_id, customer_name, customer_contact_no, customer_address, agreement_no, down_payment_amount, monthly_installment_amount)
   Partial Dependency:
   (customer_ic_no -> customer_ name, customer_contact_no, customer_address,)

5. SERVICE (**service_id**, description, customer_ic_no, payment_id)

6. CASH_PAYMENT (**payment_id**, total_amount)

7. RECEIPT (**receipt_no**, timestamp, payment_id, staff_id, warranty)

8. STAFF (**staff_id**, name, ic_no, address, contact_no, position_name)

9. WORK_TIME (**staff_id**, **date**, check_in_time, check_out_time)

10. POSITION (**position_name**, base_salary, overtime_rate)

2nd Normal Form (2NF)

- Table is in 2NF when it is in 1NF and has no partial dependencies
- No attribute is dependent on only portion of primary key

For example for CUSTOMER_ORDER table, when 1NF, there are some non-key attributes (customer_name,customer_contact_no,customer_address,agreement_no,down_payment_amount, monthly_installment_amount) which are dependent on one of the two candidate keys in this table (customer_ic_no) only. These attributes are not dependent on engine_no.

customer_ic_no	**engine_no**	**date**	**payment_id**	**customer_name**	**customer_contact_no**	**customer_address**	**agreement_no**	**down_payment_no**	**monthly_installment_amount**
010220-03-1258	A1000	20-03-2020	1023	Ali	012-3457690	1, Taman A, 48000, Selangor	10250	2500	400
010220-03-1258	F8989	5-10-2021	1024	Abu	012-345769	2, Taman A, 48000, Selangor	8900	1000	300
010330-05-8858	K231Z	25-12-2021	1025	Omar	012-3457692	3, Taman A, 48000, Selangor			
010330-05-8858	A8000Z	30-10-2020	1026	Siti	012-3457693	4, Taman A, 48000, Selangor	10230	2000	500
010330-05-8858	BU2500	8-9-2021	1027	Mei	012-3457694	5, Taman A, 48000, Selangor	8989	5000	700

After normalization, original CUSTOMER is now converted into 2 tables as below:

CUSTOMER(**customer_ic_no**, customer_name, customer_contact_no, customer_address, agreement_no, down_payment_no, monthly_installment_amount)

customer_ic_no	**customer_name**	**customer_contact_no**	**customer_address**	**agreement_no**	**down_payment_no**	**monthly_installment_amount**
010220-03-1258	Ali	012-3457690	1, Taman A, 48000, Selangor	10250	2500	400
010220-03-1258	Abu	012-345769	2, Taman A, 48000, Selangor	8900	1000	300
010330-05-8858	Omar	012-3457692	3, Taman A, 48000, Selangor			
010330-05-8858	Siti	012-3457693	4, Taman A, 48000, Selangor	10230	2000	500
010330-05-8858	Mei	012-3457694	5, Taman A, 48000, Selangor	8989	5000	700

CUSTOMER_ORDER**(customer_ic_no, engine_no, date, payment_id)**

customer_ic_no	**engine_no**	**date**	**payment_id**
010220-03-1258	A1000	20-03-2020	1023
010220-03-1258	F8989	05-10-2021	1024
010330-05-8858	K231Z	25-12-2021	1025
010330-05-8858	A8000Z	30-10-2020	1026
010330-05-8858	BU2500	08-09-2021	1027

At the end of the first normalization, the entities and attributes are below:
References: -> means "determines"

1. SUPPLIER (**supplier_id**, supplier_name, address, pic_contact_no, pic_name)
2. SUPPLIER_ORDER (**supplier_order_id**, supplier_id, engine_no, total_price)
3. PRODUCT (**engine_no**, chassis_no, unit_cost, unit_price, brand, model, colour)
4. CUSTOMER_ORDER (**customer_ic_no**, **engine_no**, date, payment_id)

5. CUSTOMER (**customer_ic_no**, name, contact_no, address, agreement_no)
6. SERVICE (**service_id**, description, customer_ic_no, payment_id)
7. CASH_PAYMENT (**payment_id**, total_amount)
8. RECEIPT (**receipt_no**, timestamp, payment_id, staff_id, warranty)
9. STAFF (**staff_id**, name, ic_no, address, contact_no, position_name)
10. WORK_TIME (**staff_id**, **date**, check_in_time, check_out_time)
11. POSITION (**position_name**, base_salary, overtime_rate)

3rd Normal Form (3NF)

- Satisfy all conditions in 1NF and 2NF
- All fields can be determined only by the non primary key in the table

Next, for the CUSTOMER table, when 1NF, there are some non-key attributes (down_payment_amount, monthly_installment_amount) which are dependent on one of the candidate keys in this table (agreement_no) only.

customer_ic_no	**name**	**contact_no**	**address**	**agreement_no**	**down_payment_amount**	**monthly_installment_amount**
010220-03-1258	Ali	012-3457690	1, Taman A, 48000, Selangor	A123	400	1200
010220-03-1258	Abu	012-3457691	2, Taman A, 48000, Selangor	A124	400	1300
010330-05-8858	Omar	012-3457692	3, Taman A, 48000, Selangor	A125	300	900
010330-05-8858	Siti	012-3457693	4, Taman A, 48000, Selangor	A126	200	700
010330-05-8858	Mei	012-3457694	5, Taman A, 48000, Selangor	A127	100	700

After normalization, original CUSTOMER is now converted into 2 tables as below:

CUSTOMER (**customer_ic_no**, name, contact_no, address, agreement_no)

customer_ic_no	**name**	**contact_no**	**address**
010220-03-1258	Ali	012-3457690	1, Taman A, 48000, Selangor
010220-03-1258	Abu	012-3457691	2, Taman A, 48000, Selangor
010330-05-8858	Omar	012-3457692	3, Taman A, 48000, Selangor
010330-05-8858	Siti	012-3457693	4, Taman A, 48000, Selangor
010330-05-8858	Mei	012-3457694	5, Taman A, 48000, Selangor

INSTALLMENT (**agreement_no**, down_payment_amount, monthly_instalment, amount)

agreement_no	**down_payment_amount**	**monthly_installment_amount**
A123	400	1200
A124	400	1300
A125	300	900
A127	100	700
A128	400	1200

At the end of the first normalization, the entities and attributes are below:
References: -> means "determines"

1. SUPPLIER (**supplier_id**, supplier_name, address, pic_name,pic_contact_no)
2. SUPPLIER_ORDER (**supplier_order_id,** supplier_id, engine_no, total_price)
3. PRODUCT (**engine_no,** chassis_no, unit_cost, unit_price, brand, model, colour)
4. CUSTOMER_ORDER (**customer_ic_no**, **engine_no**, date, payment_id)
5. CUSTOMER (**customer_ic_no**, name, contact_no, address, agreement_no)
6. INSTALLMENT(**agreement_no**,down_payment_amount, monthly_installment_amount)
7. SERVICE (**service_id**, description, customer_ic_no, payment_id)
8. CASH_PAYMENT (**payment_id**, total_amount)
9. RECEIPT (**receipt_no**, timestamp, payment_id, staff_id, warranty)
10. STAFF (**staff_id**, name, ic_no, address, contact_no, position_name)
11. WORK_TIME (**staff_id**, **date**, check_in_time, check_out_time)
12. POSITION (**position_name**, base_salary, overtime_rate)

Next, for SUPPLIER table, there is a non-key attribute (pic_name) which is dependent on the other non-key attribute (pic_contact_no)

supplier_id	**supplier_name**	**address**	**pic_name**	**pic_contact_no**
0521	Ban Huat	50,Jalan Temenggung	PA003	016-8582869
0369	Bermaz Auto	20, Jalan Banggol	PA009	016-2365858
0428	Kian Giap Motor	39, Jalan Kelana	PA011	012-8596321

After normalization, original SUPPLIER table is converted into 2 tables as below :

SUPPLIER (**supplier_id**, supplier_name, address)
PIC (**pic_contact_no**, pic_name)

SUPPLIER table:

supplier_id	**supplier_name**	**address**
0521	Ban Huat	50,Jalan Temenggung
0369	Bermaz Auto	20, Jalan Banggol
0428	Kian Giap Motor	39, Jalan Kelana

PIC table:

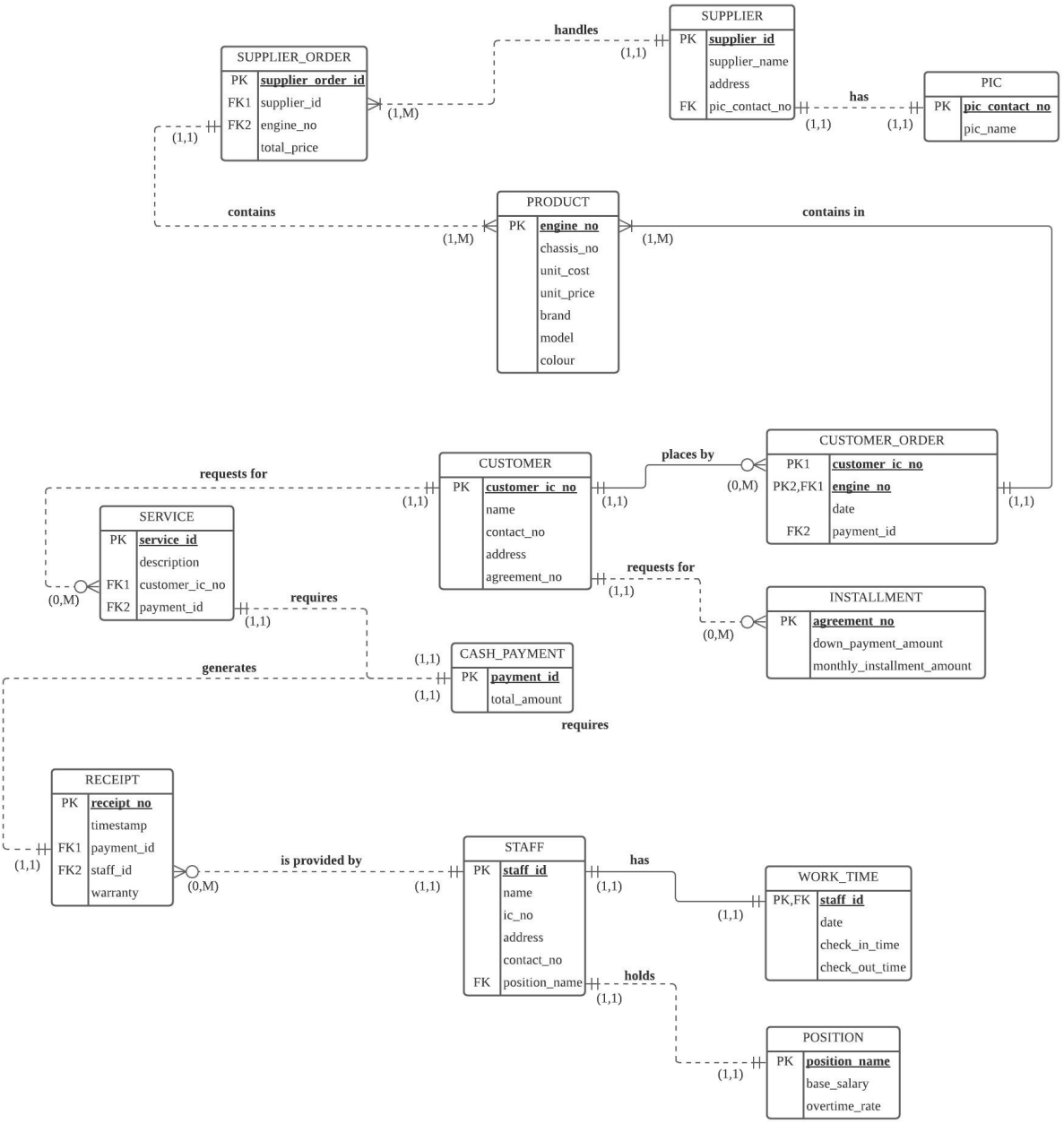
pic_contact_no	**pic_name**
016-8582869	PA003
016-2365858	PA009
012-8596321	PA011

At the end of the first normalization, the entities and attributes are below:
References: -> means "determines"

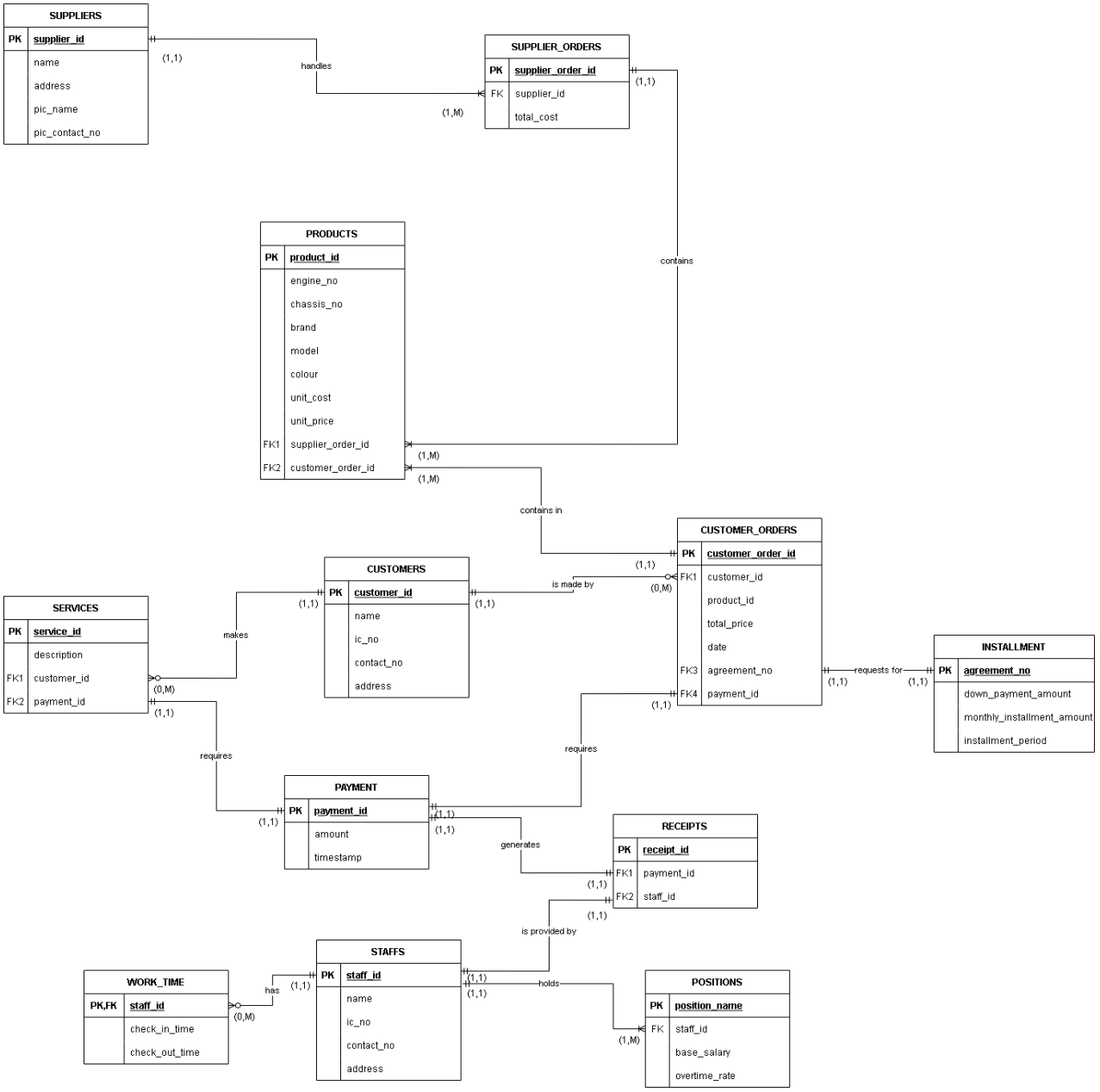
1. SUPPLIER (**supplier_id**, supplier_name, address, pic_contact_no)
2. SUPPLIER_ORDER (**supplier_order_id,** supplier_id, engine_no, total_price)
3. PRODUCT (**engine_no,** chassis_no, unit_cost, unit_price, brand, model, colour)
4. CUSTOMER_ORDER (**customer_ic_no**, **engine_no**, date, payment_id)
5. CUSTOMER (**customer_ic_no**, name, contact_no, address, agreement_no)

6. INSTALLMENT(**agreement_no**,down_payment_amount, monthly_instalment_amount)
7. SERVICE (**service_id**, description, customer_ic_no, payment_id)
8. CASH_PAYMENT (**payment_id**, total_amount)
9. RECEIPT (**receipt_no**, timestamp, payment_id, staff_id, warranty)
10. STAFF (**staff_id**, name, ic_no, address, contact_no, position_name)
11. WORK_TIME (**staff_id**, **date**, check_in_time, check_out_time)
12. POSITION (**position_name**, base_salary, overtime_rate)
13. PIC (**pic_contact_no**, pic_name)

7. Implementable Entity Relationship Diagram



Updated Implementable Entity Relationship Diagram



8. Sample SQL Statements

8.1 Data Definition Language (DDL) commands:

1. **CREATE**
   The CREATE TABLE command creates a new table in the database.

Table	SQL Command
suppliers The following SQL command creates a table called "***suppliers***" that contains five columns: ***supplier_id***, ***name***, ***address***, ***pic_name*** **and** ***pic_contact_no***	*CREATE TABLE "suppliers"* *(* *"supplier_id" NUMBER GENERATED ALWAYS AS IDENTITY,* *"name" VARCHAR2(100) NOT NULL,* *"address" VARCHAR2(200) NULL,* *"pic_name" VARCHAR2(100) NULL,* *"pic_contact_no" VARCHAR2(50) NULL,* *CONSTRAINT "suppliers_pk"* *PRIMARY KEY ("supplier_id")* *);*
customers The following SQL command creates a table called "***customers***" that contains five columns: ***customer_id***, ***name***, ***ic_no***, ***contact_no*** **and** ***address***	*CREATE TABLE "customers"* *(* *"customer_id" NUMBER GENERATED ALWAYS AS IDENTITY,* *"name" VARCHAR2(100) NOT NULL,* *"ic_no" VARCHAR2(20) NULL UNIQUE,* *"contact_no" VARCHAR2(20) NULL,* *"address" VARCHAR2(200) NULL,* *CONSTRAINT "customers_pk"* *PRIMARY KEY ("customer_id")* *);*
payment The following SQL command creates a table called "***payment***" that contains three columns: ***payment_id***, ***amount*** **and** ***timestamp***	*CREATE TABLE "payment"* *(* *"payment_id" NUMBER GENERATED ALWAYS AS IDENTITY,* *"amount" NUMBER(10,2) NOT NULL,* *"timestamp" TIMESTAMP NOT NULL,* *CONSTRAINT "payment_pk"* *PRIMARY KEY ("payment_id")* *);*
installments The following SQL command creates a table called "***installments***" that contains four columns: ***agreement_no***, ***down_payment_amount***, ***monthly_installment_amount*** **and** ***installment_period***	*CREATE TABLE "installments"* *(* *"agreement_no" VARCHAR2(50),* *"down_payment_amount" NUMBER(10,2) NOT NULL,* *"monthly_installment_amount" NUMBER(10,2) NOT NULL,* *"installment_period" NUMBER(5,0),* *CONSTRAINT "installments_pk"* *PRIMARY KEY ("agreement_no")* *);*

staffs The following SQL command creates a table called "***staffs***" that contains five columns: ***staff_id***, ***name***, ***ic_no***, ***contact_no*** **and** ***address***	*CREATE TABLE "staffs"* *(* *"staff_id" NUMBER GENERATED ALWAYS AS IDENTITY,* *"name" VARCHAR2(100) NOT NULL,* *"ic_no" VARCHAR2(20) NOT NULL,* *"contact_no" VARCHAR2(20) NOT NULL,* *"address" VARCHAR2(200) NOT NULL,* *CONSTRAINT "staffs_pk"* *PRIMARY KEY ("staff_id")* *);*
work_time The following SQL command creates a table called "***work_time***" that contains three columns: ***staff_id***, ***check_in_time*** **and** ***check_out_time***	*CREATE TABLE "work_time"* *(* *"staff_id" NUMBER NOT NULL,* *"check_in_time" TIMESTAMP NOT NULL,* *"check_out_time" TIMESTAMP NULL,* *CONSTRAINT "work_time_pk"* *PRIMARY KEY ("staff_id"),* *CONSTRAINT "work_time_fk"* *FOREIGN KEY ("staff_id")* *REFERENCES "staffs" ("staff_id")* *);*
positions The following SQL command creates a table called "***positions***" that contains four columns: ***position_name***, ***staff_id***, ***base_salary*** **and** ***overtime_rate***	*CREATE TABLE "positions"* *(* *"position_name" VARCHAR2(50),* *"staff_id" NUMBER NULL,* *"base_salary" NUMBER(10,2) NOT NULL,* *"overtime_rate" NUMBER(10,2) NULL,* *CONSTRAINT "positions_pk"* *PRIMARY KEY ("position_name"),* *CONSTRAINT "positions_fk"* *FOREIGN KEY ("staff_id")* *REFERENCES "staffs" ("staff_id")* *);*
receipts The following SQL command creates a table called "***receipts***" that contains three columns: ***receipt_id***, ***payment_id*** **and** ***staff_id***	*CREATE TABLE "receipts"* *(* *"receipt_id" NUMBER GENERATED ALWAYS AS IDENTITY,* *"payment_id" NUMBER NOT NULL,* *"staff_id" NUMBER NOT NULL,* *CONSTRAINT "receipts_pk"* *PRIMARY KEY ("receipt_id"),* *CONSTRAINT "receipts_fk1"* *FOREIGN KEY ("payment_id")* *REFERENCES "payment" ("payment_id"),* *CONSTRAINT "receipts_fk2"* *FOREIGN KEY ("staff_id")* *REFERENCES "staffs" ("staff_id")* *);*

services The following SQL command creates a table called "***services***" that contains four columns: ***service_id***, ***description***, ***customer_id*** **and** ***payment_id***	*CREATE TABLE "services"* *(* *"service_id" NUMBER GENERATED ALWAYS AS IDENTITY,* *"description" VARCHAR2(200) NOT NULL,* *"customer_id" NUMBER NOT NULL,* *"payment_id" NUMBER NOT NULL,* *CONSTRAINT "services_pk"* *PRIMARY KEY ("service_id"),* *CONSTRAINT "services_fk1"* *FOREIGN KEY ("customer_id")* *REFERENCES "customers" ("customer_id"),* *CONSTRAINT "services_fk2"* *FOREIGN KEY ("payment_id")* *REFERENCES "payment" ("payment_id")* *);*
supplier_orders The following SQL command creates a table called "***supplier_orders***" that contains three columns: ***supplier_order_id***, ***supplier_id*** **and** ***total_cost***	*CREATE TABLE "supplier_orders"* *(* *"supplier_order_id" NUMBER GENERATED ALWAYS AS IDENTITY,* *"supplier_id" NUMBER NOT NULL,* *"total_cost" NUMBER(10,2) NULL,* *CONSTRAINT "supplier_orders_pk"* *PRIMARY KEY ("supplier_order_id"),* *CONSTRAINT "supplier_order_fk"* *FOREIGN KEY ("supplier_id")* *REFERENCES "suppliers" ("supplier_id")* *);*
customer_orders The following SQL command creates a table called "***customer_orders***" that contains seven columns: ***customer_order_id***, ***customer_id***, ***product_id***, ***total_price, date, agreement_no*** **and** ***payment_id***	*CREATE TABLE "customer_orders"* *(* *"customer_order_id" NUMBER GENERATED ALWAYS AS IDENTITY,* *"customer_id" NUMBER NOT NULL,* *"product_id" NUMBER NOT NULL,* *"total_price" NUMBER NULL,* *"date" DATE NOT NULL,* *"agreement_no" VARCHAR2(50) NULL,* *"payment_id" NUMBER NULL,* *CONSTRAINT "customer_orders_pk"* *PRIMARY KEY ("customer_order_id"),* *CONSTRAINT "customer_orders_fk1"* *FOREIGN KEY ("customer_id")* *REFERENCES "customers" ("customer_id"),* *CONSTRAINT "customer_orders_fk2"* *FOREIGN KEY ("agreement_no")* *REFERENCES "installments" ("agreement_no"),* *CONSTRAINT "customer_orders_fk3"* *FOREIGN KEY ("payment_id")* *REFERENCES "payment"*

	("payment_id") *);*
products The following SQL command creates a table called "***products***" that contains ten columns: ***product_id***, ***engine_no***, ***chassis_no***, ***brand, model, colour, unit_cost, unit_price, supplier_order_id*** **and** ***customer_order_id***	*CREATE TABLE "products"* *(* *"product_id" NUMBER GENERATED ALWAYS AS IDENTITY,* *"engine_no" VARCHAR2(50) NOT NULL UNIQUE,* *"chassis_no" VARCHAR2(50) NOT NULL UNIQUE,* *"brand" VARCHAR2(50) NOT NULL,* *"model" VARCHAR2(50) NOT NULL UNIQUE,* *"colour" VARCHAR2(50) NULL,* *"unit_cost" NUMBER(10,2) NOT NULL,* *"unit_price" NUMBER(10,2) NOT NULL,* *"supplier_order_id" NUMBER NOT NULL,* *"customer_order_id" NUMBER NULL,* *CONSTRAINT "products_pk"* *PRIMARY KEY ("product_id"),* *CONSTRAINT "products_fk1"* *FOREIGN KEY ("supplier_order_id")* *REFERENCES "supplier_orders" ("supplier_order_id"),* *CONSTRAINT "products_fk2"* *FOREIGN KEY ("customer_order_id")* *REFERENCES "customer_orders" ("customer_order_id")* *);*

"suppliers" table

	COLUMN_NAME	DATA_TYPE	NULLABLE	DATA_DEFAULT	COLUMN_ID	COMMENTS
1	supplier_id	NUMBER	No	"DEMOUSER"."ISEQ$$_79289".nextval	1	(null)
2	name	VARCHAR2(100 BYTE)	No	(null)	2	(null)
3	address	VARCHAR2(200 BYTE)	Yes	(null)	3	(null)
4	pic_name	VARCHAR2(100 BYTE)	Yes	(null)	4	(null)
5	pic_contact_no	VARCHAR2(50 BYTE)	Yes	(null)	5	(null)

"customers" table

	COLUMN_NAME	DATA_TYPE	NULLABLE	DATA_DEFAULT	COLUMN_ID	COMMENTS
1	customer_id	NUMBER	No	"DEMOUSER"."ISEQ$$_79292".nextval	1	(null)
2	name	VARCHAR2(100 BYTE)	No	(null)	2	(null)
3	ic_no	VARCHAR2(20 BYTE)	Yes	(null)	3	(null)
4	contact_no	VARCHAR2(20 BYTE)	Yes	(null)	4	(null)
5	address	VARCHAR2(200 BYTE)	Yes	(null)	5	(null)

"payment" table

	COLUMN_NAME	DATA_TYPE	NULLABLE	DATA_DEFAULT	COLUMN_ID	COMMENTS
1	payment_id	NUMBER	No	"DEMOUSER"."ISEQ$$_79296".nextval	1	(null)
2	amount	NUMBER(10,2)	No	(null)	2	(null)
3	timestamp	TIMESTAMP(6)	No	(null)	3	(null)

"installments" table

	COLUMN_NAME	DATA_TYPE	NULLABLE	DATA_DEFAULT	COLUMN_ID	COMMENTS
1	agreement_no	VARCHAR2(50 BYTE)	No	(null)	1	(null)
2	down_payment_amount	NUMBER(10,2)	No	(null)	2	(null)
3	monthly_installment_amount	NUMBER(10,2)	No	(null)	3	(null)
4	installment_period	NUMBER(5,0)	Yes	(null)	4	(null)

“staffs” table

	COLUMN_NAME	DATA_TYPE	NULLABLE	DATA_DEFAULT	COLUMN_ID	COMMENTS
1	staff_id	NUMBER	No	"DEMOUSER"."ISEQ$$_79301".nextval	1	(null)
2	name	VARCHAR2(100 BYTE)	No	(null)	2	(null)
3	ic_no	VARCHAR2(20 BYTE)	No	(null)	3	(null)
4	contact_no	VARCHAR2(20 BYTE)	No	(null)	4	(null)
5	address	VARCHAR2(200 BYTE)	No	(null)	5	(null)

“work_time” table

	COLUMN_NAME	DATA_TYPE	NULLABLE	DATA_DEFAULT	COLUMN_ID	COMMENTS
1	staff_id	NUMBER	No	(null)	1	(null)
2	check_in_time	TIMESTAMP(6)	No	(null)	2	(null)
3	check_out_time	TIMESTAMP(6)	Yes	(null)	3	(null)

“positions” table

	COLUMN_NAME	DATA_TYPE	NULLABLE	DATA_DEFAULT	COLUMN_ID	COMMENTS
1	position_name	VARCHAR2(50 BYTE)	No	(null)	1	(null)
2	staff_id	NUMBER	Yes	(null)	2	(null)
3	base_salary	NUMBER(10,2)	No	(null)	3	(null)
4	overtime_rate	NUMBER(10,2)	Yes	(null)	4	(null)

“receipts” table

	COLUMN_NAME	DATA_TYPE	NULLABLE	DATA_DEFAULT	COLUMN_ID	COMMENTS
1	receipt_id	NUMBER	No	"DEMOUSER"."ISEQ$$_79308".nextval	1	(null)
2	payment_id	NUMBER	No	(null)	2	(null)
3	staff_id	NUMBER	No	(null)	3	(null)

“services” table

	COLUMN_NAME	DATA_TYPE	NULLABLE	DATA_DEFAULT	COLUMN_ID	COMMENTS
1	service_id	NUMBER	No	"DEMOUSER"."ISEQ$$_79311".nextval	1	(null)
2	description	VARCHAR2(200 BYTE)	No	(null)	2	(null)
3	customer_id	NUMBER	No	(null)	3	(null)
4	payment_id	NUMBER	No	(null)	4	(null)

“supplier_orders” table

	COLUMN_NAME	DATA_TYPE	NULLABLE	DATA_DEFAULT	COLUMN_ID	COMMENTS
1	supplier_order_id	NUMBER	No	"DEMOUSER"."ISEQ$$_79314".nextval	1	(null)
2	supplier_id	NUMBER	No	(null)	2	(null)
3	total_cost	NUMBER(10,2)	Yes	(null)	3	(null)

“customer_orders” table

	COLUMN_NAME	DATA_TYPE	NULLABLE	DATA_DEFAULT	COLUMN_ID	COMMENTS
1	customer_order_id	NUMBER	No	"DEMOUSER"."ISEQ$$_79317".nextval	1	(null)
2	customer_id	NUMBER	No	(null)	2	(null)
3	product_id	NUMBER	No	(null)	3	(null)
4	total_price	NUMBER	Yes	(null)	4	(null)
5	date	DATE	No	(null)	5	(null)
6	agreement_no	VARCHAR2(50 BYTE)	Yes	(null)	6	(null)
7	payment_id	NUMBER	Yes	(null)	7	(null)

“products” table

	COLUMN_NAME	DATA_TYPE	NULLABLE	DATA_DEFAULT	COLUMN_ID	COMMENTS
1	product_id	NUMBER	No	"DEMOUSER"."ISEQ$$_79320".nextval	1	(null)
2	engine_no	VARCHAR2(50 BYTE)	No	(null)	2	(null)
3	chassis_no	VARCHAR2(50 BYTE)	No	(null)	3	(null)
4	brand	VARCHAR2(50 BYTE)	No	(null)	4	(null)
5	model	VARCHAR2(50 BYTE)	No	(null)	5	(null)
6	colour	VARCHAR2(50 BYTE)	Yes	(null)	6	(null)
7	unit_cost	NUMBER(10,2)	No	(null)	7	(null)
8	unit_price	NUMBER(10,2)	No	(null)	8	(null)
9	supplier_order_id	NUMBER	No	(null)	9	(null)
10	customer_order_id	NUMBER	Yes	(null)	10	(null)

2. ALTER

The ALTER TABLE command adds, deletes, or modifies columns in a table.
For example in "*customers*" table,

Original:

	COLUMN_NAME	DATA_TYPE	NULLABLE	DATA_DEFAULT	COLUMN_ID	COMMENTS
1	customer_id	NUMBER	No	"DEMOUSER"."ISEQ$$_79485".nextval	1	(null)
2	name	VARCHAR2(100 BYTE)	No	(null)	2	(null)
3	ic_no	VARCHAR2(20 BYTE)	Yes	(null)	3	(null)
4	contact_no	VARCHAR2(20 BYTE)	Yes	(null)	4	(null)
5	address	VARCHAR2(200 BYTE)	Yes	(null)	5	(null)

	customer_id	name	ic_no	contact_no	address
1	1	Hawk	870412030523	0187878562	NO 123, Jalan Kamunting, 46200 Petaling Jaya, Selangor
2	2	John	670420035285	0175656438	(null)
3	3	Ahmad	590303036363	(null)	NO 789, Jalan Kemuning, 46200 Petaling Jaya, Selangor

The following SQL adds an "email" column to the "customers" table:

ALTER TABLE "DEMOUSER"."customers"
ADD "email" varchar2(255);

After ALTER (ADD) :

	COLUMN_NAME	DATA_TYPE	NULLABLE	DATA_DEFAULT	COLUMN_ID	COMMENTS
1	customer_id	NUMBER	No	"DEMOUSER"."ISEQ$$_79409".nextval	1	(null)
2	name	VARCHAR2(100 BYTE)	No	(null)	2	(null)
3	ic_no	VARCHAR2(20 BYTE)	Yes	(null)	3	(null)
4	contact_no	VARCHAR2(20 BYTE)	Yes	(null)	4	(null)
5	address	VARCHAR2(200 BYTE)	Yes	(null)	5	(null)
6	email	VARCHAR2(255 BYTE)	Yes	(null)	6	(null)

	customer_id	name	ic_no	contact_no	address	email
1	1	Hawk	870412030523	0187878562	NO 123, Jalan Kamunting, 46200 Petaling Jaya, Selangor	(null)
2	2	John	670420035285	0175656438	(null)	(null)
3	3	Ahmad	590303036363	(null)	NO 789, Jalan Kemuning, 46200 Petaling Jaya, Selangor	(null)

The following SQL deletes the "email" column from the "customers" table:

ALTER TABLE "DEMOUSER"."customers"
DROP COLUMN "email";

After ALTER (DROP) :

	COLUMN_NAME	DATA_TYPE	NULLABLE	DATA_DEFAULT	COLUMN_ID	COMMENTS
1	customer_id	NUMBER	No	"DEMOUSER"."ISEQ$$_79409".nextval	1	(null)
2	name	VARCHAR2(100 BYTE)	No	(null)	2	(null)
3	ic_no	VARCHAR2(20 BYTE)	Yes	(null)	3	(null)
4	contact_no	VARCHAR2(20 BYTE)	Yes	(null)	4	(null)
5	address	VARCHAR2(200 BYTE)	Yes	(null)	5	(null)

	customer_id	name	ic_no	contact_no	address
1	1	Hawk	870412030523	0187878562	NO 123, Jalan Kamunting, 46200 Petaling Jaya, Selangor
2	2	John	670420035285	0175656438	(null)
3	3	Ahmad	590303036363	(null)	NO 789, Jalan Kemuning, 46200 Petaling Jaya, Selangor

3. DROP

The DROP TABLE command deletes a table in the database.
The following SQL deletes all the tables:

```
DROP TABLE "products";
DROP TABLE "customer_orders";
DROP TABLE "supplier_orders";
DROP TABLE "services";
DROP TABLE "receipts";
DROP TABLE "positions";
DROP TABLE "work_time";
DROP TABLE "staffs";
DROP TABLE "installments";
DROP TABLE "payment";
DROP TABLE "customers";
DROP TABLE "suppliers";
```

4. TRUNCATE

The TRUNCATE TABLE command deletes the data inside a table, but not the table itself.
For example in "*receipts*" table,

Original:

	receipt_id	payment_id	staff_id
1	1	1	1
2	2	2	2
3	3	1	1

The following SQL truncates the table "*receipts*":

```
TRUNCATE TABLE "DEMOUSER"."receipts";
```

After TRUNCATE

	receipt_id	payment...	staff_id

5. COMMENT

Comments are used to explain sections of SQL statements, or to prevent execution of SQL statements.

Single Line Comments
Single line comments start with --.
Any text between -- and the end of the line will be ignored (will not be executed).

EXAMPLES:
The following example uses a single-line comment as an explanation:

```
--Select all:
```

The following example uses a single-line comment to ignore the end of a line:

```
SELECT * FROM "DEMOUSER"."customers" -- WHERE "name"='Hawk';
```

The following example uses a single-line comment to ignore a statement:

```
--SELECT * FROM "DEMOUSER"."customers";
```

Multi-line Comments
Multi-line comments start with /* and end with */.
Any text between /* and */ will be ignored.

EXAMPLES:
The following example uses a multi-line comment as an explanation:

```
/*Select all the columns
of all the records
in the Customers table:*/
SELECT * FROM "DEMOUSER"."customers";
```

The following example uses a multi-line comment to ignore many statements:

```
/*SELECT * FROM "DEMOUSER"."customers";
SELECT * FROM "DEMOUSER"."products";
SELECT * FROM "DEMOUSER"."staffs";*/
SELECT * FROM "DEMOUSER"."suppliers";
```

To ignore just a part of a statement, also use the /* */ comment.
The following example uses a comment to ignore part of a line:

```
SELECT "name", /*"contact_no",*/ "address" FROM "DEMOUSER"."customers";
```

6. RENAME

RENAME command is used to change the name of the table or a database object.
The following SQL rename the *"work_time" table to "workTime".*

```
RENAME "work_time" TO "workTime";
```

8.2 Data Manipulation Language (DML) commands:

1. **INSERT**
   The INSERT INTO statement is used to insert new records in a table.
   For example, the following sql command adds 1 staff to the shop:

   INSERT INTO "DEMOUSER"."staffs" ("name", "ic_no", "contact_no", "address")
   VALUES ('Tee Fen', '001121070707', '0123456789', 'NO 701, Jalan Tepi Pantai, 46200 Petaling Jaya, Selangor');

	staff_id	name	ic_no	contact_no	address
1	1	Tee Fen	001121070707	0123456789	NO 701, Jalan Tepi Pantai, 46200 Petaling Jaya, Selangor

2. **UPDATE**
   The UPDATE statement is used to modify the existing records in a table.
   For example, the following sql command updates the data in the "*supplier*" table.

   Original:

	supplier_id	name	address	pic_name	pic_contact_no
1	1	Kee Tech Motor Sdn Bhd	NO 388, Jalan Kenari, 46200 Petaling Jaya, Selangor	Tan Bee Bee	0128787566
2	2	Kian Shyan Motor Sdn Bhd	NO 466, Jalan Timan, 46200 Petaling Jaya, Selangor	(null)	(null)
3	3	Bauto Motor Sdn Bhd	(null)	See Won	0165813566

   The following SQL statement updates the first supplier (supplier_id = 1) with a new name:
   UPDATE "DEMOUSER"."suppliers"
   SET "name"='Jonathon'
   WHERE "supplier_id" = 1;

   After UPDATE:

	supplier_id	name	address	pic_name	pic_contact_no
1	1	Jonathon	NO 388, Jalan Kenari, 46200 Petaling Jaya, Selangor	Tan Bee Bee	0128787566
2	2	Kian Shyan Motor Sdn Bhd	NO 466, Jalan Timan, 46200 Petaling Jaya, Selangor	(null)	(null)
3	3	Bauto Motor Sdn Bhd	(null)	See Won	0165813566

3. **DELETE**
   The DELETE statement is used to delete existing records in a table.

   Original:

	position_name	staff_id	base_salary	overtime_rate
1	Sales Assistant	1	2500	(null)
2	Manager	2	8000	(null)
3	Accountant	3	3500	(null)

   For example, the following SQL statement deletes the *position_name* "Sales Assistant" from the "*positions*" table:
   DELETE FROM "DEMOUSER"."positions"
   WHERE "position_name"='Sales Assistant';

   After DELETE:

	position_name	staff_id	base_salary	overtime_rate
1	Manager	2	8000	(null)
2	Accountant	3	3500	(null)

4. JOIN

The JOIN clause is used to combine rows from two or more tables, based on a related column between them.

In the "supplier" table:

	supplier_id	name	address	pic_name	pic_contact_no
1	1	Jonathon	NO 388, Jalan Kenari, 46200 Petaling Jaya, Selangor	Tan Bee Bee	0128787566
2	2	Kian Shyan Motor Sdn Bhd	NO 466, Jalan Timan, 46200 Petaling Jaya, Selangor	(null)	(null)
3	3	Bauto Motor Sdn Bhd	(null)	See Won	0165813566

Next, look at another section from the "supplier_order" table:

	supplier_order_id	supplier_id	total_cost
1	1	1	8900
2	2	2	8900
3	3	3	8900

We can see that the "supplier_id" column in the "supplier" table refers to the "supplier_id" in the "supplier_order" table. There is a relationship between the two tables above which is the "supplier_id" column.

Hence, we can create the following SQL statement (that contains an INNER JOIN), that selects records that have matching values- 'supplier_id' in both tables:

```
SELECT s."supplier_id", s."name", s."address", sod."total_cost"
FROM "DEMOUSER"."suppliers" s
INNER JOIN "DEMOUSER"."supplier_orders" sod ON sod."supplier_id"=
s."supplier_id";
```

Output:

	supplier_id	name	address	total_cost
1	1	Jonathon	NO 388, Jalan Kenari, 46200 Petaling Jaya, Selangor	8900
2	2	Kian Shyan Motor Sdn Bhd	NO 466, Jalan Timan, 46200 Petaling Jaya, Selangor	8900
3	3	Bauto Motor Sdn Bhd	(null)	8900

5. COUNT

The COUNT() function returns the number of rows that matches a specified criterion.

	st...	name	ic_no	contact_no	address
1	1	Tee Fen	001121070707	0123456789	NO 701, Jalan Tepi Pantai, 46200 Petaling Jaya, Selangor
2	2	Kei Yang	011121030507	0135956789	NO 606, Jalan Atas Pantai, 46200 Petaling Jaya, Selangor
3	3	Jun Jie	001120070505	0123585889	NO 801, Jalan Gajah, 46200 Petaling Jaya, Selangor

For instance, the following SQL statement finds the number of staff:

```
SELECT COUNT("staff_id")
FROM "staffs";
```

	COUNT("STAFF_ID")
1	3

6. MAX

The MAX() function returns the largest value of the selected column.
The following SQL statement finds the price of the most expensive product:

	product_id	engine_no	chassis_no	brand	model	colour	unit_cost	unit_price	supplier_order_id	customer_order_id
1	1	83901283018	7839217398	HONDA	C100B	BLUE	4500	5000	1	(null)
2	2	83901283818	6824217398	KAWASAKI	Z100	GREEN	23000	28000	2	(null)
3	3	83901569818	6824333398	YAMAHA	A10	WHITE	8900	9200	3	(null)

SELECT MAX("unit_price") AS HighestPrice
FROM "products";

	HIGHESTPRICE
1	28000

7. MIN

The MIN() function returns the smallest value of the selected column.
The following SQL statement finds the price of the cheapest product:

	product_id	engine_no	chassis_no	brand	model	colour	unit_cost	unit_price	supplier_order_id	customer_order_id
1	1	83901283018	7839217398	HONDA	C100B	BLUE	4500	5000	1	(null)
2	2	83901283818	6824217398	KAWASAKI	Z100	GREEN	23000	28000	2	(null)
3	3	83901569818	6824333398	YAMAHA	A10	WHITE	8900	9200	3	(null)

SELECT MIN("unit_price") AS LowestPrice
FROM "products";

	LOWESTPRICE
1	5000

8. Queries that the company may perform (using SELECT keyword)

1. View the suppliers details and the total_cost made from each supplier in the database.

   SQL Statements:

   SELECT s."supplier_id", s."name" AS "supplier_name", s."address" AS "supplier_address", sod."total_cost"
   FROM "DEMOUSER"."suppliers" s
   INNER JOIN "DEMOUSER"."supplier_orders" sod ON sod."supplier_id"= s."supplier_id";

```
SELECT s."supplier_id", s."name" AS "supplier_name", s."address" AS "supplier_address", sod."total_cost"
FROM "DEMOUSER"."suppliers" s
INNER JOIN "DEMOUSER"."supplier_orders" sod ON sod."supplier_id"= s."supplier_id";
```

   Output:

	supplier_id	supplier_name	supplier_address	total_cost
1	1	Jonathon	NO 388, Jalan Kenari, 46200 Petaling Jaya, Selangor	8900
2	2	Kian Shyan Motor Sdn Bhd	NO 466, Jalan Timan, 46200 Petaling Jaya, Selangor	8900
3	3	Bauto Motor Sdn Bhd	(null)	8900

2. View the product that is contained in each supplier_order.

   SQL Statements:

   SELECT sod."supplier_id", p."product_id", p."brand", p."model", p."colour"
   FROM "supplier_orders" sod
   INNER JOIN "products" p ON sod."supplier_order_id" = p."supplier_order_id";

```
SELECT sod."supplier_id", p."product_id", p."brand", p."model", p."colour"
FROM "supplier_orders" sod
INNER JOIN "products" p ON sod."supplier_order_id" = p."supplier_order_id";
```

   Output:

	supplier_id	product_id	brand	model	colour
1	1	1	HONDA	C100B	BLUE
2	2	2	KAWASAKI	Z100	GREEN
3	3	3	YAMAHA	A10	WHITE

3. View the specifications of the product ordered by each customer.

   SQL Statements:

   SELECT c."name" AS "customer_name", p."brand", p."model", p."colour"
   FROM "customer_orders" cod
   INNER JOIN "customers" c ON cod."customer_id" = c."customer_id"
   INNER JOIN "products" p ON cod."customer_order_id" = p."customer_order_id";

```
SELECT c."name" AS "customer_name", p."brand", p."model", p."colour"
FROM "customer_orders" cod
INNER JOIN "customers" c ON cod."customer_id" = c."customer_id"
INNER JOIN "products" p ON cod."customer_order_id" = p."customer_order_id";
```

   Output:

	customer_name	brand	model	colour
1	Hawk	HONDA	C100B	BLUE
2	John	KAWASAKI	Z100	GREEN
3	Ahmad	YAMAHA	A10	WHITE

4. View the staff that serve the customers.

SQL Statements:

SELECT c."name" AS "customer_name", s."name" AS "staff_name" FROM "receipts" r
INNER JOIN "payment" p ON r."payment_id" = p."payment_id"
INNER JOIN "customer_orders" cod ON cod."payment_id" = r."payment_id"
INNER JOIN "customers" c ON c."customer_id" = cod."customer_id"
INNER JOIN "staffs" s ON r."staff_id" = s."staff_id";

```
SELECT c."name" AS "customer_name", s."name" AS "staff_name" FROM "receipts" r
INNER JOIN "payment" p ON r."payment_id" = p."payment_id"
INNER JOIN "customer_orders" cod ON cod."payment_id" = r."payment_id"
INNER JOIN "customers" c ON c."customer_id" = cod."customer_id"
INNER JOIN "staffs" s ON r."staff_id" = s."staff_id";
```

Output:

	customer_name	staff_name
1	Hawk	Tee Fen
2	John	Kei Yang
3	Ahmad	Tee Fen

5. View the receipt issued along with the staff who issues it.

SQL Statements:

SELECT r."receipt_id", s."name" AS "staff_name" FROM "receipts" r
INNER JOIN "staffs" s ON r."staff_id" = s."staff_id";

```
SELECT r."receipt_id", s."name" AS "staff_name" FROM "receipts" r
INNER JOIN "staffs" s ON r."staff_id" = s."staff_id";
```

Output:

	receipt_id	staff_name
1	1	Tee Fen
2	2	Kei Yang
3	3	Tee Fen

9. Screenshots of Tables and Queries' Output of the Developed System

In SUPPLIERS table,

	supplier_id	name	address	pic_name	pic_contact_no
1	1	Kee Tech Motor Sdn Bhd	NO 388, Jalan Kenari, 46200 Petaling Jaya, Selangor	Tan Bee Bee	0128787566
2	2	Kian Shyan Motor Sdn Bhd	NO 466, Jalan Timan, 46200 Petaling Jaya, Selangor	(null)	(null)
3	3	Bauto Motor Sdn Bhd	(null)	See Won	0165813566

In CUSTOMERS table,

	customer_id	name	ic_no	contact_no	address
1	1	Hawk	870412030523	0187878562	NO 123, Jalan Kamunting, 46200 Petaling Jaya, Selangor
2	2	John	670420035285	0175656438	(null)
3	3	Ahmad	590303036363	(null)	NO 789, Jalan Kemuning, 46200 Petaling Jaya, Selangor

In PAYMENT table,

	payment_id	amount	timestamp
1	1	5000	16/01/22 12:38:54.142000000 AM
2	2	28000	16/01/22 12:38:54.170000000 AM
3	3	9200	16/01/22 12:38:54.191000000 AM

In INSTALLMENT table,

	agreement_no	down_payment_amount	monthly_installment_amount	installment_period
1	8409238043298	1000	200	60
2	7779236943298	16000	400	30
3	8409238052893	2900	105	60

In STAFFS table,

	staff_id	name	ic_no	contact_no	address
1	1	Tee Fen	001121070707	0123456789	NO 701, Jalan Tepi Pantai, 46200 Petaling Jaya, Selangor
2	2	Kei Yang	011121030507	0135956789	NO 606, Jalan Atas Pantai, 46200 Petaling Jaya, Selangor
3	3	Jun Jie	001120070505	0123585889	NO 801, Jalan Gajah, 46200 Petaling Jaya, Selangor

In WORK_TIME table,

	staff_id	check_in_time	check_out_time
1	1	16/01/22 12:38:55.023000000 AM	16/01/22 12:38:55.253000000 AM
2	2	16/01/22 12:38:55.055000000 AM	16/01/22 12:38:55.277000000 AM
3	3	16/01/22 12:38:55.075000000 AM	16/01/22 12:38:55.295000000 AM

In POSITIONS table,

	position_name	staff_id	base_salary	overtime_rate
1	Sales Assistant	1	2500	(null)
2	Manager	2	8000	(null)
3	Accountant	3	3500	(null)

In RECEIPTS table,

	receipt_id	payment_id	staff_id
1	1	1	1
2	2	2	2
3	3	3	1

In SERVICES table,

	service_id	description	customer_id	payment_id
1	1	change gear oil	1	1
2	2	change engine oil	2	2

In SUPPLIER_ORDERS table,

	supplier_order_id	supplier_id	total_cost
1	1	1	8900
2	2	2	8900
3	3	3	8900

In CUSTOMER_ORDERS table,

	customer_order_id	customer_id	product_id	total_price	date	agreement_no	payment_id
1	1	1	1	5000	16/01/22	8409238043298	1
2	2	2	2	28000	16/01/22	7779236943298	2
3	3	3	3	9200	16/01/22	8409238052893	3

In PRODUCTS table,

	product_id	engine_no	chassis_no	brand	model	colour	unit_cost	unit_price	supplier_order_id	customer_order_id
1	1	83901283018	7839217398	HONDA	C100B	BLUE	4500	5000	1	1
2	2	83901283818	6824217398	KAWASAKI	Z100	GREEN	23000	28000	2	2
3	3	83901569818	6824333398	YAMAHA	A10	WHITE	8900	9200	3	3

10. Testing

Source code	Result
INSERT INTO "DEMOUSER"."products" ("engine_no", "chassis_no", "brand", "model", "colour", "unit_cost", "unit_price", "supplier_order_id") *VALUES ('83901283018', '89423094820', 'YAMAHA', 'Y15ZR', 'GP BLUE', '7800', '11000', 1);*	*Error starting at line : 273 in command -* *INSERT INTO "DEMOUSER"."products" ("engine_no", "chassis_no", "brand", "model", "colour", "unit_cost", "unit_price", "supplier_order_id")* *VALUES ('83901283018', '89423094820', 'YAMAHA', 'Y15ZR', 'GP BLUE', '7800', '11000', 1)* *Error report -* *ORA-00001: unique constraint (DEMOUSER.SYS_C009337) violated* **Reason:** The “engine_no” of the product is the same as the one previously inserted product, because “engine_no” has a “unique” constraint, this act violated the constraint and thus cannot be executed successfully.
INSERT INTO "DEMOUSER"."customers" ("name", "ic_no") *VALUES ('Ben', '870412030523');*	*Error starting at line : 277 in command -* *INSERT INTO "DEMOUSER"."customers" ("name", "ic_no")* *VALUES ('Ben', '870412030523')* *Error report -* *ORA-00001: unique constraint (DEMOUSER.SYS_C009282) violated* **Reason:** The “ic_no” of the product is the same as the one previously inserted customer’s ic number, because “ic_no” has a “unique” constraint, this act violated the constraint and thus cannot be executed successfully.
INSERT INTO "DEMOUSER"."suppliers" ("address", "pic_name", "pic_contact_no") *VALUES ('NO 508, Jalan Damai, 46200 Petaling Jaya, Selangor', 'See Won', '0165813566');*	*Error starting at line : 14 in command -* *INSERT INTO "DEMOUSER"."suppliers" ("address", "pic_name", "pic_contact_no")* *VALUES ('NO 508, Jalan Damai, 46200 Petaling Jaya, Selangor', 'See Won', '0165813566')* *Error report -* *ORA-01400: cannot insert NULL into ("DEMOUSER"."suppliers"."name")* **Reason:** The attribute “name” had not been inserted into the “suppliers” table. This act violated the constraint and thus cannot be executed successfully because the “name” of table “suppliers” had been set to NOT NULL.

11. Discussion

We have learnt a lot through this project like new skills and grown in numerous ways as a result of it.

From the start of this project, we were able to develop actual life skills. We were first concerned about the lecturer's request because we needed to blend race and gender in group formation before beginning the project. However, we all came to an agreement and saw this as a life challenge because we had to work with individuals from other backgrounds, genders, and ethnicities and could not live in our own circle. We kept everyone's spirits up during the process by having group conversations and discussions.There are few challenged faces and future improvements we include below.

11.1 Challenge(s) faced

At first, we were trying to approach a few companies for collaboration, but unfortunately, all of them did not reply to our email. We were glad to contact the owner of Akenz Motor, Mr. Liew, who is suggested by one of our group members. He agreed with the collaboration and we proceeded to the interview with him immediately.

In terms of communicating with stakeholders, we had faced communication problems. Firstly, the Akenz Motor owner, Mr. Liew Yoon, was using Malay as the medium of communication. Since we are not proficient in using Malay to speak, the communication process encountered translation problems as the question to interview and the answer got from him need to be translated instantly for record purpose. The interview duration increased due to many clarification processes. Moreover, the owner faced a technical issue, which was that at the beginning of the interview, his microphone was unusable, but it was resolved after a few minutes.

During the implementation of the database, the knowledge had limited us in developing the database as it was the first time for all our groupmates to get in touch with the database course. Firstly, we had spent a lot of time deciding data type and constraints. We got double confirmation with Mr. Liew on the data type he wished to assign to ensure the database can be ideally functioned for the company operation. Next, we spent a lot of time in the debugging process. For example, we encountered a failed creation of a table due to sequence error. Thus, we had spent more time in learning progress in W3school to overcome our limitation in knowledge.

Furthermore, we had faced communication difficulties within group members as all of the group members are packed with different schedules making group meetings scheduling takes more time. Luckily, all the group members are willing to contribute to this project, in order to overcome this problem, communication within members is replaced by WhatsApp group texting instead of group meetings.

11.2 Future improvements

To facilitate communication with stakeholders, we can first confirm what language the stakeholders prefer to do more preparation for if the language is not in our proficiency. For example,

translation can be done earlier and multiple group members can take charge of taking notes from the conversation.

In the current project, there was still room for improvement in the ERD diagram, luckily we got advice from the lecturer during the presentation. The diagram was improved based on her advice in the current project. Thus, the project can be improved in the researching process. Before drawing an ERD diagram, Research and interview with stakeholders is a must to meet the stakeholder's requirements and reduce database redundancy. We should get more lecturer's advice on the drawing stage as the ERD diagram is the core for further implementation. Data type and constraints can also be determined during the early stage to ensure the effectiveness of the implementation process. Besides, we can do more research and study before the implementation phase as a preparation to equip ourselves with more knowledge and avoid problems regarding limitations in knowledge during the implementation phase. Besides, we can also seek for lecturer's help when we need clarification of database queries.

Furthermore, group meetings can also be held once per week to update each of the members' progress and also to allocate more time for group discussion, as face-to-face communication will be more effective than texting.

12. Conclusions

We have completed our Database assignment by using Akenz Motor as our chosen business to create and analyse database systems. To summarise, the project has been finished successfully, with the database being improved. During these months, we've sharpened our database design skills and successfully learned a new programming language, SQL, as well as a relational database management system, MySQL. We have gained tremendous experience by stepping outside of our comfort zones at university and truly immersing ourselves in real-life circumstances and solving real-life problems. We are grateful for the opportunity to take the course, and we hope to learn more about databases in the future. This project enables us to not only learn, but also to practise and apply what we've learned.We would like to thank to our lecturer, Profesor Madya Dr. Fariza Hanum Binti Md Nasaruddin for guiding us on our database and we would like to thank Mr Liew for giving us the opportunity to create a sample database for his business. Last but not least we would like to thank all of the group members for committing to completing the given task in the time given.

13. Appendix

Appendix A (Tasks distribution)

No.	Tasks	Member
1	Interview session with owner of Akenz Motor	● Lee Kai Yang ● Liew Rui Zhi ● Chow Shang Shyan ● Chin Chin Fang ● Wan Muhammad Azib
2	Report Writing: Introduction to business	● Lee Kai Yang
3	Report Writing: Objectives	● Liew Rui Zhi
4	Report Writing: Business operation	● Chow Shang Shyan
5	Report Writing: Issues in managing data	● Chin Chin Fang
6	Report Writing: Business rule	● Wan Muhammad Azib
7	Report Writing: User requirements	● Wan Muhammad Azib
8	Report Writing: Entities, Attributes, Relationships, Cardinalities, and Constraints	● Lee Kai Yang ● Liew Rui Zhi
9	Report Writing: Entity Relationship Diagram	● Liew Rui Zhi
10	Report Writing: Normalisation Process	● Chow Shang Shyan ● Chin Chin Fang
11	Develop database	● Lee Kai Yang ● Liew Rui Zhi ● Chow Shang Shyan
12	Check & format SQL queries Report Writing: Sample SQL Statement, Screenshot of Tables of the Developed System	● Lee Kai Yang ● Liew Rui Zhi ● Chow Shang Shyan ● Chin Chin Fang ● Wan Muhammad Azib
13	Report Writing: Testing	● Chow Shang Shyan
14	Report Writing: Discussion	● Chin Chin Fang
15	Report Writing: Conclusion	● Wan Muhammad Azib
16	Report Writing: Appendix	● Lee Kai Yang ● Chin Chin Fang
17	Finalisation of report	● Lee Kai Yang ● Liew Rui Zhi ● Chow Shang Shyan ● Chin Chin Fang ● Wan Muhammad Azib

Appendix B (Proof of interview with owner of Akenz Motor and meeting with group members)

Permission letter together with our lecturer, Dr Fariza's signature to request to use the company, *Akenz Motor* as a model for database project

Lee Kai Yang

Faculty of Computer Science & Information Technology
University of Malaya
50603 Kuala Lumpur
Malaysia

Mr. Liew

Akenz Motor,
No 71 Jalan Pasir Akar,
22000 Jertih,
Terengganu.

24 NOVEMBER 2021

Dear Mr. Liew,

Request to use your company as a model for database project

I am currently pursuing my Bachelor Degree of Computer Science/ Information Technology at the University of Malaya. Design and development of a database is one of the assignments which we have to complete as a group project. I am writing on behalf of my group members to inquire whether you would allow us to use your company as our model for studying the needs for a database.

For the database project, we would like to understand the workflow and data flow of your company and we do not need to go into details or to see any confidential parts of your system or your data. We would also appreciate it if you would acknowledge our discussions by signing our report.

We sincerely hope that you will agree to meet with us (on-line) and assist us in understanding the flow of data and the needs for data management in your company. This will help us to complete our database project. We look forward to receiving a positive and prompt reply.

Thank you

Yours sincerely

(LEE KAI YANG)

Database Lecturer

(DR FARIZA HANUM BINTI MD NASARUDDIN)

Screenshots of meeting with Mr. Liew, the owner of *Akenz Motor*

Peer Work Group Evaluation Forms

Name: Lee Kai Yang
Matric no: U2005398

Peer Work Group Evaluation Forms (To be treated as private and confidential, and to be submitted SEPARATELY to the lecturer)

Course: WIA2001 DATABASE Semester: Semester 1 Session: 2021/2022
Lecturer: Dr. Fariza
Assignment: ~~**Lab Practice**~~**/Group Project/**~~**Presentation**~~

Evaluator (Student's Name): Lee Kai Yang
Date: 17/1/2022

Group Members:

	Matrix Number	Name
1	U2005270	LIEW RUI ZHI
2	U2005283	CHOW SHANG SHYAN
3	U2005349	CHIN CHIN FANG
4	17207678	WAN MUHAMMAD AZIB

Directions: In the space below, honestly evaluate the work of other students in your group by answering **yes** or **no** and by using a scale from 1 to 3, **1 being poor, 2 being average, 3 being above average. Please circle your answer.**

		Group Member 1	Group Member 2	Group Member 3	Group Member 4
1	Did this group member complete his/her assigned tasks for the group	Yes ~~No~~	Yes ~~No~~	Yes ~~No~~	Yes ~~No~~
2	How would you rate the quality of this person's work	~~1 2~~ 3	~~1 2~~ 3	~~1 2~~ 3	~~1 2~~ 3
3	How would you rate the timeliness of the completion of the work?	~~1 2~~ 3	~~1 2~~ 3	~~1 2~~ 3	~~1 2~~ 3
4	How would you rate the accuracy of the work	~~1 2~~ 3	~~1 2~~ 3	~~1 2~~ 3	~~1 2~~ 3
5	Overall, how would you rank this group member's performance in the group?	~~1 2~~ 3	~~1 2~~ 3	~~1 2~~ 3	~~1 2~~ 3
6	Would you want to work with this person again? Explain why in the space below.	Yes ~~No~~	Yes ~~No~~	Yes ~~No~~	Yes ~~No~~
		All group members are good, hardworking and committed to work with the group assignment.			

Name: Chin Chin Fang
Matric no: U2005349

Course: WIA2001 DATABASE Semester: Semester 1 Session: 2021/2022
Lecturer: Dr. Fariza
Assignment: ~~**Lab Practice**~~**/Group Project/**~~**Presentation**~~

Evaluator (Student's Name): Chin Chin Fang
Date: 17/1/2022

Group Members:

	Matrix Number	Name
1	U2005398	Lee Kai Yang
2	U2005283	Chow Shang Shyan
3	U2005270	Liew Rui Zhi
4	17207678	Wan Muhammad Azib

Directions: In the space below, honestly evaluate the work of other students in your group by answering **yes** or **no** and by using a scale from 1 to 3, **1 being poor, 2 being average, 3 being above average. Please circle your answer.**

		Group Member 1	**Group Member 2**	**Group Member 3**	**Group Member 4**
1	Did this group member complete his/her assigned tasks for the group	Yes ~~No~~	Yes ~~No~~	Yes ~~No~~	Yes ~~No~~
2	How would you rate the quality of this person's work	~~1 2~~ 3	~~1 2~~ 3	~~1 2~~ 3	~~1 2~~ 3
3	How would you rate the timeliness of the completion of the work?	~~1 2~~ 3	~~1 2~~ 3	~~1 2~~ 3	~~1 2~~ 3
4	How would you rate the accuracy of the work	~~1 2~~ 3	~~1 2~~ 3	~~1 2~~ 3	~~1 2~~ 3
5	Overall, how would you rank this group member's performance in the group?	~~1 2~~ 3	~~1 2~~ 3	~~1 2~~ 3	~~1 2~~ 3
6	Would you want to work with this person again? Explain why in the space below.	Yes ~~No~~	Yes ~~No~~	Yes ~~No~~	Yes ~~No~~
		They are flexible to communicate and perform greatly in this project.			

Name: Chow Shang Shyan
Matric no: U2005283

Peer Work Group Evaluation Forms (To be treated as private and confidential, and to be submitted SEPARATELY to the lecturer)

Course: WIA2001 DATABASE Semester: Semester 1 Session: 2021/2022
Lecturer: Dr. Fariza
Assignment: ~~**Lab Practice**~~**/Group Project/**~~**Presentation**~~

Evaluator (Student's Name): Chow Shang Shyan
Date: 17/1/2022

Group Members:

	Matrix Number	Name
1	U2005270	Liew Rui Zhi
2	U2005398	Lee Kai Yang
3	U2005349	Chin Chin Fang
4	17207678	Wan Muhammad Azib

Directions: In the space below, honestly evaluate the work of other students in your group by answering **yes** or **no** and by using a scale from 1 to 3, **1 being poor, 2 being average, 3 being above average. Please circle your answer.**

		Group Member 1	Group Member 2	**Group Member 3**	**Group Member 4**
1	Did this group member complete his/her assigned tasks for the group	Yes ~~No~~	Yes ~~No~~	Yes ~~No~~	Yes ~~No~~
2	How would you rate the quality of this person's work	~~1 2~~ 3	~~1 2~~ 3	~~1 2~~ 3	~~1 2~~ 3
3	How would you rate the timeliness of the completion of the work?	~~1 2~~ 3	~~1 2~~ 3	~~1 2~~ 3	~~1 2~~ 3
4	How would you rate the accuracy of the work	~~1 2~~ 3	~~1 2~~ 3	~~1 2~~ 3	~~1 2~~ 3
5	Overall, how would you rank this group member's performance in the group?	~~1 2~~ 3	~~1 2~~ 3	~~1 2~~ 3	~~1 2~~ 3
6	Would you want to work with this person again? Explain why in the space below.	Yes ~~No~~	Yes ~~No~~	Yes ~~No~~	Yes ~~No~~
		All group members are cooperative, nice and committed to work with the group assignment.			

Name: Liew Rui Zhi
Matric no: U2005270

Peer Work Group Evaluation Forms (To be treated as private and confidential, and to be submitted SEPARATELY to the lecturer)

Course: **WIA2001 DATABASE**
Semester: **Semester 1**
Session: **2021/2022**
Lecturer: **DR. FARIZA**
Assignment: ~~**Lab Practice**~~**/Group Project/**~~**Presentation**~~

Evaluator (Student's Name): **LIEW RUI ZHI**
Date: **17/01/2022**

Group Members:

	Matrix Number	Name
1	**U2005398**	**LEE KAI YANG**
2	**U2005283**	**CHOW SHANG SHYAN**
3	**U2005349**	**CHIN CHIN FANG**
4	**17207678**	**WAN MUHAMMAD AZIB**

Directions: In the space below, honestly evaluate the work of other students in your group by answering **yes** or **no** and by using a scale from 1 to 3, **1 being poor, 2 being average, 3 being above average. Please circle your answer.**

		Group Member 1	**Group Member 2**	**Group Member 3**	**Group Member 4**
1	Did this group member complete his/her assigned tasks for the group	Yes ~~No~~	Yes ~~No~~	Yes ~~No~~	Yes ~~No~~
2	How would you rate the quality of this person's work	~~1 2~~ 3	~~1 2~~ 3	~~1 2~~ 3	~~1 2~~ 3
3	How would you rate the timeliness of the completion of the work?	~~1 2~~ 3	~~1 2~~ 3	~~1 2~~ 3	~~1 2~~ 3
4	How would you rate the accuracy of the work	~~1 2~~ 3	~~1 2~~ 3	~~1 2~~ 3	~~1 2~~ 3
5	Overall, how would you rank this group member's performance in the group?	~~1 2~~ 3	~~1 2~~ 3	~~1 2~~ 3	~~1 2~~ 3
6	Would you want to work with this person again? Explain why in the space below.	Yes ~~No~~	Yes ~~No~~	Yes ~~No~~	Yes ~~No~~
		All group members are very cooperative and working as a great team, very great to work with them!			

Name: Wan Muhammad Azib
Matric no:17207678

Course:**WIA2001 DATABASE** Semester: **Semester1** Session: **2021/2022**
Lecturer: **Dr.Fariza**
Assignment: ~~**Lab Practice**~~**/Group Project/**~~**Presentation**~~

Evaluator (Student's Name): **WAN MUHAMMAD AZIB**
Date: **17/1/2022**

Group Members:

	Matrix Number	Name
1	U2005398	Lee Kai Yang
2	U2005270	Liew Rui Zhi
3	U2005283	Chow Shang Shyan
4	U2005349	Chin Chin Fang

Directions: In the space below, honestly evaluate the work of other students in your group by answering **yes** or **no** and by using a scale from 1 to 3, **1 being poor, 2 being average, 3 being above average. Please circle your answer.**

		Group Member 1	Group Member 2	**Group Member 3**	**Group Member 4**
1	Did this group member complete his/her assigned tasks for the group	Yes ~~No~~	Yes ~~No~~	Yes ~~No~~	Yes ~~No~~
2	How would you rate the quality of this person's work	~~1 2~~ 3	~~1 2~~ 3	~~1 2~~ 3	~~1 2~~ 3
3	How would you rate the timeliness of the completion of the work?	~~1 2~~ 3	~~1 2~~ 3	~~1 2~~ 3	~~1 2~~ 3
4	How would you rate the accuracy of the work	~~1 2~~ 3	~~1 2~~ 3	~~1 2~~ 3	~~1 2~~ 3
5	Overall, how would you rank this group member's performance in the group?	~~1 2~~ 3	~~1 2~~ 3	~~1 2~~ 3	~~1 2~~ 3
6	Would you want to work with this person again? Explain why in the space below.	Yes ~~No~~	Yes ~~No~~	Yes ~~No~~	Yes ~~No~~
		All group members are very committed and hardworking to work with.			