\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**SCHOOL OF ENGINEERING AND TECHNOLOGY**

**COURSEWORK FOR**

BSC (HONS) IN COMPUTER SCIENCE

BSC (HONS) INFORMATION TECHNOLOGY

BACHELOR OF SOFTWARE ENGINEERING (HONS)

BSC (HONS) INFORMATION SYSTEMS (DATA ANALYTICS)

BSC (HONS) INFORMATION TECHNOLOGY (COMPUTER NETWORKING AND SECURITY)

**YEAR 1;** ACADEMIC SESSION APRIL 2023

**SEG1201:** DATABASE FUNDAMENTALS

**Part 1 Due date:** Week 9 Monday, 10am

**Part 2 Due date:** Week 11 Monday, 10am

**Parts 3&4 Due date:** Week 14 Monday, 10am

**Part 5** To be scheduled

**STUDENT NAMES: Tham Rou Yi, Lim Xiwei, Lam Hui Theng, Ng Jia Wen**

**STUDENT IDS: 21024161, 21045596, 21029137, 21057146** \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**INSTRUCTIONS**

* This final assessment contributes 50% to your final grade.
* This four-member group assignment is primarily for Course Learning Outcome 2 - Implement a database design group project using appropriate tools such as Oracle SQL.

**IMPORTANT**

The University requires students to adhere to submission deadlines for any form of assessment. Penalties are applied in relation to unauthorized late submission of work.

**Academic Honesty Acknowledgement**

“We **Tham Rou Yi, Lim Xiwei, Lam Hui Theng** and **Ng Jia Wen** verify that this paper contains entirely our own work.  We have not consulted with any outside person or materials other than what was specified (an interviewee, for example) in the assignment or the syllabus requirements.  Further, we have not copied or inadvertently copied ideas, sentences, or paragraphs from another student.  We realize the penalties *(refer to page 16, 5.5, Appendix 2, page 44 of the student handbook diploma and undergraduate programme)* for any kind of copying or collaboration on any assignment.”

A black background with a black square

Description automatically generated**A signature on a white background

Description automatically generated with medium confidence19/6/2023, ……............19/6/2023, …………………19/6/2023, ….…........19/6/2023,**



(Student signatures/Date)

Part 1: Enhance the chosen case scenario (10 marks)

**APPENDIX 1**

Scenario 1: Picky Pizza Palace

Picky Pizza Palace sells pizzas in the Kuala Lumpur, Shah Alam, and Petaling Jaya regions in Malaysia. This food business offers pizzas through two main methods: delivery and dining-in. To introduce their menu of delicious pizzas to the public, Picky Pizza Palace utilizes pamphlets. Additionally, they have a website showcasing their menu and specialties as part of their business strategies.

The management stores information about their menu, including item number, description, type of food, and price. All food and drink items are identified by their item number. Orders placed by customers are recorded in the database, capturing details such as order number, customer number, order date, order time, telephone operator, rider number (identified by their staff number), and outlet number. Each order is recognized by its unique order number.

For each order, the database maintains a list of all food and drink items ordered by customers. This list, known as Order Lines, includes information such as the order number, item number, and quantity of each item for every order placed by a customer. Each record in the Order Lines is identified by its order number and item number.

Picky Pizza Palace operates eight outlets situated in Damansara Utama, Bangsar Baru, Kelana Jaya, Shah Alam, Setia Alam, Klang, and Subang Jaya. Information about the food outlets, such as outlet number, telephone number, street, suburb, and postcode, is stored in the database. Each outlet is identified by its unique outlet number.

Information about the staff, including their staff number, outlet number, title, first name, last name, phone number, street, suburb, postcode, position, date of entry into the company, and any disciplinary actions taken, is stored in the database. All staff members are identified by their staff number.

As a well-known pizza parlour, Picky Pizza Palace stores all customer information in their database, including customer number, phone number, title, first name, last name, street, suburb, and postcode. Each customer is identified by their customer number. This strategy of storing customer information aims to enhance productivity and professionalism in the business environment, making customers feel valued and emphasizing convenience for them.

Considering the success and rapid growth of their pizza business, Picky Pizza Palace keeps track of their vehicles' condition and usability by storing relevant information in a database for easy management. Details about their vehicles, such as vehicle number, vehicle product, type, date of purchase, last checked date, mileage, and outlet number, are recorded. Each vehicle is identified by its unique vehicle number.

Picky Pizza Palace incorporates an online reviews and ratings system on their website to gather feedback from customers and improve their overall service quality. The feedback entity includes attributes such as review ID, customer number, order number, rating score, review text, and review date. Each review is identified by its unique review ID. After receiving their order, customers may leave a review and rate their experience through the website.

Picky Pizza Palace maintains a database of their ingredient suppliers to manage their supply chain effectively. The supplier entity includes attributes such as supplier ID, outlet number, ingredient ID, supplier name, contact number, and email. Each supplier is identified by a unique supplier ID.

A list of suppliers who supply a specific outlet, known as Supply Lines, is recorded. It includes information such as the outlet number, supplier ID, and delivery frequency. Each record in the Supply Lines is identified by its outlet number and supplier ID.

Picky Pizza Palace maintains an inventory of ingredients to ensure efficient supply management and prevent stockouts. The ingredient inventory entity includes attributes such as ingredient ID, ingredient name and quantity in stock. Each ingredient inventory is identified by its unique ingredient ID.

(198 words)

Business Rules:

1. An order can have multiple menu items.
2. A menu item can be found in many orders.
3. An order can have multiple order lines.
4. An order line must associate with only one order.
5. Each order line is associated with only one menu item.
6. A menu item may have many order lines.
7. Each customer can place many orders.
8. Each order is placed by one and only one customer.
9. Each customer may give many reviews.
10. Each review can only be associated with one and only one customer.
11. Each order may or may not be delivered by only one rider.
12. Each rider may deliver many orders.
13. Each order is placed from a specific outlet.
14. An outlet can place many orders.
15. Each outlet has one or more vehicles.
16. Each vehicle can only be assigned to one outlet.
17. Each outlet employs multiple staff members.
18. Each staff member can work at only one outlet.
19. Each supplier is associated with one or more outlets.
20. Each outlet can be supplied by one or more suppliers.
21. A supplier can have multiple supply lines.
22. A supply line must associate with only one supplier.
23. Each supply line is associated with only one outlet.
24. An outlet may have many supply lines.
25. Each supplier only supplies one ingredient type (e.g., meat, flour, vegetables).
26. Each ingredient can be supplied by one or more suppliers.
27. Every menu item must have a price that is only a positive integer.
28. Every menu item must have a description.
29. Every menu item must be labelled with one of the food categories indicated by the company.
30. The quantity of each item in the order lines must be a positive integer.
31. Every order must have a date, time, and order method.
32. Every order may have a telephone operator.
33. Each customer must have complete information such as name, address, and phone number recorded. However, for dine-in orders, the address field can be left empty.
34. Each review must have a rating score, ranging from 1 to 5 stars, and a review date.
35. Each review may have a review text.
36. Each outlet must have its own telephone number and address.
37. Each vehicle must have vehicle number, vehicle product, type, date of purchase, and last checked date recorded.
38. The recorded mileage of the vehicle cannot be a negative integer.
39. Each staff must have complete information such as name, address, date of entry and phone number recorded.
40. Any disciplinary actions taken against staff members should be documented.
41. Each supplier must have their complete information such as name, contact number, and email recorded.
42. Delivery frequency in supply lines must be specified (e.g., daily, weekly, monthly).
43. Each ingredient's stock quantity must be a positive integer or zero.

Part 2: Design a database (30 marks)

**Please include ERD in Crow foot or UML notation and RDM in 3NF here. If you decide to use/choose an ID as a primary key, please also include an alternate key.**

**Please DO NOT include the normalisation process.**

**Note: Your submission for part 2 will only be graded during the presentation. Hence, please proceed with the subsequent parts in order to meet the final deadline.**

**RELATIONAL DATA MODEL (RDM)**

MENU(itemID, description, food\_type, price)

ORDERS (orderID , customerID, order\_date, order\_time, order\_method, telephone\_operator, riderID, outletID)

*FOREIGN KEY customerID REFRENCES TABLE from CUSTOMER*

*FOREIGN KEY outletID REFRENCES TABLE from OUTLETS*

*FOREIGN KEY riderID REFRENCES TABLE from STAFF*

*FOREIGN KEY telephone\_operator REFRENCES TABLE from STAFF*

*ALTERNATE KEY ( customerID, order\_date, order\_time)*

ORDERLINES (orderID,itemID , quantity)

*FOREIGN KEY orderID REFRENCES TABLE from ORDERS*

*FOREIGN KEY itemID REFRENCES TABLE from MENU*

OUTLETS (outletID, telephone\_num, street, suburb, postcode)

*ALTERNATE KEY (telephone\_num)*

STAFF (staffID, outletID, title, first\_name, last\_name, phone\_num, street, suburb, postcode, position, date\_of\_entry, disciplinary\_action)

*FOREIGN KEY outletID REFERENCES TABLE from OUTLETS*

*ALTERNATE KEY (phone\_num)*

CUSTOMER (customerID, phone\_num, title, first\_name, last\_name, street, suburb, postcode)

*ALTERNATE KEY (phone\_num)*

VEHICLES (vehicleID, brand, vehicle\_type, date\_of\_purchase, last\_checked \_date, mileage, outletID)  
*FOREIGN KEY outletID REFERENCES TABLE from OUTLETS*

*ALTERNATE KEY (vehicleID, outletID)*

FEEDBACK (reviewID, customerID, orderID, rating\_score, review\_text, review\_date)

*FOREIGN KEY customerID REFERENCES TABLE from CUSTOMER*

*FOREIGN KEY orderID REFERENCES TABLE from ORDERS*

*ALTERNATE KEY (customerID, orderID)*

SUPPLIERS (supplierID, ingredientID, supplier\_name, contact\_num, email)

*FOREIGN KEY ingredientID REFERENCES TABLE from INGREDIENTS*

*ALTERNATE KEY (supplier\_name)*

*ALTERNATE KEY (email)*

*ALTERNATE KEY (contact\_num)*

SUPPLYLINES (outletID, supplierID, delivery\_frequency)

*FOREIGN KEY outletID REFERENCES TABLE from OUTLETS*

*FOREIGN KEY supplierID REFRENCES TABLE from SUPPLIERS*

INGREDIENTS (ingredientID, ingredient\_type, quantity\_in\_stock)

*ALTERNATE KEY (ingredient\_type)*

If you have an update to the ERD and RDM after submission Part 2 submission, please use the table to explain the changes done and reasons. The red prints are examples only and they need to be removed.

|  |  |  |  |
| --- | --- | --- | --- |
| No | Changes | Reason for the change/update | Changes in ERD/RDM |
| 1 | Before:  SUPPLIERS table consists of the field, outletID  After:  outletID is removed from the SUPPLIERS table | This field is redundant as it exists in the SUPPLYLINES table, and play no crucial role in SUPPLIER table | RDM |
| 2 | Before:  STAFF table do not have the field, title  After:  title is added to the STAFF table | This field has been left out | RDM |
| 3 | Before:  ORDER table does not consist of the field order\_method  After:  Order\_method is added to the ORDER table | To accommodate the business rule where telephone\_operator may be null, depending on whether the order method is delivery or dine-in | RDM |
| 4 | Before:  ‘*FOREIGN KEY ingredientID REFERENCES TABLE from OUTLETS’*  This statement exists in ingredients table  After:  The statement is removed, where ingredientID is not a foreign key for ingredients table | A typo error that we overlooked | RDM |

**ENTITY RELATIONSHIP DIAGRAM (ERD)**

***A diagram of a computer

Description automatically generated***

Part 3: Implement a database (20 marks)

1. Use the following format.

|  |  |  |  |
| --- | --- | --- | --- |
| Table/attributes | description | Data Type | Justification |
| **MENU** | Table name | **-** |  |
| itemID | Unique code for each menu item | CHAR (6) | Fixed number of characters, where first two characters is the food type.  (e.g., PZ1001, PT2001, SD3001, DS4001, DR5001) |
| description | Menu item description | VARCHAR2 (200) | A variable-length field |
| food\_type | Type/category of food | VARCHAR2 (50) | A variable-length field  (e.g., Pizzas, Pasta, Sides, Desserts, Drinks) |
| price | Menu item price | NUMBER (5, 2) | A numeric data type with 5 digits precision and 2 decimal places |
| **ORDERS** | Table name | **-** |  |
| orderID | Unique code for each order | CHAR (12) | Fixed number of characters  (e.g., DE0123456ABC  DI6543210DEF) |
| customerID | Unique code for each customer | CHAR (10) | Fixed number of characters (e.g., C123456789) |
| order\_date | Date of order | DATE | Use the date format 'DD-Mon-YYYY' |
| order\_time | Time of order | TIMESTAMP (0) | Use the timestamp format ‘HH24:MI:SS’ without fractional seconds |
| order\_method | Method of ordering | VARCHAR2 (8) | Variable length field for two ordering methods  (e.g., delivery, dine-in) |
| telephone\_operator | Staff taking delivery orders over the phone | CHAR (7) | Fixed number of 7 characters (e.g., CC05297) |
| riderID | Unique code for each staff | CHAR (7) | Fixed number of 7 characters (e.g., DD09286) |
| outletID | Unique code for each outlet | CHAR (5) | Fixed number of characters. Where first 2 characters ‘OL’ indicates outlet  (e.g., OL001, OL002) |
| **ORDERLINES** | Table name | **-** |  |
| orderID | Unique code for each order | CHAR (12) | Fixed number of characters  (e.g., DE0123456ABC  DI6543210DEF) |
| itemID | Unique code for each menu item | CHAR (6) | Fixed number of characters, where first two characters is the food type.  (e.g., PZ1001, PT2001, SD3001, DS4001, DR5001) |
| Quantity | Quantity of each item ordered | NUMBER (2) | A numeric data type with 2 digits precision |
| **OUTLETS** | Table name | **-** |  |
| outletID | Unique code of each outlet | CHAR (5) | Fixed number of characters. Where first 2 characters ‘OL’ indicates outlet  (e.g., OL001, OL002) |
| telephone\_num | Telephone number of each outlet | VARCHAR2(12) | Variable length field for different formats of phone numbers, including landline and mobile for Malaysia.  (e.g., 03-5192 0607,  010-234 5678,  012-1234 5678) |
| street | Street where each outlet is located | VARCHAR2(50) | Variable length field |
| suburb | Suburb where each outlet is located | VARCHAR2(30) | Variable length field |
| postcode | Postcode of each outlet | CHAR (5) | Fixed number of 5 digits for Malaysia’s postcode number |
| **STAFF** | Table name | **-** |  |
| staffID | Staff identification number | CHAR (7) | Fixed number of characters where the first 2 character determines the staff position (e.g., SM12345 is a staff ID for a store manager) |
| outletID | Unique code for the outlet where the staff works | CHAR (5) | Fixed number of characters. Where first 2 characters ‘OL’ indicates outlet  (e.g., OL001, OL002) |
| title | Title of the staff | VARCHAR2(20) | A variable-length field (e.g., Mr, Mrs, and Miss) |
| first\_name | First name of the staff | VARCHAR2(30) | A variable-length field |
| last\_name | Last name of the staff | VARCHAR2(30) | A variable-length field |
| phone\_num | Phone number of the staff | VARCHAR2(12) | Variable length field for different formats of phone numbers, including landline and mobile for Malaysia.  (e.g., 03-5192 0607,  010-234 5678,  012-1234 5678) |
| street | Street address of the staff | VARCHAR2(100) | A variable-length field |
| suburb | Suburb/city of the staff’s address | VARCHAR2(100) | A variable-length field |
| postcode | Postal code of the staff’s address | NUMBER (5,0) | Fixed number of 5 digits |
| position | Position of the staff | VARCHAR (40) | A variable-length field |
| date\_of\_entry | The date the staff join Picky Pizza Palace | DATE | Use the date format 'DD-Mon-YYYY' |
| disciplinary\_action | Disciplinary action details | VARCHAR2(400) | A variable-length field |
| **CUSTOMER** | Table name | **-** |  |
| customerID | Unique code for each customer | CHAR (10) | Fixed number of characters (e.g., C123456789) |
| phone\_num | Phone number of the customer | VARCHAR2(12) | A variable-length field |
| title | The title of the customer | VARCHAR2(10) | A variable-length field (e.g., Mr, Mrs, and Miss) |
| first\_name | The first name of the customer | VARCHAR2(50) | A variable-length field |
| last\_name | The last name of the customer | VARCHAR2(50) | A variable-length field |
| street | The street of the customer address | VARCHAR2(100) | A variable-length field |
| suburb | The suburb of the customer address | VARCHAR2(50) | A variable-length field |
| postcode | The postcode of customer address | CHAR (5) | Fixed number of 5 digits |
| **VEHICLE** | Table name | **-** |  |
| vehicleID | Unique code for each vehicle | VARCHAR2 (20) | A variable-length field (e.g., ABC1234, DEF456) |
| brand | The brand of the outlet or delivery vehicle | VARCHAR2(30) | A variable-length field (e.g., Motorbike, car) |
| vehicle\_type | The different type of the vehicle | VARCHAR2(10) | A variable-length field |
| date\_of\_purchase | The date of the purchase vehicle | DATE | Use the date format 'DD-Mon-YYYY' |
| last\_checked\_date | The last checked date for the outlet or delivery vehicle | DATE | Use the date format 'DD-Mon-YYYY' |
| mileage | The mileage of the vehicle | FLOAT (3) | A floating-point number, the p value of 3 indicates that it is of data type float, which holds approximate values such as mileage |
| outletID | Unique code of each outlet | CHAR (5) | Fixed number of characters. Where first 2 characters ‘OL’ indicates outlet  (e.g., OL001, OL002) |
| **FEEDBACK** | Table name | **-** |  |
| reviewID | The identification code for the review | CHAR (10) | Fixed number of characters (e.g., QEW2345678) |
| customerID | Unique code for each customer | CHAR (10) | Fixed number of characters (e.g., C123456789) |
| orderID | Unique code for each order | CHAR (12) | Fixed number of characters  (e.g., DE0123456ABC  DI6543210DEF) |
| rating\_score | The rating score of each customer | NUMBER (5) | A numeric data type with 5 digits precision |
| review\_text | The review text for each customer | VARCHAR2(100) | A variable-length field |
| review\_date | The review date of each customer | DATE | Use the date format 'DD-Mon-YYYY' |
| **SUPPLIERS** |  |  |  |
| supplierID | The identification code for each supplier | CHAR (5) | Fixed number of characters. Where first 2 characters SP indicates supplier  (e.g., SP001, SP002) |
| ingredientID | The identification code for each ingredient | CHAR (5) | Fixed number of characters. Where first 2 characters indicates the type of ingredient  (e.g., FL100 is one of the ingredient codes for flour) |
| supplier\_name | The name of each supplier | VARCHAR (50) | Variable length field. |
| contact\_num | The contact number of each supplier | VARCHAR (12) | Variable length field for different formats of phone numbers, including landline and mobile for Malaysia.  (e.g., 03-5192 0607,  010-234 5678,  012-1234 5678) |
| email | The email of the supplier | VARCHAR (50) | Variable length field. |
| **SUPPLYLINES** | Table name | **-** |  |
| outletID | Unique code for each outlet | CHAR (5) | Fixed number of characters. Where first 2 characters ‘OL’ indicates outlet  (e.g., OL001, OL002) |
| supplierID | Unique code for each supplier | CHAR (5) | Fixed number of characters. Where first 2 characters SP indicates supplier  (e.g., SP001, SP002) |
| delivery\_frequency | Frequency of delivery made | VARCHAR2(20) | Variable length field  (e.g., daily, weekly, monthly) |
| **INGREDIENTS** | Table name | **-** |  |
| ingredientID | Unique code for each ingredient | CHAR (5) | Fixed number of characters. Where first 2 characters indicates the type of ingredient  (e.g., FL100 is one of the ingredient codes for flour) |
| ingredient\_type | Type of ingredients | VARCHAR2(40) | Fixed number of characters. (e.g., Flour, Sauce, Cheese, Meats, Vegetables, Herbs and Spices, Seafoods and Beverages) |
| quantity\_in\_stock | Quantity of each ingredient in stock | NUMBER (5) | A numeric data type with 5 digits precision |

1. List *DBscript* here (please use 9-point size for the listing)

ALTER SESSION SET NLS\_DATE\_FORMAT='DD-Mon-YYYY';

DROP TABLE outlets CASCADE CONSTRAINTS;

DROP TABLE staff CASCADE CONSTRAINTS;

DROP TABLE menu CASCADE CONSTRAINTS;

DROP TABLE orders CASCADE CONSTRAINTS;

DROP TABLE orderlines CASCADE CONSTRAINTS;

DROP TABLE customer CASCADE CONSTRAINTS;

DROP TABLE vehicles CASCADE CONSTRAINTS;

DROP TABLE feedback CASCADE CONSTRAINTS;

DROP TABLE suppliers CASCADE CONSTRAINTS;

DROP TABLE supplylines CASCADE CONSTRAINTS;

DROP TABLE ingredients CASCADE CONSTRAINTS;

DROP VIEW staff\_customer\_view;

DROP VIEW monthly\_revenue\_view;

CREATE TABLE outlets (

outletID CHAR(5) PRIMARY KEY,

telephone\_num VARCHAR2(12) NOT NULL

CONSTRAINT chk\_outlet\_phone\_num\_digits

CHECK (REGEXP\_LIKE(telephone\_num, '^[0-9-]+$')),

street VARCHAR(50) NOT NULL,

suburb VARCHAR2(30) NOT NULL,

postcode CHAR(5) NOT NULL,

UNIQUE (telephone\_num)

);

CREATE TABLE staff(

staffID CHAR(7) PRIMARY KEY,

outletID CHAR(5) NOT NULL,

title VARCHAR2(20) NOT NULL,

first\_name VARCHAR2(30) NOT NULL,

last\_name VARCHAR2(30) NOT NULL,

phone\_num VARCHAR2(12) NOT NULL,

CONSTRAINT chk\_staff\_phone\_num\_digits

CHECK (REGEXP\_LIKE(phone\_num, '^[0-9-]+$')),

street VARCHAR2(100) NOT NULL,

suburb VARCHAR2(100) NOT NULL,

postcode NUMBER(5,0) NOT NULL,

position VARCHAR2(40) NOT NULL,

date\_of\_entry DATE NOT NULL,

disciplinary\_action VARCHAR2(400),

FOREIGN KEY (outletID) REFERENCES outlets(outletID),

UNIQUE(first\_name, last\_name,phone\_num)

);

CREATE TABLE menu (

itemID CHAR(6) PRIMARY KEY,

description VARCHAR2(200) NOT NULL,

food\_type VARCHAR2(50) NOT NULL,

price NUMBER(5,2) NOT NULL

CONSTRAINT chk\_price

CHECK (price > 0)

);

CREATE TABLE customer (

customerID char(10) PRIMARY KEY,

phone\_num VARCHAR2(12) NOT NULL

CONSTRAINT chk\_phone\_num\_digits

CHECK (REGEXP\_LIKE(phone\_num, '^[0-9-]+$')),

title VARCHAR2(10) NOT NULL,

first\_name VARCHAR2(50) NOT NULL,

last\_name VARCHAR2(50) NOT NULL,

street VARCHAR2(100),

suburb VARCHAR2(50),

postcode char(5),

UNIQUE (phone\_num)

);

CREATE TABLE orders (

orderID char(12) PRIMARY KEY,

customerID char(10) NOT NULL,

order\_date DATE NOT NULL,

order\_time TIMESTAMP(0) NOT NULL,

order\_method VARCHAR2(8) NOT NULL

CONSTRAINT chk\_order\_method

CHECK (order\_method IN ('dine-in', 'delivery')),

telephone\_operator char(7),

riderID char(7),

outletID char(5) NOT NULL,

UNIQUE (customerID, order\_date, order\_time),

FOREIGN KEY (customerID) REFERENCES customer(customerID),

FOREIGN KEY (telephone\_operator) REFERENCES staff(staffID),

FOREIGN KEY (riderID) REFERENCES staff(staffID),

FOREIGN KEY (outletID) REFERENCES outlets(outletID)

);

CREATE TABLE orderlines (

orderID CHAR(12),

itemID CHAR(6),

quantity NUMBER(2) NOT NULL

CONSTRAINT chk\_quantity

CHECK (quantity > 0),

PRIMARY KEY(orderID, itemID),

FOREIGN KEY (orderID) REFERENCES orders(orderID),

FOREIGN KEY (itemID) REFERENCES menu(itemID)

）;

CREATE TABLE vehicles (

vehicleID VARCHAR2(20) PRIMARY KEY,

brand VARCHAR2(30) NOT NULL,

vehicle\_type VARCHAR2(10) NOT NULL,

date\_of\_purchase DATE NOT NULL,

last\_checked\_date DATE NOT NULL,

mileage FLOAT(3) NOT NULL

CONSTRAINT chk\_mileage

CHECK (mileage >=0),

outletID CHAR(5),

FOREIGN KEY (outletID) REFERENCES outlets(outletID),

UNIQUE (vehicleID, outletID)

);

CREATE TABLE feedback (

reviewID CHAR(10) PRIMARY KEY,

customerID CHAR(10) NOT NULL,

orderID CHAR(12) NOT NULL,

review\_score NUMBER(5) NOT NULL

CONSTRAINT chk\_score

CHECK (review\_score >= 1 AND review\_score <= 5),

review\_text VARCHAR2(100),

review\_date DATE NOT NULL，

FOREIGN KEY (customerID) REFERENCES customer(customerID),

FOREIGN KEY (orderID) REFERENCES orders(orderID),

UNIQUE (customerID, orderID)

);

CREATE TABLE ingredients (

ingredientID CHAR(5) PRIMARY KEY,

ingredient\_type VARCHAR2(40) NOT NULL,

quantity\_in\_stock NUMBER(5) NOT NULL,

UNIQUE (ingredient\_type)

);

CREATE TABLE suppliers (

supplierID CHAR(5) PRIMARY KEY,

ingredientID CHAR(5) NOT NULL,

supplier\_name VARCHAR(50) NOT NULL,

contact\_num VARCHAR(12) NOT NULL

CONSTRAINT chk\_cus\_phone\_num\_digits

CHECK (REGEXP\_LIKE(contact\_num, '^[0-9-]+$')),

email VARCHAR(50) NOT NULL

CONSTRAINT chk\_email

CHECK (email LIKE '%\_@\_\_%.%\_%'),

FOREIGN KEY (ingredientID) REFERENCES ingredients(ingredientID)，

UNIQUE (supplier\_name),

UNIQUE (email),

UNIQUE (contact\_num)

);

CREATE TABLE supplylines (

outletID CHAR(5) NOT NULL,

supplierID CHAR(5) NOT NULL,

delivery\_frequency VARCHAR2(20) NOT NULL,

CONSTRAINT chk\_delivery\_frequency

CHECK (delivery\_frequency IN ('daily', 'weekly', 'monthly')),

PRIMARY KEY (outletID, supplierID),

FOREIGN KEY (outletID) REFERENCES outlets(outletID),

FOREIGN KEY (supplierID) REFERENCES suppliers(supplierID)

);

INSERT INTO outlets VALUES ('OL001', '016-98765432', 'Jalan SS 21/1', 'Damansara Utama', '47400');

INSERT INTO outlets VALUES ('OL002', '011-65437890', 'Jalan Telawi 5', 'Bangsar Baru', '59100');

INSERT INTO outlets VALUES ('OL003', '012-4563742', 'Jalan SS 6/1', 'Kelana Jaya', '47301');

INSERT INTO outlets VALUES ('OL004', '03-51920607', 'Jalan Setia Prima E U13/E', 'Shah Alam', '47310');

INSERT INTO outlets VALUES ('OL005', '010-2345892', 'Jalan Setia Nusantara U13/N', 'Setia Alam', '40170');

INSERT INTO outlets VALUES ('OL006', '012-12345678', 'Jalan Kota Raja E 27/E', 'Klang', '41100');

INSERT INTO outlets VALUES ('OL007', '010-9876543', 'Jalan USJ 10/1G', 'Subang Jaya', '47600');

INSERT INTO staff VALUES ('SM12345', 'OL005', 'Ms.', 'Mei Ling', 'Low', '012-4567894', 'Jalan Apartment Puncak Baiduri', 'Cheras', 43200, 'Store Manager', '18-Jan-2019',NULL);

INSERT INTO staff VALUES ('DD09286', 'OL002', 'Mr.', 'Jaya', 'Changuuran', '013-5647283', 'Jalan Klang Lama', 'Old Klang', 58800, 'Delivery Driver', '09-Feb-2022', 'Suspended one week');

INSERT INTO staff VALUES ('SL23857', 'OL001', 'Mrs.', 'Shinna', 'Chopra', '012-5765784', 'Jalan Akik 7/3', 'Shah Alam', 40000, 'Shift Leader', '01-Apr-2018', NULL);

INSERT INTO staff VALUES ('SC27384', 'OL003', 'Mr.', 'Ali', 'Bin Muhammad Abu', '014-4859372', 'Jalan Alur 8/27', 'Shah Alam', 40000,'Store Cleaner', '29-Mar-2020', 'Fine');

INSERT INTO staff VALUES ('TM28475', 'OL005', 'Mr.', 'Jamal', 'Bin Abu Bakar', '016-4839528', 'Jalan SS 8/39', 'Kelana Jaya', 47300, 'Team Member', '30-May-2021', 'Suspended one week');

INSERT INTO staff VALUES ('TM36482', 'OL004', 'Ms.', 'Shelly', 'Lim', '019-2748576', 'Jalan Klang', 'Klang', 41100, 'Team Member','04-Jun-2017', NULL);

INSERT INTO staff VALUES ('SM36475', 'OL002', 'Dr.', 'Vesper', 'Noir', '017-2465738', 'Jalan BU6/12', 'Bandar Utama', 47800 ,'Store Manager','08-Jul-2019', NULL);

INSERT INTO staff VALUES ('SL27384', 'OL003', 'Mrs.', 'Lei Chang', 'Tan', '014-3865759', 'Jalan Bangsa Baru', 'Bangsa Baru', 59100, 'Shift Leader', '04-Aug-2022',NULL);

INSERT INTO staff VALUES ('SC37109', 'OL001', 'Mrs.', 'Kelly', 'Kang', '016-4487395', 'Jalan BU3/4', 'Bandar Utama', 47800, 'Store Cleaner', '08-Mar-2017', 'Fine');

INSERT INTO staff VALUES ('CC64270', 'OL005', 'Mr.', 'Eddy', 'Chan', '012-3456789', '30, Jalan Setia Indah U13/12q', 'Setia Alam', 40170, 'Counter Cashier', '12-Jan-2023', NULL);

INSERT INTO staff VALUES ('CC34058', 'OL004', 'Ms.', 'Nurul', 'Mustafa', '011-23456789', 'Jalan Setia Alam Street', 'Setia Alam', 40200, 'Counter Cashier', '28-Feb-2022', 'Suspended for one week');

INSERT INTO staff VALUES ('CC23948', 'OL005', 'Ms.', 'Siti', 'Jaffar', '019-87654321', 'Jalan Shah Alam Avenue', 'Shah Alam', 40450, 'Counter Cashier', '09-Mar-2022', NULL);

INSERT INTO staff VALUES ('CC92683', 'OL002', 'Mr.', 'Mohd', 'Azman', '013-65789012', '87, Jalan Terasek 8', 'Bangsar', 59100, 'Counter Cashier', '16-Apr-2022', NULL);

INSERT INTO staff VALUES ('CC19238', 'OL005', 'Ms.', 'Farah', 'Yusri', '018-43210987', 'Jalan Eco park', 'Setia Alam', 40200, 'Counter Cashier', '05-Jan-2022', NULL);

INSERT INTO staff VALUES ('CC05297', 'OL001', 'Mr.', 'Amir', 'Iskandar', '014-56789012', '17, Jalan SS 20/22', 'Damansara Kim', 47400, 'Counter Cashier', '21-Jun-2017', NULL);

INSERT INTO staff VALUES ('CC14384', 'OL007', 'Ms.', 'Xin Ling', 'Chia', '017-12345678', '17, Jalan USJ 3/4j, Usj 3', 'Subang Jaya', 47600, 'Counter Cashier', '14-Jul-2019', 'fine');

INSERT INTO staff VALUES ('CC02938', 'OL006', 'Ms.', 'Azizah', 'Hamid', '016-78901234', 'Jalan Bahagia', 'Klang', 41100, 'Counter Cashier', '30-Aug-2021', NULL);

INSERT INTO staff VALUES ('DD75103', 'OL005', 'Ms.', 'Nor', 'Ibrahim', '015-67890123', 'Jalan Seri Harmoni', 'Setia Alam', 40200, 'Delivery Driver', '11-Feb-2019', NULL);

INSERT INTO staff VALUES ('DD32104', 'OL004', 'Mr.', 'Hafiz', 'Abdul Razak', '012-78901234', 'Jalan Mutiara', 'Shah Alam', 40450, 'Delivery Driver', '24-Mar-2017', NULL);

INSERT INTO staff VALUES ('DD94275', 'OL005', 'Ms.', 'Zara', 'Ramlan', '012-34567890', 'Jalan Damai', 'Setia Alam', 40200, 'Delivery Driver', '27-Aug-2019', NULL);

INSERT INTO staff VALUES ('DD79462', 'OL001', 'Mr.', 'Amirul', 'Jamaluddin', '013-90123456', 'Jalan Damai Utama', 'Damansara Utama', 47800, 'Delivery Driver', '28-Feb-2018', 'Suspended for one week');

INSERT INTO staff VALUES ('DD12038', 'OL007', 'Mr.', 'Syed', 'Ismail Hashim', '019-23456789', 'Jalan Indah Makmur', 'Subang Jaya', 47600, 'Delivery Driver', '04-Apr-2017', NULL);

INSERT INTO staff VALUES ('DD29382', 'OL006', 'Ms.', 'Fatimah', 'Saidin', '019-78901234', 'Jalan Cendana Murni', 'Klang', 41100, 'Delivery Driver', '07-Apr-2021', 'Suspended for one week');

INSERT INTO staff VALUES ('SM12444', 'OL001', 'Ms.', 'Gigi', 'Tan', '012-4567894', '56, Jalan BU 2/11', 'Damansara Utama', 47900, 'Store Manager','30-Apr-2017', NULL);

INSERT INTO staff VALUES ('SM12897', 'OL003', 'Mr.', 'Shawn', 'Li', '013-5647283', '68, Jalan Kelana Jaya 3/11','Kelana Jaya', 47301,'Store Manager', '15-Feb-2018', NULL);

INSERT INTO staff VALUES ('SM87903', 'OL004', 'Ms.', 'Lily', 'Nim', '014-4859372', '57, Jalan Shah 9/10', 'Shah Alam', 40450, 'Store Manager', '11-Sep-2018', NULL);

INSERT INTO staff VALUES ('SM12198', 'OL006', 'Mrs.', 'Mandy', 'Kok', '016-4839528', '69, Jalan Klang 5/67', 'Klang', 41100, 'Store Manager','18-May-2020', NULL);

INSERT INTO staff VALUES ('SM38576', 'OL007', 'Ms.', 'Paula', 'Si', '019-2748576', '76, Jalan Subang Jaya 9/11', 'Subang Jaya', 47600, 'Store Manager', '13-Jun-2017', NULL);

INSERT INTO staff VALUES ('CC09374', 'OL004', 'Mr.', 'Faisal', 'Mahmod', '010-89012345', 'Jalan Murni', 'Shah Alam', 40450, 'Counter Cashier', '23-Jan-2023', 'fine');

INSERT INTO staff VALUES ('DD02938', 'OL003', 'Ms.', 'Aina', 'Abdullah Sani', '023-56789012', 'Jalan Mutiara Permai', 'Kelana Jaya', 47301, 'Delivery Driver', '11-Feb-2022', 'fine');

INSERT INTO staff VALUES ('DD83239', 'OL004', 'Mrs.', 'Kai Ming', 'Lim', '026-90123456', 'Jalan Mewah Indah', 'Shah Alam', 40200, 'Delivery Driver', '29-May-2023', NULL);

INSERT INTO staff VALUES ('CC29472', 'OL003', 'Ms.', 'Vasuda', 'Lal', '012-2394234', '36, Jalan SS 5d/8, Ss 5', 'Kelana Jaya', 47301, 'Counter Cashier', '12-Sep-2022', 'Verbal warning');

INSERT INTO staff VALUES ('CC12934', 'OL007', 'Mr.', 'Shan Yuan', 'Chew', '014-2394787', '45, Jalan USJ 11/4m, Usj 11', 'Subang Jaya', 47620, 'Counter Cashier', '26-Apr-2022', 'Verbal warning');

INSERT INTO menu VALUES ('PZ1001', 'Blazing Seafood with spicy sweet sour sauce, tuna, crabsticks, pineapples, capsicums, onions, mozzarella cheese.', 'Pizza', 30.00);

INSERT INTO menu VALUES ('PT2001', 'Spaghetti with Tomato Bolognaise sauce, topped with Chicken Meatballs.', 'Pasta', 20.00);

INSERT INTO menu VALUES ('SD3001', 'Huts poppers sausuage', 'Sides' , 5.50);

INSERT INTO menu VALUES ('DR4001', 'Red Bull', 'Drinks', 3.40);

INSERT INTO menu VALUES ('DS5001', 'Cheese cake with chocalte ice cream', 'Desserts', 8.90);

INSERT INTO menu VALUES ('DS6072', 'Brownie Ala Mode', 'Desserts', 9.40);

INSERT INTO menu VALUES ('DR3450', 'Coke','Drinks', 5.30);

INSERT INTO menu VALUES ('PZ2301', 'Island Suprime with thousand island sauce, crabsticks, tuna, pineapples, onions, mozzarella cheese.', 'Pizza', 28.00);

INSERT INTO menu VALUES ('PT1001', 'Spaghetti with Creamy Carbonara sauce, chicken rolls, mushrooms and herbs.', 'Pasta', 21.00);

INSERT INTO menu VALUES ('PT3101', 'Mushroom & Garlic Spaghetti', 'Pasta', 21.00);

INSERT INTO menu VALUES ('SD3102', 'Crazy Chicken Crunchies with bbq sauce', 'Sides', 25.00);

INSERT INTO menu VALUES ('PT4101', 'Penne pasta tossed in a savory marinara sauce with sautéed mushrooms.', 'Pasta', 19.90);

INSERT INTO menu VALUES ('PT3304', 'A crispy thin crust topped with BBQ chicken and caramelized onions.', 'Pizza', 23.60);

INSERT INTO menu VALUES ('DS2467', 'Rich chocolate lava cake with a molten center, served with a scoop of vanilla ice cream.','Desserts', 15.40);

INSERT INTO menu VALUES ('DR2365', '100 plus', 'Drinks', 5.50);

INSERT INTO menu VALUES ('DR4467', 'A refreshing blend of freshly squeezed oranges and pineapples.', 'Drinks',6.70);

INSERT INTO menu VALUES ('PZ4301', 'A classic Margherita pizza with fresh basil and tomatoes on a soft, chewy crust.', 'Pizza', 25.70);

INSERT INTO menu VALUES ('SD5164', 'Golden-brown onion rings with a zesty dipping sauce.', 'Sides', 6.60);

INSERT INTO menu VALUES（'DS9499', 'A delectable slice of creamy cheesecake topped with mixed berries.', 'Desserts', 10.10);

INSERT INTO menu VALUES ('PZ9067', 'A delightful vegetarian pizza with roasted bell peppers, black olives, and feta cheese.', 'Pizza', 20.90);

INSERT INTO customer VALUES ('C696284290', '012-3456789', 'Mr.', 'John', 'Doe', '17, Jalan USJ 3/4j, Usj 3', 'Subang Jaya', '47600');

INSERT INTO customer VALUES ('C984027513', '019-1284248', 'Mrs.', 'Xiu Ying', 'Lee', '22, Jalan Setia Damai U13/15f', 'Setia Alam', '40170');

INSERT INTO customer VALUES ('C784039281', '011-23909405', 'Mr.', 'Mohammed Ashrul', 'bin Hamizuddin', NULL, NULL, NULL);

INSERT INTO customer VALUES ('C532938109', '012-2866789', 'Mr.', 'John', 'Ngu', '65, Jalan Pualam 7/32, Seksyen 7', 'Shah Alam', '40000');

INSERT INTO customer VALUES ('C232934893', '012-5453196', 'Mrs.', 'Khairun Fairos', 'binti Yuszelan', '33, Lorong Sungai 8, Taman Kota Jaya', 'Klang', '41000');

INSERT INTO customer VALUES ('C239457108', '011-19718250', 'Mr.', 'David', 'Wong', '90, Jalan Terasek 1', 'Bangsar', '59100');

INSERT INTO customer VALUES ('C128302094', '017-6391219', 'Mrs.', 'Nithya', 'a/l Santhara', NULL, NULL, NULL);

INSERT INTO customer VALUES ('C930485752', '016-2136823', 'Mr.', 'Daniel', 'Tan', '20, Jalan Setia Indah U13/11n', 'Setia Alam', '40170');

INSERT INTO customer VALUES ('C124978532', '019-3020070', 'Mr.', 'Vijaya Krishnan', 'a/l Ravichandran', NULL, NULL, NULL);

INSERT INTO customer VALUES ('C123456789', '013-2446968', 'Mr.', 'Wei Chen', 'Lim', '37, JLN SS 21/24', 'Damansara Utama', '47400');

INSERT INTO customer VALUES ('C758203419', '011-51041592', 'Ms.', 'Nur Liyana Nashriq', 'binti Zulfadhli', NULL, NULL, NULL);

INSERT INTO customer VALUES ('C923649123', '011-11061791', 'Mrs.', 'Noor Raihah Hadzis', 'binti Sharum', '47, Jalan USJ 11/4l, Usj 11', 'Subang Jaya', '47620');

INSERT INTO customer VALUES ('C023850342', '014-3456789', 'Ms.', 'Xin Lin', 'Lan', NULL, NULL, NULL);

INSERT INTO customer VALUES ('C203770683', '017-8063737', 'Ms.', 'Priya', 'a/l Sothinathan', NULL, NULL, NULL);

INSERT INTO customer VALUES ('C370598463', '014-6199227', 'Mr.', 'Benjamin', 'Chou', '39, Jalan Tambatan 19/16, Seksyen 19', 'Shah Alam', '40300');

INSERT INTO customer VALUES ('C436285973', '013-2957384', 'Mr.', 'Alexander', 'Wong', 'Blok 4, Flat PKNS, Seksyen 16', 'Shah Alam', '40200');

INSERT INTO customer VALUES ('C239179283', '017-2047534', 'Ms.', 'Rashaa', 'binti Baheej', '31, Jalan USJ 13/1a, Usj 13', 'Subang Jaya', '47630');

INSERT INTO customer VALUES ('C348239940', '016-2394758', 'Mr.', 'Qaaid', 'bin Naaif', '15, Jalan Batu Mandi 26/29, Taman Bukit Saga', 'Shah Alam', '40400');

INSERT INTO customer VALUES ('C120398498', '012-2304859', 'Mr.', 'Kailash', 'Subramanian', '24, Jalan SS 21/52', 'Damansara Utama', '47400');

INSERT INTO customer VALUES ('C928390212', '013-2939348', 'Mrs.', 'Aarushi', 'Srivastava', '109, Jalan Terasek 7', 'Bangsar', '59100');

INSERT INTO orders VALUES ('DI2950478MQW', 'C696284290', '20-Mar-2023', TO\_TIMESTAMP('17:23:10', 'HH24:MI:SS'), 'dine-in', NULL, NULL, 'OL007');

INSERT INTO orders VALUES ('DE9407536QDS', 'C984027513', '04-Feb-2023', TO\_TIMESTAMP('19:21:57', 'HH24:MI:SS'), 'delivery', 'CC64270', 'DD75103', 'OL005');

INSERT INTO orders VALUES ('DI1583947JDG', 'C784039281', '01-Apr-2023', TO\_TIMESTAMP('11:12:39', 'HH24:MI:SS'), 'dine-in', NULL, NULL, 'OL005');

INSERT INTO orders VALUES ('DE5682930CRW', 'C532938109', '19-Jun-2023', TO\_TIMESTAMP('16:25:42', 'HH24:MI:SS'), 'delivery', 'CC34058', 'DD32104', 'OL004');

INSERT INTO orders VALUES ('DE7384956CDN', 'C984027513', '29-Mar-2023', TO\_TIMESTAMP('12:31:59', 'HH24:MI:SS'), 'delivery', 'CC23948', 'DD94275', 'OL005');

INSERT INTO orders VALUES ('DI4761852OFD', 'C239457108', '23-Jul-2023', TO\_TIMESTAMP('08:55:41', 'HH24:MI:SS'), 'dine-in', NULL, NULL, 'OL006');

INSERT INTO orders VALUES ('DE9023839NAS', 'C232934893', '18-Feb-2023', TO\_TIMESTAMP('13:28:09', 'HH24:MI:SS'), 'delivery', 'CC92683', 'DD09286', 'OL002');

INSERT INTO orders VALUES ('DI1082389ASD', 'C128302094', '24-Jun-2023', TO\_TIMESTAMP('16:03:12', 'HH24:MI:SS'), 'dine-in', NULL, NULL, 'OL002');

INSERT INTO orders VALUES ('DE2039485CAA', 'C930485752', '02-Feb-2023', TO\_TIMESTAMP('11:39:55', 'HH24:MI:SS'), 'delivery', 'CC19238', 'DD75103', 'OL005');

INSERT INTO orders VALUES ('DI6389284ACE', 'C124978532', '12-Apr-2023', TO\_TIMESTAMP('15:43:09', 'HH24:MI:SS'), 'dine-in', NULL, NULL, 'OL003');

INSERT INTO orders VALUES ('DE0123456ABC', 'C123456789', '02-Feb-2023', TO\_TIMESTAMP('13:23:05', 'HH24:MI:SS'), 'delivery', 'CC05297', 'DD79462', 'OL001');

INSERT INTO orders VALUES ('DI1092837CEB', 'C758203419', '08-Apr-2023', TO\_TIMESTAMP('18:22:44', 'HH24:MI:SS'), 'dine-in', NULL, NULL, 'OL003');

INSERT INTO orders VALUES ('DE0294982AOM', 'C923649123', '08-Apr-2023', TO\_TIMESTAMP('08:17:21', 'HH24:MI:SS'), 'delivery', 'CC14384', 'DD12038', 'OL007');

INSERT INTO orders VALUES ('DI6543210DEF', 'C023850342', '10-Jul-2023', TO\_TIMESTAMP('09:20:11', 'HH24:MI:SS'), 'dine-in', NULL, NULL, 'OL001');

INSERT INTO orders VALUES ('DI9028301HDS', 'C203770683', '06-Jul-2023', TO\_TIMESTAMP('09:47:18', 'HH24:MI:SS'), 'dine-in', NULL, NULL, 'OL003');

INSERT INTO orders VALUES ('DE0128302ASW', 'C239457108', '09-Jul-2023', TO\_TIMESTAMP('12:34:23', 'HH24:MI:SS'), 'delivery', 'CC02938', 'DD29382', 'OL006');

INSERT INTO orders VALUES ('DE2410956MDS', 'C370598463', '15-Mar-2023', TO\_TIMESTAMP('14:06:32', 'HH24:MI:SS'), 'delivery', 'CC34058', 'DD32104', 'OL004');

INSERT INTO orders VALUES ('DI3439248SDJ', 'C436285973', '16-Jan-2023', TO\_TIMESTAMP('18:23:04', 'HH24:MI:SS'), 'dine-in', NULL, NULL, 'OL004');

INSERT INTO orders VALUES ('DE1930248SVD', 'C124978532', '02=Jul-2023', TO\_TIMESTAMP('10:23:22', 'HH24:MI:SS'), 'delivery', 'CC29472', 'DD02938', 'OL003');

INSERT INTO orders VALUES ('DE0239419EJD', 'C696284290', '12-May-2023', TO\_TIMESTAMP('12:45:23', 'HH24:MI:SS'), 'delivery', 'CC12934', 'DD12038', 'OL007');

INSERT INTO orders VALUES ('DI9384274GHJ', 'C239179283', '12=Apr=2023', TO\_TIMESTAMP('10:12:28', 'HH24:MI:SS'), 'delivery', 'CC64270', 'DD94275', 'OL007');

INSERT INTO orders VALUES ('DE2349238SEH', 'C348239940', '08-Jun-2023', TO\_TIMESTAMP('12:29:34', 'HH24:MI:SS'), 'delivery', 'CC09374', 'DD83239', 'OL004');

INSERT INTO orders VALUES ('DI2934859ADE', 'C120398498', '30-May-2023', TO\_TIMESTAMP('20:29:34', 'HH24:MI:SS'), 'dine-in', NULL, NULL, 'OL001');

INSERT INTO orders VALUES ('DI1239424SKE', 'C928390212', '12-Mar-2023', TO\_TIMESTAMP('18:23:34', 'HH24:MI:SS'), 'dine-in', NULL, NULL, 'OL002');

INSERT INTO orders VALUES ('DI1293848AKI', 'C023850342', '12-Jul-2023', TO\_TIMESTAMP('12:18:24', 'HH24:MI:SS'), 'dine-in', NULL, NULL, 'OL001');

INSERT INTO orders VALUES ('DI3829478MHK', 'C370598463', '20-May-2023', TO\_TIMESTAMP('15:12:28', 'HH24:MI:SS'), 'dine-in', NULL, NULL, 'OL004');

INSERT INTO orders VALUES ('DI2946194UCS', 'C120398498', '17-Jun-2023', TO\_TIMESTAMP('10:12:34', 'HH24:MI:SS'), 'dine-in', NULL, NULL, 'OL001');

INSERT INTO orders VALUES ('DI7482349BVM', 'C239457108', '20-Jul-2023', TO\_TIMESTAMP('14:23:38', 'HH24:MI:SS'), 'dine-in', NULL, NULL, 'OL006');

INSERT INTO orders VALUES ('DE3849762KVN', 'C532938109', '28-Jul-2023', TO\_TIMESTAMP('17:37:12', 'HH24:MI:SS'), 'delivery', 'CC09374', 'DD32104', 'OL004');

INSERT INTO orders VALUES ('DE1298352ZLK', 'C436285973', '30-Jul-2023', TO\_TIMESTAMP('10:58:23', 'HH24:MI:SS'), 'delivery', 'CC34058', 'DD83239', 'OL004');

INSERT INTO orderlines VALUES ('DE9407536QDS', 'DR3450', 2);

INSERT INTO orderlines VALUES ('DE9407536QDS', 'SD3102', 3);

INSERT INTO orderlines VALUES ('DE9407536QDS', 'SD5164', 1);

INSERT INTO orderlines VALUES ('DE9407536QDS', 'PZ1001', 1);

INSERT INTO orderlines VALUES ('DE5682930CRW', 'DS9499', 1);

INSERT INTO orderlines VALUES ('DE5682930CRW', 'SD5164', 1);

INSERT INTO orderlines VALUES ('DE5682930CRW', 'SD3001', 1);

INSERT INTO orderlines VALUES ('DE5682930CRW', 'PT2001', 1);

INSERT INTO orderlines VALUES ('DE7384956CDN', 'DR2365', 4);

INSERT INTO orderlines VALUES ('DE7384956CDN', 'DS2467', 2);

INSERT INTO orderlines VALUES ('DE7384956CDN', 'SD3001', 3);

INSERT INTO orderlines VALUES ('DE7384956CDN', 'PT1001', 2);

INSERT INTO orderlines VALUES ('DE9023839NAS', 'PT2001', 2);

INSERT INTO orderlines VALUES ('DE9023839NAS', 'DR4467', 2);

INSERT INTO orderlines VALUES ('DE9023839NAS', 'DR3450', 2);

INSERT INTO orderlines VALUES ('DE9023839NAS', 'DR4001', 4);

INSERT INTO orderlines VALUES ('DE2039485CAA', 'DS5001', 2);

INSERT INTO orderlines VALUES ('DE0123456ABC', 'PZ2301', 3);

INSERT INTO orderlines VALUES ('DE0123456ABC', 'PT4101', 1);

INSERT INTO orderlines VALUES ('DE0123456ABC', 'PT3304', 1);

INSERT INTO orderlines VALUES ('DE0123456ABC', 'DS6072', 5);

INSERT INTO orderlines VALUES ('DE0294982AOM', 'PZ2301', 4);

INSERT INTO orderlines VALUES ('DE0294982AOM', 'PZ4301', 1);

INSERT INTO orderlines VALUES ('DE0294982AOM', 'DR3450', 2);

INSERT INTO orderlines VALUES ('DE0128302ASW', 'PZ9067', 2);

INSERT INTO orderlines VALUES ('DE0128302ASW', 'DR4001', 3);

INSERT INTO orderlines VALUES ('DE0128302ASW', 'PZ2301', 2);

INSERT INTO orderlines VALUES ('DE2410956MDS', 'PZ9067', 2);

INSERT INTO orderlines VALUES ('DE2410956MDS', 'DS9499', 3);

INSERT INTO orderlines VALUES ('DE2410956MDS', 'PT3101', 1);

INSERT INTO orderlines VALUES ('DE2410956MDS', 'SD5164', 2);

INSERT INTO orderlines VALUES ('DE2410956MDS', 'PT1001', 3);

INSERT INTO orderlines VALUES ('DI2950478MQW', 'DR3450', 1);

INSERT INTO orderlines VALUES ('DI2950478MQW', 'DR2365', 1);

INSERT INTO orderlines VALUES ('DI2950478MQW', 'DR4467', 2);

INSERT INTO orderlines VALUES ('DI2950478MQW', 'SD3001', 2);

INSERT INTO orderlines VALUES ('DI2950478MQW', 'PT1001', 2);

INSERT INTO orderlines VALUES ('DI1583947JDG', 'DR3450', 2);

INSERT INTO orderlines VALUES ('DI1583947JDG', 'PZ9067', 1);

INSERT INTO orderlines VALUES ('DI1583947JDG', 'PT3304', 1);

INSERT INTO orderlines VALUES ('DI1583947JDG', 'SD3001', 1);

INSERT INTO orderlines VALUES ('DI1583947JDG', 'DS5001', 1);

INSERT INTO orderlines VALUES ('DI1583947JDG', 'DS6072', 1);

INSERT INTO orderlines VALUES ('DI4761852OFD', 'PT2001', 1);

INSERT INTO orderlines VALUES ('DI1082389ASD', 'PZ1001', 1);

INSERT INTO orderlines VALUES ('DI1082389ASD', 'PT2001', 1);

INSERT INTO orderlines VALUES ('DI1082389ASD', 'DR3450', 2);

INSERT INTO orderlines VALUES ('DI6389284ACE', 'PZ2301', 4);

INSERT INTO orderlines VALUES ('DI6389284ACE', 'DS6072', 2);

INSERT INTO orderlines VALUES ('DI6389284ACE', 'SD5164', 1);

INSERT INTO orderlines VALUES ('DI6389284ACE', 'PT4101', 1);

INSERT INTO orderlines VALUES ('DI6389284ACE', 'DR3450', 6);

INSERT INTO orderlines VALUES ('DI1092837CEB', 'PT2001', 1);

INSERT INTO orderlines VALUES ('DI1092837CEB', 'PZ2301', 1);

INSERT INTO orderlines VALUES ('DI1092837CEB', 'SD3102', 1);

INSERT INTO orderlines VALUES ('DI1092837CEB', 'DR4001', 1);

INSERT INTO orderlines VALUES ('DI6543210DEF', 'PZ2301', 1);

INSERT INTO orderlines VALUES ('DI6543210DEF', 'DS2467', 2);

INSERT INTO orderlines VALUES ('DI6543210DEF', 'DR2365', 2);

INSERT INTO orderlines VALUES ('DI6543210DEF', 'PT2001', 3);

INSERT INTO orderlines VALUES ('DI6543210DEF', 'SD3001', 2);

INSERT INTO orderlines VALUES ('DI6543210DEF', 'DR4001', 1);

INSERT INTO orderlines VALUES ('DI9028301HDS', 'PT3101', 2);

INSERT INTO orderlines VALUES ('DI9028301HDS', 'PT4101', 1);

INSERT INTO orderlines VALUES ('DI9028301HDS', 'DR4467', 1);

INSERT INTO orderlines VALUES ('DI9028301HDS', 'SD5164', 1);

INSERT INTO orderlines VALUES ('DI9028301HDS', 'PT1001', 3);

INSERT INTO orderlines VALUES ('DI9028301HDS', 'DS6072', 2);

INSERT INTO orderlines VALUES ('DI3439248SDJ', 'DS5001', 1);

INSERT INTO orderlines VALUES ('DI3439248SDJ', 'PT2001', 2);

INSERT INTO orderlines VALUES ('DI3439248SDJ', 'PZ2301', 1);

INSERT INTO orderlines VALUES ('DI3439248SDJ', 'PT4101', 1);

INSERT INTO orderlines VALUES ('DI3439248SDJ', 'PT3304', 2);

INSERT INTO orderlines VALUES ('DI3439248SDJ', 'SD3001', 1);

INSERT INTO orderlines VALUES ('DE1930248SVD', 'PZ2301', 3);

INSERT INTO orderlines VALUES ('DE1930248SVD', 'PZ1001', 3);

INSERT INTO orderlines VALUES ('DE1930248SVD', 'DR4001', 4);

INSERT INTO orderlines VALUES ('DE1930248SVD', 'DR3450', 4);

INSERT INTO orderlines VALUES ('DE0239419EJD', 'DS6072', 2);

INSERT INTO orderlines VALUES ('DI9384274GHJ', 'DR3450', 2);

INSERT INTO orderlines VALUES ('DI9384274GHJ', 'DR4001', 1);

INSERT INTO orderlines VALUES ('DI9384274GHJ', 'DS2467', 1);

INSERT INTO orderlines VALUES ('DI9384274GHJ', 'PZ1001', 2);

INSERT INTO orderlines VALUES ('DI9384274GHJ', 'DS5001', 1);

INSERT INTO orderlines VALUES ('DE2349238SEH', 'DR4467', 6);

INSERT INTO orderlines VALUES ('DE2349238SEH', 'PZ4301', 4);

INSERT INTO orderlines VALUES ('DE2349238SEH', 'DS2467', 2);

INSERT INTO orderlines VALUES ('DE2349238SEH', 'DS9499', 1);

INSERT INTO orderlines VALUES ('DE2349238SEH', 'SD3001', 3);

INSERT INTO orderlines VALUES ('DE2349238SEH', 'DR3450', 1);

INSERT INTO orderlines VALUES ('DI2934859ADE', 'PT1001', 2);

INSERT INTO orderlines VALUES ('DI2934859ADE', 'DS5001', 2);

INSERT INTO orderlines VALUES ('DI2934859ADE', 'PT4101', 2);

INSERT INTO orderlines VALUES ('DI2934859ADE', 'PT3101', 2);

INSERT INTO orderlines VALUES ('DI2934859ADE', 'DR4001', 5);

INSERT INTO orderlines VALUES ('DI2934859ADE', 'PZ9067', 1);

INSERT INTO orderlines VALUES ('DI1239424SKE', 'SD3102', 1);

INSERT INTO orderlines VALUES ('DI1239424SKE', 'SD5164', 1);

INSERT INTO orderlines VALUES ('DI1239424SKE', 'SD3001', 1);

INSERT INTO orderlines VALUES ('DI1239424SKE', 'PT2001', 1);

INSERT INTO orderlines VALUES ('DI1293848AKI', 'PZ4301', 1);

INSERT INTO orderlines VALUES ('DI1293848AKI', 'DS5001', 1);

INSERT INTO orderlines VALUES ('DI3829478MHK', 'PT3101', 1);

INSERT INTO orderlines VALUES ('DI3829478MHK', 'SD3102', 2);

INSERT INTO orderlines VALUES ('DI2946194UCS', 'PZ2301', 1);

INSERT INTO orderlines VALUES ('DI2946194UCS', 'DR2365', 1);

INSERT INTO orderlines VALUES ('DI7482349BVM', 'PT3304', 2);

INSERT INTO orderlines VALUES ('DI7482349BVM', 'DS2467', 2);

INSERT INTO orderlines VALUES ('DI7482349BVM', 'SD5164', 1);

INSERT INTO orderlines VALUES ('DI7482349BVM', 'DR3450', 2);

INSERT INTO orderlines VALUES ('DE3849762KVN', 'SD3001', 2);

INSERT INTO orderlines VALUES ('DE3849762KVN', 'DS5001', 1);

INSERT INTO orderlines VALUES ('DE3849762KVN', 'PZ9067', 1);

INSERT INTO orderlines VALUES ('DE1298352ZLK', 'DS6072', 1);

INSERT INTO orderlines VALUES ('DE1298352ZLK', 'DR4467', 3);

INSERT INTO orderlines VALUES ('DE1298352ZLK', 'PT4101', 3);

INSERT INTO orderlines VALUES ('DE1298352ZLK', 'PZ4301', 2);

INSERT INTO vehicles VALUES ('ABC 1234', 'Honda', 'Motorbike', '21-Jan-2023', '29-Mar-2023', 4665,'OL005');

INSERT INTO vehicles VALUES ('DEF 456', 'Honda', 'Motorbike', '12-Feb-2021', '02-Jul-2023', 52103,'OL002');

INSERT INTO vehicles VALUES ('HGI 5566', 'Yamaha', 'Motorbike', '27-Apr-2021','27-Apr-2023', 22210,'OL001');

INSERT INTO vehicles VALUES ('BNC 9997', 'Honda', 'Car','01-Feb-2023','30-Jun-2023', 4112,'OL003');

INSERT INTO vehicles VALUES ('PNP 2234', 'Yamaha', 'Motorbike', '28-Mar-2019', '02-Mar-2023', 200002,'OL005');

INSERT INTO vehicles VALUES ('WSP 9801', 'Yamaha', 'Motorbike', '01-Jun-2023', '16-Jul-2023', 659,'OL004');

INSERT INTO vehicles VALUES ('SDE 2016', 'Yamaha', 'Motorbike', '04-Jan-2021', '04-Jul-2023', 86123,'OL002');

INSERT INTO vehicles VALUES ('PTP 2012', 'Perodua', 'Car', '05-Jan-2022','05-Jun-2023', 10020,'OL003');

INSERT INTO vehicles VALUES ('BSS 4034', 'Yamaha', 'Motorbike', '26-Feb-2023','06-May-2023', 999,'OL001');

INSERT INTO vehicles VALUES ('WQS 151', 'Honda', 'Motorbike', '26-Aug-2021', '22-Jun-2023', 54560, 'OL004');

INSERT INTO vehicles VALUES ('WTY 8690', 'Yamaha', 'Motorbike', '22-Jan-2022', '30-Apr-2023', 25680, 'OL003');

INSERT INTO vehicles VALUES ('BPN 4468', 'Yamaha', 'Motorbike', '26-Dec-2021', '19-Mar-2023', 49800, 'OL007');

INSERT INTO vehicles VALUES ('SYE 6666', 'Perodua', 'Car', '01-Jan-2023', '26-Jul-2023', 6900, 'OL007');

INSERT INTO vehicles VALUES ('BNB 7779', 'Honda', 'Motorbike', '17-Feb-2022', '01-Dec-2022', 10972, 'OL004');

INSERT INTO vehicles VALUES ('JOR 7997', 'Yamaha', 'Motorbike', '04-Apr-2023', '06-Jul-2023', 579, 'OL001');

INSERT INTO vehicles VALUES ('JYP 3400', 'Honda', 'Car', '06-Sep-2019', '25-Dec-2022', 245601, 'OL002');

INSERT INTO vehicles VALUES ('SQW 5677', 'Yamaha', 'Motorbike', '09-May-2022', '30-May-2023', 12309, 'OL006');

INSERT INTO vehicles VALUES ('SMT 5900', 'Yamaha', 'Motorbike', '10-Nov-2021', '07-Feb-2023', 54061.32, 'OL006');

INSERT INTO vehicles VALUES ('PKL 9963', 'Yamaha', 'Motorbike', '29-Jul-2019', '05-Jul-2023', 167129, 'OL006');

INSERT INTO vehicles VALUES ('WJE 210', 'Honda', 'Car', '03-Mar-2019', '23-Mar-2023', 172110, 'OL006');

INSERT INTO feedback VALUES ('QEW2345678', 'C984027513', 'DE9407536QDS', 5, 'The pizza same as the picture', '04-Feb-2023');

INSERT INTO feedback VALUES ('AAA3246751', 'C532938109', 'DE5682930CRW', 5, 'Value for money', '20-Jun-2023');

INSERT INTO feedback VALUES ('BWS5900125', 'C984027513', 'DE7384956CDN', 4, 'Fast delivery', '29-Mar-2023');

INSERT INTO feedback VALUES ('PYS1994328', 'C232934893', 'DE9023839NAS', 3, 'Ordinary taste', '20-Feb-2023');

INSERT INTO feedback VALUES ('JCK1910909', 'C930485752', 'DE2039485CAA', 5, NULL, '03-Feb-2023');

INSERT INTO feedback VALUES ('BBM2067917', 'C123456789', 'DE0123456ABC', 4, 'The pizza same as the picture', '02-Feb-2023');

INSERT INTO feedback VALUES ('SWT2010710', 'C923649123', 'DE0294982AOM', 4, 'Delicious', '12-Apr-2023');

INSERT INTO feedback VALUES ('FFD5689012', 'C239457108', 'DE0128302ASW', 5, 'Fast delivery', '10-Feb-2023');

INSERT INTO feedback VALUES ('DFG2839407', 'C370598463', 'DE2410956MDS', 5, NULL, '15-Mar-2023');

INSERT INTO feedback VALUES ('ABB6703221', 'C124978532', 'DE1930248SVD', 3, 'It was not terrible, but it did not quite meet my expectations.', '02-Feb-2023');

INSERT INTO feedback VALUES ('BQA2690120', 'C696284290', 'DE0239419EJD', 5, 'The pizza same as the picture', '15-May-2023');

INSERT INTO feedback VALUES ('YGA2561122', 'C348239940', 'DE2349238SEH', 5, NULL, '8-Jun-2023');

INSERT INTO feedback VALUES ('UIO7781245', 'C120398498', 'DI2934859ADE', 5, 'Delicious', '30-May-2023');

INSERT INTO feedback VALUES ('MNL4421941', 'C928390212', 'DI1239424SKE', 5, NULL, '12-Mar-2023');

INSERT INTO feedback VALUES ('UPP8001760', 'C023850342', 'DI6543210DEF', 3, 'The food was average; some dishes were good, while others fell short in taste.', '10-Jun-2023');

INSERT INTO feedback VALUES ('RTZ6602934', 'C436285973', 'DE1298352ZLK', 2, 'The toppings are all over the place!', '30-Jul-2023');

INSERT INTO feedback VALUES ('TYW2910547', 'C436285973', 'DI3439248SDJ', 4, NULL, '16-Jan-2023');

INSERT INTO feedback VALUES ('QQI8678034', 'C239179283', 'DI9384274GHJ', 4, NULL, '12-Apr-2023');

INSERT INTO feedback VALUES ('KIM7943276', 'C696284290', 'DI2950478MQW', 5, 'Value for money', '20-Mar-2023');

INSERT INTO feedback VALUES ('KLM3659122', 'C784039281', 'DI1583947JDG', 5, NULL, '1-Apr-2023');

INSERT INTO ingredients VALUES ('FL100', 'Flour', 1000);

INSERT INTO ingredients VALUES ('SA230', 'Sauce', 1000);

INSERT INTO ingredients VALUES ('DY450', 'Dairy', 950);

INSERT INTO ingredients VALUES ('ME234', 'Meats', 500);

INSERT INTO ingredients VALUES ('VE257', 'Vegetable', 500);

INSERT INTO ingredients VALUES ('HE235', 'Herbs', 400);

INSERT INTO ingredients VALUES ('SP765', 'Spices', 300);

INSERT INTO ingredients VALUES ('SE657', 'Seafoods', 500);

INSERT INTO ingredients VALUES ('CD465', 'Carbonated Drinks',1000);

INSERT INTO ingredients VALUES ('CB836', 'Cake Base',3450);

INSERT INTO ingredients VALUES ('PN324', 'Pasta Noodles', 512);

INSERT INTO ingredients VALUES ('CO983', 'Coooking Oil', 742);

INSERT INTO ingredients VALUES ('ET293', 'Exotic Toppings', 264);

INSERT INTO ingredients VALUES ('CH346', 'Chocolate' , 456);

INSERT INTO ingredients VALUES ('FF930', 'Fresh Fruits', 180);

INSERT INTO ingredients VALUES ('IC937', 'Ice Cream', 200);

INSERT INTO ingredients VALUES ('JU127', 'Juice Drinks', 389);

INSERT INTO ingredients VALUES ('SD282', 'Stimulant Drinks', 500);

INSERT INTO ingredients VALUES ('NS039', 'Nuts and Seeds', 380);

INSERT INTO ingredients VALUES ('ST946', 'Sweetening Agent', 210);

INSERT INTO suppliers VALUES ('SP001', 'FL100', 'FlourCo Sdn Bhd', '012-6728875', 'info@flourco.com');

INSERT INTO suppliers VALUES ('SP002', 'SA230', 'SauceMasters Trading (M) Sdn Bhd', '018-9003728', 'sales@saucemasters.com');

INSERT INTO suppliers VALUES ('SP003', 'DY450', 'DairyDelights MY', '03-52738599', 'contact@dairydelights.com');

INSERT INTO suppliers VALUES ('SP004', 'ME234', 'MeatSupreme MY', '013-4675821', 'info@meatsupreme.com');

INSERT INTO suppliers VALUES ('SP005', 'VE257', 'VeggieFresh Supplies (M) Sdn Bhd', '03-62341234', 'sales@veggiefreshsupplies.com');

INSERT INTO suppliers VALUES ('SP006', 'SP765', 'SpiceEmporium (M) Sdn Bhd', '016-2550978', 'contact@spiceemporium.com');

INSERT INTO suppliers VALUES ('SP007', 'SE657', 'SeafoodHarvesters MY', '012-4567890', 'info@seafoodharvesters.com');

INSERT INTO suppliers VALUES ('SP008', 'CD465', 'BeverageKings Sdn Bhd', '011-9987334', 'sales@beveragekings.com');

INSERT INTO suppliers VALUES ('SP009', 'CB836', 'FlourMillers MY', '03-51234567', 'contact@flourmillers.com');

INSERT INTO suppliers VALUES ('SP010', 'HE235', 'GlobalSauces Trading (M) Sdn Bhd', '014-69337841', 'info@globalsauces.com');

INSERT INTO suppliers VALUES ('SP011', 'PN324', 'PastaNoodle World (M) Sdn Bhd', '03-51920607', 'info@pastanoodleworld.com');

INSERT INTO suppliers VALUES ('SP012', 'CO983', 'HealthyCooking Oils', '010-2345678', 'contact@healthycookingoils.com');

INSERT INTO suppliers VALUES ('SP013', 'ET293', 'ExoticPizzas MY', '012-43215678', 'sales@exoticpizzas.com');

INSERT INTO suppliers VALUES ('SP014', 'CH346', 'ChocoDelights Trading (M) Sdn Bhd', '016-98765432', 'info@chocodelights.com');

INSERT INTO suppliers VALUES ('SP015', 'FF930', 'FreshFruits MY', '011-65437890', 'contact@freshfruits.com');

INSERT INTO suppliers VALUES ('SP016', 'IC937', 'IceCream Paradise (M) Sdn Bhd', '014-92301673', 'sales@icecreamparadise.com');

INSERT INTO suppliers VALUES ('SP017', 'JU127', 'JuiceMania Enterprise', '019-67129837', 'info@juicemania.com');

INSERT INTO suppliers VALUES ('SP018', 'SD282', 'BeverageZone MY', '017-09129234', 'contact@beveragezone.com');

INSERT INTO suppliers VALUES ('SP019', 'NS039', 'NuttyNuts Trading (M) Sdn Bhd', '011-22438847', 'sales@nuttynuts.com');

INSERT INTO suppliers VALUES ('SP020', 'ST946', 'SweetTreats Sdn Bhd', '014-33188374', 'info@sweettreats.com');

INSERT INTO suppliers VALUES ('SP021', 'SA230', 'SauceFusion Enterprise', '03-51920551', 'saucefusion@email.com');

INSERT INTO suppliers VALUES ('SP022', 'DY450', 'CheezyMansion Sbn Bhd', '012-3953763', 'cheezy\_mansion@gmail.com');

INSERT INTO suppliers VALUES ('SP023', 'ME234', 'MeatLover MY', '010-4026301', 'meat\_lover@gmail.com');

INSERT INTO suppliers VALUES ('SP024', 'VE257', 'Up Grocer Sbn Bhd', '03-51930890', 'up\_grocer@gmail.com');

INSERT INTO suppliers VALUES ('SP025', 'PN324', 'Italiano Enterprise', '03-51916783', 'italiano\_enterprise@gmail.com');

INSERT INTO suppliers VALUES ('SP026', 'CD465', 'Bubbly Beverage Co.', '017-2384723', 'info@bubblybeverage.com');

INSERT INTO suppliers VALUES ('SP027', 'JU127', 'FreshSips Beverages', '014-9436274', 'contact@freshsips.com');

INSERT INTO suppliers VALUES ('SP028', 'SD282', 'EnergizeBrew Co.', '03-273482373', 'sales@energizebrewco.com');

INSERT INTO suppliers VALUES ('SP029', 'CD465', 'Fizztastic Beverages', '012-3456789', 'info@fizztasticbeverages.com');

INSERT INTO suppliers VALUES ('SP030', 'JU127', 'ZestyJuice Delights', '011-23456789', 'info@zestyjuicedelights.com');

INSERT INTO suppliers VALUES ('SP040', 'SD282', 'ReviveBrew Co.', '016-7890123', 'contact@revivebrewco.com');

INSERT INTO supplylines VALUES ('OL005', 'SP002', 'daily');

INSERT INTO supplylines VALUES ('OL002', 'SP002', 'daily');

INSERT INTO supplylines VALUES ('OL001', 'SP010', 'monthly');

INSERT INTO supplylines VALUES ('OL003', 'SP008', 'weekly');

INSERT INTO supplylines VALUES ('OL005', 'SP008', 'weekly');

INSERT INTO supplylines VALUES ('OL004', 'SP006', 'weekly');

INSERT INTO supplylines VALUES ('OL002', 'SP005', 'monthly');

INSERT INTO supplylines VALUES ('OL003', 'SP004', 'monthly');

INSERT INTO supplylines VALUES ('OL001', 'SP003', 'daily');

INSERT INTO supplylines VALUES ('OL006', 'SP001', 'weekly');

INSERT INTO supplylines VALUES ('OL007', 'SP009', 'daily');

INSERT INTO supplylines VALUES ('OL006', 'SP007', 'daily');

INSERT INTO supplylines VALUES ('OL001', 'SP011', 'weekly');

INSERT INTO supplylines VALUES ('OL001', 'SP012', 'monthly');

INSERT INTO supplylines VALUES ('OL002', 'SP013', 'weekly');

INSERT INTO supplylines VALUES ('OL007', 'SP014', 'weekly');

INSERT INTO supplylines VALUES ('OL004', 'SP015', 'daily');

INSERT INTO supplylines VALUES ('OL006', 'SP016', 'weekly');

INSERT INTO supplylines VALUES ('OL003', 'SP017', 'weekly');

INSERT INTO supplylines VALUES ('OL005', 'SP018', 'monthly');

INSERT INTO supplylines VALUES ('OL004', 'SP019', 'monthly');

INSERT INTO supplylines VALUES ('OL005', 'SP020', 'weekly');

INSERT INTO supplylines VALUES ('OL003', 'SP013', 'daily');

INSERT INTO supplylines VALUES ('OL001', 'SP021', 'weekly');

INSERT INTO supplylines VALUES ('OL003', 'SP022', 'daily');

INSERT INTO supplylines VALUES ('OL002', 'SP023', 'weekly');

INSERT INTO supplylines VALUES ('OL007', 'SP024', 'daily');

INSERT INTO supplylines VALUES ('OL005', 'SP025', 'weekly');

INSERT INTO supplylines VALUES ('OL007', 'SP020', 'monthly');

INSERT INTO supplylines VALUES ('OL007', 'SP026', 'weekly');

INSERT INTO supplylines VALUES ('OL007', 'SP027', 'weekly');

INSERT INTO supplylines VALUES ('OL003', 'SP028', 'weekly');

INSERT INTO supplylines VALUES ('OL002', 'SP029', 'weekly');

INSERT INTO supplylines VALUES ('OL002', 'SP030', 'weekly');

INSERT INTO supplylines VALUES ('OL007', 'SP040', 'weekly');

/\*As part of our customer service tradition to provide personalized monthly order summaries to valued customers,

we have decided to create an index on the order\_date column. This decision is based on the frequent retrieval of

order information for this purpose. With the index in place, the database engine can quickly locate the relevant orders,

making the query execution faster and more efficient.\*/

CREATE INDEX idx\_order\_date ON orders (order\_date);

/\*This index improves the database's ability to swiftly locate and access relevant data from the orderlines table during outlet

sales analysis. This optimization is essential because sales and outlet performance analysis is performed on a monthly basis and

involves aggregating and processing large volumes of data. \*/

CREATE INDEX idx\_orderlines\_revenue ON orderlines (orderID, itemID, quantity);

/\*provides information about staff members who have interacted with customers through orders\*/

CREATE VIEW staff\_customer\_view AS

SELECT sf.staffID, sf.first\_name AS staff\_first\_name, sf.last\_name AS staff\_last\_name, c.customerID, c.street AS customer\_street, c.suburb AS customer\_suburb

FROM staff sf, orders odr, customer c

WHERE (sf.staffID = odr.telephone\_operator OR sf.staffID = odr.riderID)

AND odr.customerID = c.customerID;

/\*provides a summary of monthly revenue for each outlet based on the last month of order data\*/

CREATE VIEW monthly\_revenue\_view AS

SELECT ol.outletID, TO\_CHAR(odr.order\_date, 'Mon-YYYY') AS month, SUM(odrln.quantity \* m.price) AS total\_revenue

FROM outlets ol, orders odr, orderlines odrln, menu m

WHERE ol.outletID = odr.outletID

AND odr.orderID = odrln.orderID

AND odrln.itemID = m.itemID

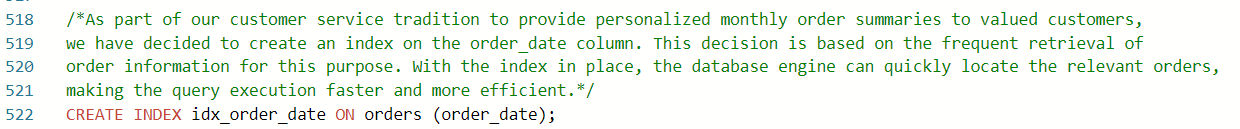
AND oDR.order\_date >= TRUNC(ADD\_MONTHS(SYSDATE, -1), 'MONTH') -- Filter orders for the last month

AND oDR.order\_date < TRUNC(SYSDATE, 'MONTH') -- Exclude any orders from the current month

GROUP BY ol.outletID, TO\_CHAR(odr.order\_date, 'Mon-YYYY');

**Part 4: Query a database (each sub-part carries 4 marks; max of 20 marks)**

|  |
| --- |
| **QUERY 1**  **Scenario: Monthly order summary for customers**  As Picky Pizza Palace management team, part of our customer service tradition is to provide monthly order summaries to our valued customers. For instance, let's take David (customerID = 'C239457108'), one of our loyal patrons. We are delighted to offer David a personalized monthly order summary for his reference and convenience. This summary will provide the total number of orders he has placed with us over the last 30 days. |



**Figure 1.** Creating an Index on the Orders Table for Order Date Column

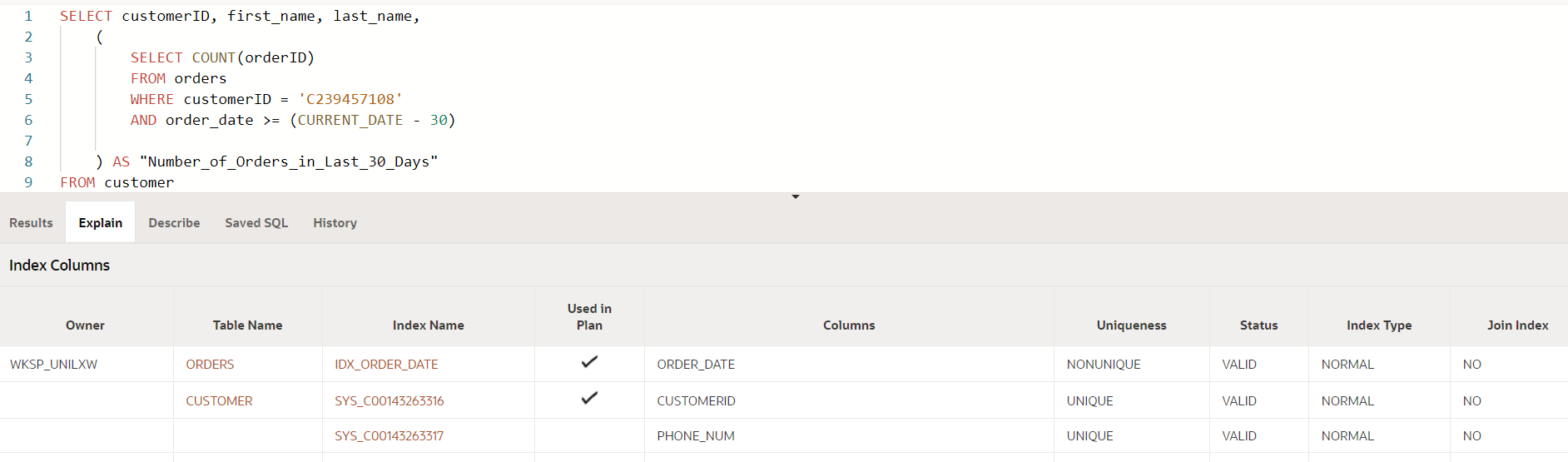
Figure 1 presents SQL code to create an index named idx\_order\_date on the order\_date column in the orders table. This index will improve the performance of queries that involve filtering or searching based on the order\_date, which is essential for generating monthly order summaries for customers.

A screenshot of a computer

Description automatically generated

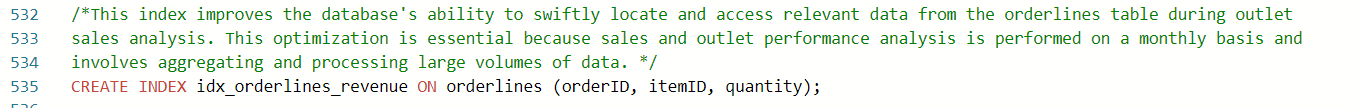
**Figure 2.** User Query and Result for Customer 'C239457108'

In Figure 2, the SQL query retrieves specific information about a customer named David, identified by the customerID 'C239457108'. The query includes his first name and last name. By using a sub-query, the query calculates the number of orders placed by David in the last 30 days using the COUNT function. The main query filters the results to show details solely for David, ensuring a focused and personalized summary of his recent order activity.



**Figure 3.** Verification of Index Usage for idx\_order\_date

|  |
| --- |
| **QUERY 2**  **Scenario: Sales Analysis and Outlet Performance**  As Picky Pizza Palace management team, we want to analyse sales performance across different outlets and identify the outlets with the highest revenue within the last 90 days. Additionally, we want to assess customer satisfaction levels at these outlets to gain valuable insights into our overall performance. |



**Figure 4.** idx\_orderlines\_revenue Index Creation

Figure 2 shows the SQL code to create the idx\_orderlines\_revenue index on the orderlines table, targeting the orderID, itemID, and quantity columns. This index enhances the performance of queries involving revenue calculations, which are vital for analyzing outlet sales and performance.

A screenshot of a computer code

Description automatically generated

**Figure 5.** User Query for Question b.

A screenshot of a computer

Description automatically generated

**Figure 6.** Result for Figure 4 SQL Query

Figure 5 presents a user query that analyses sales performance and customer satisfaction for different outlets. It retrieves outlet information, the number of orders, total revenue, and average customer rating (Figure 6). The use of "COUNT (DISTINCT odr.orderID)" ensures accurate counting of unique orders without overcounting. The query involves a 5-table join (outlets, orders, orderlines, menu, and feedback) to connect relevant data for comprehensive analysis. A left join is used to include all orders, even if no feedback is associated. The query focuses on recent orders (last 90 days) and outlets with at least 2 orders. The results are grouped by outlet ID and address and sorted by total revenue in descending order.

A screenshot of a computer

Description automatically generated

**Figure 7.** Verification of Index Usage for idx\_orderlines\_revenue

|  |
| --- |
| **QUERY 3**  **Scenario: Customer Feedback Analysis to Improve Outlet Performance**  Picky Pizza Palace operates multiple outlets across different suburbs, and our goal is to identify outlets that have experienced lower total sales in the previous month, particularly those with total revenue below $200. To address this issue, we will analyse customer feedback and revenue performance for these underperforming outlets. |

Our focus is on feedback received for orders made in the past month and with review scores of 3 or lower or is NULL, as it helps us address customer concerns and improve overall satisfaction. The inclusion of NULL score reviews ensures that all orders with scores other than excellent (4 and 5) made at the outlets are considered, even if some customers did not provide any feedback. This approach is essential for maintaining data completeness and avoiding the oversight of potentially valuable information. This analysis will enable us to optimize strategies and allocate resources effectively to enhance sales and customer experience in these specific locations.

A screenshot of a computer code

Description automatically generated

**Figure 8.** Created monthly\_revenue\_view View in Script

The "monthly\_revenue\_view" provides monthly revenue summaries for outlets based on the previous month’s order data, including outletID, month (Mon-YYYY), and total revenue (Figure 7). It aids in quick sales analysis and performance evaluation.

A screenshot of a computer code

Description automatically generated

**Figure 9.** User Query for Question c.

A screenshot of a computer

Description automatically generated

**Figure 10.** Result for Figure 7 Query

The SQL query in Figure 8 retrieves past month order details, feedback information, and outlet total revenue. It filters data based on revenue, date range, and feedback scores. The right join ensures all orders are included, regardless of whether there is a matching row in the "feedback" table or not. The "monthly\_revenue\_view" is used to obtain monthly revenue data for outlets, aiding in the analysis of feedback-outlet relationships and performance evaluation.

|  |
| --- |
| **QUERY 4**  **Scenario: Personnel Management in Outlets**  Picky Pizza Palace runs multiple pizza outlets, each outlet operates with its dedicated staff. Recently, Outlet 'OL005' appointed a new store manager with the staffID 'SM12345.' To ensure smooth and efficient personnel management, the store manager requires access to crucial information about all workers operating at 'OL005.' |

A screenshot of a computer code

Description automatically generated

**Figure 11.** User Query for Scenario

The query combines data from the 'staff' table with itself to match staff members sharing the same outletID, which is OL005 in this scenario, with the store manager having staffID 'SM12345.' By utilizing the INNER JOIN clause on the 'outletID' field, it associates employee records (sf) with their corresponding manager records (sm) based on the shared outletID. The SELECT clause is used to display essential information, including staffID, staff's first\_name, last\_name, position, phone number, the corresponding manager's information, and the outletID. The condition "sf.position NOT LIKE '%Manager%'" is used to filter out manager records, ensuring that only non-managerial staff members are displayed. This helps focus on the team members under the manager while excluding the manager's own record from the results.

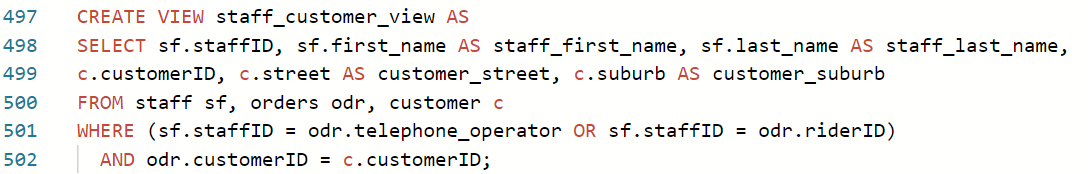
A screenshot of a computer

Description automatically generated

**Figure 12.** Result for Figure 10

The result set includes data about workers at the same outlet as the manager ('SM12345'). It shows staffID, first name, last name, position, phone number, manager's name and staffID, and all outletIDs with 'OL005'. The query provides the manager with a concise list of their team members' data, promoting better management and coordination to enhance outlet performance.

|  |
| --- |
| **QUERY 5:**  **For case scenario 1 Picky Pizza Palace, list the staff whose surnames start with ‘Ch’ and who have served customers with addresses containing ‘Jaya’.** |



**Figure 13.** View created in our database script

A screenshot of a computer

Description automatically generated

**Figure 14.** Fields available in our staff\_customer\_view

According to Figure 12 and Figure 13, the staff\_customer\_view is created by joining 3 different tables such as **Staff** sf, **Orders** odr and **Customer** c, with 2 joining conditions. The attributes selected from Staff table are staffID, staff\_first\_name and staff\_last\_name; whereas attributes selected from Customer table are customer\_ID, customer\_street and customer\_suburb. These attributes are chosen to be displayed in the view to restrict data access. It is achieved where users are unable to see the other attributes of the tables that are confidential, which in this case would be the customer first and last name that should be protected for privacy purposes.

A white background with black and red text

Description automatically generated

**Figure 15.** SQL statement

A screenshot of a computer

Description automatically generated

**Figure 16.** Results

When we run this, staff members that satisfies the user conditions will be provided. Attributes like staffID, first\_name and last\_name are selected to be displayed in the result, and the data are filtered from the ‘staff\_customer\_view’ view. The ‘WHERE’ clause is to check for conditions where last name of staff starts with ‘Ch’, and either the suburb or street address of customer consist of ‘Jaya’, both by using the like operator and wildcard character.

Part 5: Presentation & Teamwork (20 marks)

~~~~~~~~~~ end of all parts of the final assessment~~~~~~~~~~