DATABASE PROPOSAL FOR FRESH FAIR OASIS

Student Name: Russell, Abhinav, Avimannu, Arjun, Dipanjali

Student Number: C0927696, C0926240, CO927513, C0927403, C0929372

Project name: Database Design for Fresh Fair Oasis

Term: Winter 2024

Professor: Mehrnoush Ashrafi

TABLE OF CONTENTS

1. COMPANY NAME (H)

Fresh Fair Oasis

1.1 FRESH FAIR OASIS OVERVIEW/DESCRIPTION (H)

Fresh Fair Oasis stands out as an innovative grocery destination dedicated to transforming the way customers shop, boasting an extensive selection of fresh and organic items sourced directly from nearby farmers and producers. Our establishment offers a warm and inviting environment, inviting patrons to discover an array of premium fruits, vegetables, meats, dairy goods, and pantry staples.

At Fresh Fair Oasis, we place a strong emphasis on sustainability and community involvement. We are committed to eco-conscious practices, including reducing plastic waste through minimized packaging and advocating for reusable containers. Moreover, we actively collaborate with local farmers' markets and participate in community gatherings to cultivate a sense of community and connection among our clientele.

2. PRODUCT (H)

Fresh produce, packaged goods, dairy products, beverages, household items, and more.

2.1 PRODUCT ATTRIBUTES (H)

1. Product Name
2. Bar Code
3. Category (e.g. fruits, vegetables, dairy, meat, etc.)
4. Description
5. Price
6. Brand
7. Calories
8. Protein
9. Volume
10. Country of Origins

2.2 UNIQUE IDENTIFIER (H)

- Product ID

3. CUSTOMER SALES INVOICE(H)

Fresh Fair Oasis

Company Overview:

Fresh Fair Oasis stands out as an innovative grocery destination dedicated to transforming the way customers shop, boasting an extensive selection of fresh and organic items sourced directly from nearby farmers and producers. Our establishment offers a warm and inviting environment, inviting patrons to discover an array of premium fruits, vegetables, meats, dairy goods, and pantry staples.

At Fresh Fair Oasis, we place a strong emphasis on sustainability and community involvement. We are committed to eco-conscious practices, including reducing plastic waste through minimized packaging and advocating for reusable containers. Moreover, we actively collaborate with local farmers' markets and participate in community gatherings to cultivate a sense of community and connection among our clientele.

Product: Milk

Product Attributes:

Attributes	Data
Product ID	113
Product Name	Milk
Category	Dairy
Description	Organic whole milk, 1 gallon
Price	$3.99
Brand	Creamy Fields
Calories per 1 cup	160 calories
Protein per 1 cup	9 grams
Volume	1 Gallon
Country of Origins	Canada

Fresh Fair Oasis

123 Main Street

Toronto, ON M5V 2R9

+1 (416) 555-7890

Customer Sales Invoice

Customer ID: 10

Customer Name: SavorBite Bistro

Customer Address: 135 Dundas Street
Toronto, ON M7S 5K3

Order ID: 12389

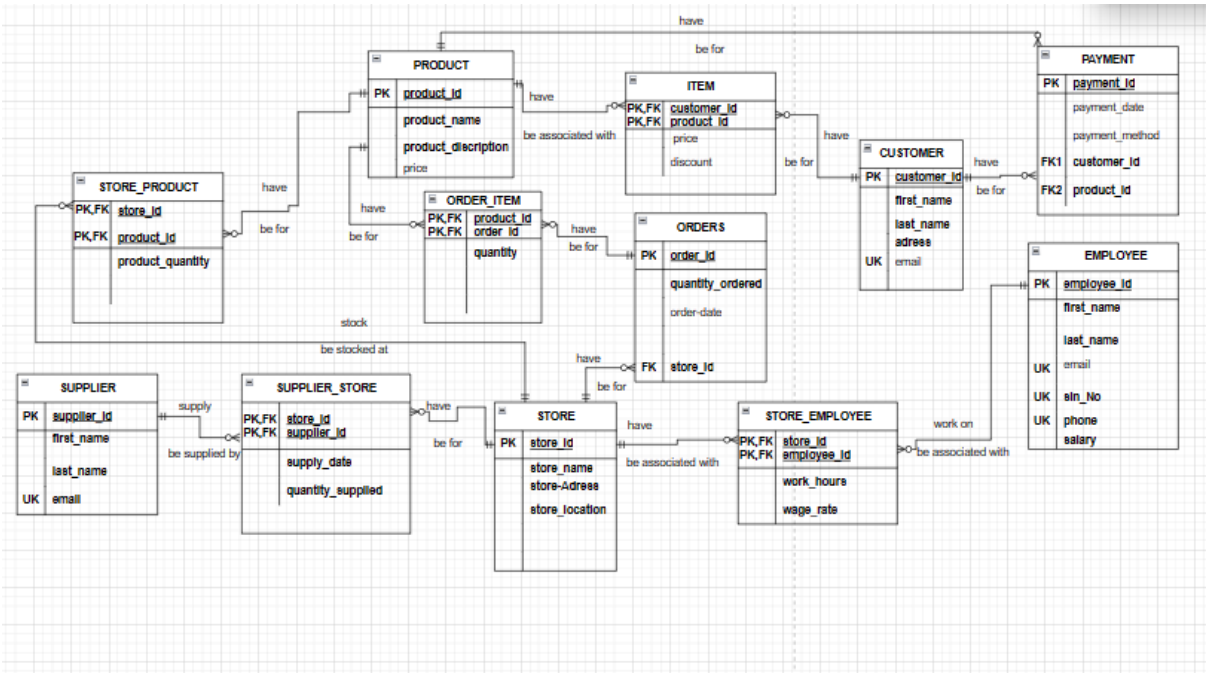
Order Date: 4/04/2024

Associate ID: 9

Associate Name: Aahori Thapa

Product ID	Product name	Category	Item Description	Qty	Unit Price	Total Price
114	Eggs	Dairy	Farm-fresh large brown eggs, dozen	5 dozen	$2.79/dozen	$13.95
119	Tomatoes	Produce	Vine-ripened tomatoes, pack of 6	4 packs	$3.49/pack	$13.96
117	Rice	Pasta & Rice	Long grain white rice, 5 lbs bag	3 bags	$4.99/bag	$14.97
116	Chicken Breast	Meat & Poultry	Boneless skinless chicken breast, 1 lb	15 lbs	$5.99/lbs	$89.85
115	Bread	Bakery	Whole wheat bread, sliced, 24 oz	4 loaves	$2.49/loaf	$9.96
120	Spinach	Produce	Fresh organic spinach, 8 oz bag	3 bags	$2.99/lbs	$8.97
					Sub-total	$151.66
					Tax (13%)	$19.72
					Total	$171.38

4. ENTITY RELATIONSHIP DIAGRAM (H)



4.1 ER LANGUAGE (H)

EACH SUPPLIER may supply zero one or many SUPPLIER_STOREs

EACH SUPPLIER-STORE must be supplied by one and only one SUPPLIER

EACH STORE may have zero one or many SUPPLIER_STOREs

EACH SUPPLIER_STORE must be for one and only STORE

EACH STORE MAY have zero one or many STORE_EMPLOYEEs

EACH STORE_EMPLOYEE must be associated with one and only one STORE

EACH EMPLOYEE may work on zero one or many STORE_EMPLOYEES

EACH STORE_EMPLOYEEs must be associated with one and only one EMPLOYEE

EACH CUSTOMER may have zero, one or many PAYMENTs

EACH PAYMENT must be for one and only one CUSTOMER

EACH CUSTOMER may have zero one or many ITEMs

EACH ITEM must be for one and only one CUSTOMER

EACH PRODUCT may have zero one or many ITEMs

EACH ITEM must be associated with one and only one PRODUCT

EACH PRODUCT may have zero one or many ORDERs

EACH ORDER must be for one and only one PRODUCT

EACH PRODUCT may have zero one or many STORE_PRODUCTs

EACH STORE_PRODUCTS must be for one and only PRODUCT

EACH STORE may stock zero one or many STORE_PRODUCTS

EACH STORE-PRODUCTS must stocked at one and only one STORE

EACH PRODUCT may have zero one or many ORDER_ITEMs

EACH ORDER_ITEM must be for one and only PRODUCT

EACH ORDER may be for zero one or many ORDER_ITEMs

EACH ORDER_ITEM must be for one and only one ORDER

EACH PRODUCT may have zero one or many PAYMENT

EACH PAYMENT must be for one and only one PRODUCT

5. RELATIONAL MODEL(H)

Customer				
Key Type	Optionality	Column Name	Data type	Length
PK	*	Customer_id	NUMERIC	2,0
	*	First_name	VARCHAR	255
	*	Last_name	VARCHAR	255
	*	Address	VARCHAR	255
UK	o	Email	VARCHAR	255

Supplier				
Key types	Optionality	Column Name	Data type	Length
PK	*	Supplier_id	NUMERIC	2,0
	*	First_name	VARCHAR	255
	*	Last_name	VARCHAR	255
UK	*	Email	VARCHAR	255

Employee				
Key types	Optionality	Column Name	Data type	Length
PK	*	Employee_id	NUMERIC	2,0
	*	First_name	VARCHAR	255
	*	Last_name	VARCHAR	255
UK	o	Email	VARCHAR	255
UK	*	Sin_no	VARCHAR	12
UK	*	Phone_no	VARCHAR	15
	*	Salary	DECIMAL	10,2

Product				
Key types	Optionality	Column Name	Data type	Length
PK	*	Product_id	NUMERIC	4,0
	*	P_name	VARCHAR	255
	*	P_description	VARCHAR	255
	o	Price	DECIMAL	10,2

Store				
Key types	Optionality	Column Name	Data type	Length
PK	*	Store_id	NUMERIC	3,0
	*	Store_name	VARCHAR	255
	*	Store_address	VARCHAR	255
	*	Store_location	VARCHAR	255

Store_product				
Key types	Optionality	Column Name	Data type	Length
PK, FK	*	Store_id	NUMERIC	3,0
PK, FK	*	Product_id	NUMERIC	4,0
	*	Product_quantity	DECIMAL	5,0

Orders				
Key types	Optionality	Column Name	Data type	Length
PK	*	Order_id	NUMERIC	4,0
	*	Quantity_order	NUMERIC	5,0
	o	Order_date	DATE	
FK	*	Store_id	NUMERIC	3,0

Supplier_store				
Key types	Optionality	Column Name	Data type	Length
PK, FK	*	Store_id	NUMERIC	3,0
PK, FK	*	Supplier_id	NUMERIC	2,0
	*	Supply_date	DATE	
	*	Quantity_supplied	NUMERIC	5,0

Store_employee				
Key types	Optionality	Column Name	Data type	Length
PK, FK	*	Store_id	NUMERIC	3,0
PK, FK	*	Employee_id	NUMERIC	2,0
	*	Work_hours	NUMERIC	2,0
	*	Wage_rate	DECIMAL	12,2

Item				
Key types	Optionality	Column Name	Data type	Length
PK, FK	*	Customer_id	NUMERIC	2,0
PK, FK	*	Product_id	NUMERIC	4,0
	o	Price	DECIMAL	10,2
	o	discount	DECIMAL	5,2

Order_item				
Key types	Optionality	Column Name	Data type	Length
PK, FK	*	Product_id	NUMERIC	4,0
PK, FK	*	Order_id	NUMERIC	4,0
	*	Quantity	NUMERIC	5,0

Payment				
Key types	Optionality	Column Name	Data type	Length
PK	*	Payment_id	NUMERIC	5,0
	o	Payment_date	DATE	
	o	Payment_method	VARCHAR	50
FK1	*	Customer_id	NUMERIC	2,0
FK2	*	Product_id	NUMERIC	4,0

6. BUSINESS RULES(H)

Table Name	Column Name	Constraint
Customer	Customer_id	The customer_id serves as the primary key, automatically generated as an identity column. first_name and last_name cannot be null. email must be unique. There are no default values or foreign keys associated with the Customer entity.
	First_name	
	Last_name	
	Customer_address	
	Email	
Supplier	Supplier_id	supplier_id is the primary key. first_name and last_name cannot be null. email must be unique. No default values or foreign keys are specified.
	First_name	
	Last_name	
	Email	
Employee	Employee_id	The employee_id is the primary key and must be unique. first_name, last_name, and email cannot be null. SIN_NO serves as a unique key. phone must be unique. salary has no constraints.
	First_name	
	Last_name	
	Email	
	SIN_no	
	Phone_no	
	Salary	
Product	Product_id	product_id is the primary key. product_name and product_description cannot be null. price has no constraints.
	Product_name	
	Product_description	
	price	
Store	Store_id	store_id is the primary key. store_name, store_address, and store_location cannot be null. No default values or foreign keys are specified.
	Store_name	
	Store_address	
	Store_location	
Store_product	Store_id	Composite primary key consisting of store_id and product_id. product_quantity has no constraints. store_id and product_id are foreign keys referencing Store and Product tables, respectively.
	Product_id	
	Product_quantity	
Orders	Order_id	order_id serves as the primary key, automatically generated as a sequence. quantity_order cannot be null. order_date cannot be null. store_id is a foreign key referencing the Store table.
	Quantity_order	
	Order_date	
	Store_id	
Supplier_store	Store_id	Composite primary key consisting of store_id and supplier_id. supply_date and quantity_supplied have no constraints. store_id and
	Supplier_id	
	Supply_date	
	Quantity_supplied	

		supplier_id are foreign keys referencing Store and Supplier tables, respectively.
Store_employee	Store_id	Composite primary key consisting of store_id and employee_id. work_hours has a default value of 0. wage_rate has no constraints. store_id and employee_id are foreign keys referencing Store and Employee tables, respectively.
	Employee_id	
	Work_hours	
	Wage_rate	
Item	Customer_id	Composite primary key consisting of customer_id and product_id. price has no constraints. discount has a default value of 0. customer_id and product_id are foreign keys referencing Customer and Product tables, respectively.
	Product_id	
	Price	
	Discount	
Order_item	Product_id	Composite primary key consisting of product_id and order_id. quantity has no constraints. product_id and order_id are foreign keys referencing Product and Orders tables, respectively.
	Order_id	
	Quantity	
Payment	Payment_id	payment_id serves as the primary key. payment_date cannot be null. payment_method, customer_id, and product_id have no constraints.
	Payment_date	
	Payment_method	
	Customer_id	
	Product_id	

6. CREATE, ALTER, AND INSERT CODES(H)

```
-- Create Customer table
CREATE TABLE Customer (
    customer_id NUMERIC(2,0) NOT NULL identity(1,1),
    first_name VARCHAR(255) NOT NULL,
    last_name VARCHAR(255) NOT NULL,
    Customer_address VARCHAR(255),
    email VARCHAR(255),
    CONSTRAINT customer_pk
        PRIMARY KEY(customer_id)
);

-- Add unique constraint on email column
ALTER TABLE Customer
ADD CONSTRAINT Customer_uk UNIQUE(email);

-- Insert  data into Customer table
INSERT INTO Customer (first_name, last_name, Customer_address, email)
VALUES
('John', 'Doe', '123 Main St', 'john@gmail.com'),
('Jane', 'Smith', '456 george St', 'jane@gmail.com'),
('Alice', 'Johnson', '789 Elm St', 'alice@gmail.com'),
('Bob', 'Brown', '101 Maple St', 'bob@gmail.com'),
('Emily', 'Davis', '222 Pine St', 'emily@gmail.com');

-- create table supplier
CREATE TABLE Supplier (
    supplier_id NUMERIC(2,0) NOT NULL,
    first_name VARCHAR(255) NOT NULL,
    last_name VARCHAR(255) NOT NULL,
    email VARCHAR(255),
        CONSTRAINT Supplier_pk
```

```
        PRIMARY KEY(supplier_id)
);

-- ADD unique constraint on email column
ALTER TABLE Supplier
ADD CONSTRAINT Supplier_uk UNIQUE (email);

-- ADD a CHECK constraint for email column
Alter table Supplier
ADD CONSTRAINT supplier_check_email
check(email like '%@%.%');

-- Inserting test data for suppliers
INSERT INTO Supplier (supplier_id, first_name, last_name, email)
VALUES
   (88, 'Ram', 'Prasad', 'ram.prasad@gmail.com.com'),
   (89, 'Shyam', 'Gurung', 'shyam.gurung@gmail.com'),
   (90, 'Gopal', 'Thapa', 'gopal.thapa@gmail.com'),
   (91, 'Sita', 'Shrestha', 'sita.shrestha@gmail.com'),
   (92, 'Mohan', 'Adhikari', 'mohan.adhikari@gmail.com');

-- create table Employee
CREATE TABLE Employee (
   employee_id NUMERIC(2,0) NOT NULL,
   first_name VARCHAR(255) NOT NULL,
   last_name VARCHAR(255) NOT NULL,
   email VARCHAR(255) NOT NULL ,
   SIN_NO VARCHAR(12) NOT NULL ,
   phone VARCHAR(15),
        salary DECIMAL(10,2),
        CONSTRAINT Employee_pk
        PRIMARY KEY(employee_id)
```

```
);

-- ADD unique constraint on email column
ALTER TABLE Employee
ADD CONSTRAINT Employee_uk_email UNIQUE(email);

-- ADD unique constraint on SIN_NO column
ALTER TABLE Employee
ADD CONSTRAINT Employee_uk_sin UNIQUE(SIN_NO);

-- ADD unique constraint to phone column
ALTER TABLE Employee
ADD CONSTRAINT Employee_uk_phone UNIQUE(PHONE);

ALTER table Employee
add constraint check_salary1 check(salary>=20000.00 and salary<=70000.99);

-- Inserting test data for employees
INSERT INTO Employee (employee_id, first_name, last_name, email, SIN_NO, phone,salary)
VALUES
    (6, 'Aarav', 'Sharma', 'aarav.sharma.com', '123-456-789', '123-456-7890',20000.00),
    (7, 'Aarohi', 'Thapa', 'aarohi.thapa@example.com', '987-654-321', '987-654-3210',30000.78),
    (8, 'Abhishek', 'Bhattarai', 'abhishek.bhattarai@example.com', '456-789-123', '456-789-
1230',45000.99),
    (9, 'Alisha', 'Pokharel', 'alisha.pokharel@example.com', '789-123-456', '789-123-4560',45000.00),
    (10, 'Anjali', 'Joshi', 'anjali.joshi@example.com', '654-321-987', '654-321-9870',70000.45);

-- create table product
CREATE TABLE Product (
    product_id NUMERIC (4,0) NOT NULL,
    product_name  VARCHAR(255) NOT NULL,
    product_description varchar(255) NOT NULL,
    price DECIMAL(10, 2),
```

```
        CONSTRAINT product_pk
        primary key (product_id)
);

-- INSERTING data for product
INSERT INTO Product (product_id, product_name, product_description, price)
VALUES
   (111, 'Apples', 'Fresh apples from local orchards', 2.99),
   (112, 'Bananas', 'Ripe bananas, perfect for a quick snack', 1.49),
   (113, 'Milk', 'Organic whole milk, 1 gallon', 3.99),
   (114, 'Eggs', 'Farm-fresh large brown eggs, dozen', 2.79),
   (115, 'Bread', 'Whole wheat bread, sliced, 24 oz', 2.49),
   (116, 'Chicken Breast', 'Boneless skinless chicken breast, 1 lb', 5.99),
   (117, 'Rice', 'Long grain white rice, 5 lb bag', 4.99),
   (118, 'Pasta', 'Spaghetti pasta, 16 oz box', 1.99),
   (119, 'Tomatoes', 'Vine-ripened tomatoes, pack of 6', 3.49),
   (120, 'Spinach', 'Fresh organic spinach, 8 oz bag', 2.99);

-- create table store
CREATE TABLE Store (
   store_id NUMERIC(3,0) NOT NULL,
   store_name VARCHAR(255) NOT NULL,
   store_address VARCHAR(255) NOT NULL,
   store_location VARCHAR(255) NOT NULL,
        CONSTRAINT store_pk
        PRIMARY KEY (store_id)
);

-- Inserting test data for stores
INSERT INTO Store (store_id, store_name, store_address, store_location)
VALUES
   (130, 'Main Street Grocery', '123 Main Street', 'City Center'),
```

```
    (140, 'Hilltop Market', '456 Hilltop Avenue', 'Suburban Area'),
    (150, 'Downtown Deli', '789 Elm Street', 'Downtown'),
    (160, 'Corner Convenience', '10 Oak Avenue', 'Urban Area');

-- create table store_product
CREATE TABLE Store_product (
    store_id NUMERIC(3,0) NOT NULL,
    product_id NUMERIC(4,0) NOT NULL,
    product_quantity DECIMAL(5,0) ,
            CONSTRAINT Store_product_pk
    PRIMARY KEY (store_id, product_id)
);

-- Add a foreign key constraint on the store_id column in the STORE_product table,
-- referencing the store_id column in the Store table.
ALTER TABLE STORE_product
ADD CONSTRAINT Store_product_fk1
FOREIGN KEY (store_id) REFERENCES Store(store_id);

-- Add a foreign key constraint on the product_id column in the Store_product table,
-- referencing the product_id column in the Product table.
ALTER TABLE Store_product
ADD CONSTRAINT Store_product_fk2
FOREIGN KEY (product_id) REFERENCES Product(product_id);

-- Inserting data for Store_product table
INSERT INTO Store_product (store_id, product_id, product_quantity)
VALUES
    (130, 113, 20),
    (140, 114, 40),
    (150, 112, 25);
```

```
-- create table Orders
CREATE TABLE Orders (
    order_id NUMERIC(4,0) NOT NULL identity(50,2),
    quantity_order NUMERIC(5,0) NOT NULL,
    order_date DATE NOT NULL,
    store_id NUMERIC(3,0),
            CONSTRAINT orders_pk
            PRIMARY KEY(order_id)
);

-- Add a foreign key constraint on the store_id column in the Orders table,
-- referencing the store_id column in the Store table.
ALTER TABLE Orders
ADD CONSTRAINT Orders_fk
FOREIGN KEY (store_id) REFERENCES Store(store_id);

-- Inserting test data for orders
INSERT INTO Orders ( quantity_order, order_date, store_id)
VALUES
    ( 2, '2024-04-01', 130),
    ( 3, '2024-04-02', 140),
    ( 2, '2024-04-03', 130),
    ( 2, '2024-04-04', 140),
    ( 4, '2024-04-05', 130),
    ( 3, '2024-04-06', 160),
    ( 2, '2024-04-07', 130),
    ( 3, '2024-04-08', 140),
    ( 4, '2024-04-09', 160),
    ( 2, '2024-04-10', 160);

-- create table supplier_store
CREATE TABLE Supplier_store (
```

```
    store_id NUMERIC(3,0) NOT NULL,
    supplier_id NUMERIC(2,0) NOT NULL,
    supply_date DATE,
    quantity_supplied NUMERIC(5,0),
        CONSTRAINT Supplier_store_pk
    PRIMARY KEY (store_id, supplier_id)
);

-- Add a foreign key constraint on the store_id column in the Supplier_store table,
-- referencing the store_id column in the Store table.
ALTER TABLE Supplier_store
ADD CONSTRAINT Supplier_store_fk1
FOREIGN KEY (store_id) REFERENCES Store(store_id);

-- Add a foreign key constraint on the supplier_id column in the Supplier_store table,
-- referencing the supplier_id column in the Supplier table.
ALTER TABLE Supplier_store
ADD CONSTRAINT Supplier_store_fk2
FOREIGN KEY (supplier_id) REFERENCES Supplier(supplier_id);

-- Inserting test data for Supplier_store
INSERT INTO Supplier_store (store_id, supplier_id, supply_date, quantity_supplied)
VALUES
    (130, 88, '2024-03-01', 100),
    (150, 89, '2024-03-02', 150),
    (160, 90, '2024-03-03', 200),
    (140, 90, '2024-03-04', 250),
    (130, 91, '2024-03-05', 300);

-- create table store_employee
CREATE TABLE Store_employee (
    store_id NUMERIC(3,0) NOT NULL,
```

```
    employee_id NUMERIC(2,0) NOT NULL,
    work_hours NUMERIC (2,0) DEFAULT 0,
    wage_rate DECIMAL(12, 2),
        CONSTRAINT Store_employee_pk
    PRIMARY KEY (store_id, employee_id)
);

-- Add a foreign key constraint named Store_employee_fk1 on the store_id column in the Store_employee table,
-- referencing the store_id column in the Store table.
ALTER TABLE Store_employee
ADD CONSTRAINT Store_employee_fk1
FOREIGN KEY (store_id) REFERENCES Store(store_id);

-- Add a foreign key constraint named Store_employee_fk2 on the employee_id column in the Store_employee table,
-- referencing the employee_id column in the Employee table.
ALTER TABLE Store_employee
ADD CONSTRAINT Store_employee_fk2
FOREIGN KEY (employee_id) REFERENCES Employee(employee_id);

-- Inserting test data for store employees
INSERT INTO Store_employee (store_id, employee_id, work_hours, wage_rate)
VALUES
    (130, 6, 4, 15.00),
    (140, 7, 3, 14.50),
    (150, 8, 3, 16.00);

-- create table Item
CREATE TABLE Item (
    customer_id NUMERIC(2,0) NOT NULL,
    product_id NUMERIC(4,0) NOT NULL,
    price DECIMAL(10, 2),
```

```
    discount DECIMAL(5, 2) DEFAULT 0,
        CONSTRAINT ITEM_PK
    PRIMARY KEY (customer_id, product_id)
);

-- Add a foreign key constraint named Item_fk1 on the customer_id column in the Item table,
-- referencing the customer_id column in the Customer table.
ALTER TABLE Item
ADD CONSTRAINT Item_fk1
FOREIGN KEY (customer_id) REFERENCES Customer(customer_id);

-- Add a foreign key constraint named Item_fk2 on the product_id column in the Item table,
-- referencing the product_id column in the Product table.
ALTER TABLE Item
ADD CONSTRAINT Item_fk2
FOREIGN KEY (product_id) REFERENCES Product(product_id);

-- Inserting test data for items
INSERT INTO Item (customer_id, product_id, price, discount)
VALUES
    (1, 111, 2.99, 0),
    (1, 113, 3.99, 0.5),
    (2, 112, 1.49, 0),
    (2, 114, 2.79, 0.2),
    (3, 115, 2.49, 0),
    (3, 117, 4.99, 0),
    (4, 113, 3.99, 0),
    (4, 118, 1.99, 0.1);
INSERT INTO Item (customer_id, product_id, price)
VALUES
    (5, 111, 2.99),
    (5, 116, 5.99);
```

```
-- create table Order_item
CREATE TABLE Order_item (
    product_id NUMERIC(4,0) NOT NULL,
    order_id NUMERIC(4,0) NOT NULL,
    quantity NUMERIC(5,0),
    PRIMARY KEY (product_id, order_id)
);

-- Add a foreign key constraint named Order_item_fk1 on the product_id column in the Order_item table,
-- referencing the product_id column in the Product table.
ALTER TABLE Order_item
ADD CONSTRAINT Order_item_fk1
FOREIGN KEY (product_id) REFERENCES Product(product_id);

-- Add a foreign key constraint named Order_item_fk2 on the order_id column in the Order_item table,
-- referencing the order_id column in the Orders table.
ALTER TABLE Order_item
ADD CONSTRAINT ORDER_item_fk2
FOREIGN KEY (order_id) REFERENCES Orders(order_id);

-- Inserting data into Order_item table
INSERT INTO Order_item (product_id, order_id, quantity)
VALUES
    (111, 50, 2),
    (113, 50, 1),
    (112, 52, 2),
    (114, 52, 1),
    (115, 54, 3),
    (116, 54, 2),
    (111, 56, 2),
```

```
    (113, 56, 1),
    (117, 58, 4),
    (118, 58, 3);

-- create table Payment
CREATE TABLE Payment (
    payment_id NUMERIC(5,0) NOT NULL,
    payment_date DATE NOT NULL,
    payment_method VARCHAR(50),
    customer_id NUMERIC(2,0),
    product_id NUMERIC(4,0),
        CONSTRAINT Payment_pk
        PRIMARY KEY(Payment_id)
);

-- Add a foreign key constraint named Payment_fk1 on the customer_id column in the Payment table,
-- referencing the customer_id column in the Customer table.
ALTER TABLE Payment
ADD CONSTRAINT Payment_fk1
FOREIGN KEY (customer_id) REFERENCES Customer(customer_id);

-- Add a foreign key constraint named Payment_fk2 on the product_id column in the Payment table,
-- referencing the product_id column in the Product table.
ALTER TABLE Payment
ADD CONSTRAINT Payment_fk2
FOREIGN KEY (product_id) REFERENCES Product(product_id);

-- ADD a check constraint for payment_method column-
ALTER TABLE Payment
ADD CONSTRAINT Payment_check_paymnet_method check( payment_method In( 'Debit
card','paypal','Cash'));

-- Inserting test data for payments
```

```
INSERT INTO Payment (payment_id, payment_date, payment_method, customer_id, product_id)
VALUES
   (3003, '2024-04-02', 'Cash', 3, 113),
   (1001, '2024-04-03', 'Debit card', 1, 111),
   (2002, '2024-04-03', 'PayPal', 2, 112),
   (4004, '2024-04-01', 'Debit card', 4, 114),
   (5005, '2024-03-31', 'Debit Card', 5,115),
   (6006, '2024-03-30', 'Cash', 1, 116),
   (7007, '2024-03-29', 'Debit Card', 2, 117),
   (8008, '2024-03-28', 'PayPal', 3, 118),
   (9009, '2024-03-27', 'Cash', 4, 119),
   (1010, '2024-03-26', 'Debit Card', 5, 120);


--- checking constraint NOT null---
--- The NOT NULL constraint ensure that a column cannot contain NULL values ---
--- when entering NULL value in column first_name which is NOT NULL, it showed error.
INSERT INTO Customer (first_name, last_name, Customer_address, email)
VALUES(NULL,'Raut','89 main st','Raut@gmail.com');


--- DEFAULT constraint checking ---
--- when no value entered to the work_hour column in store_employee tablee ,the default value 0 will
be assigned ---
INSERT INTO Store_employee (store_id, employee_id,  wage_rate)
VALUES(140,8,15.77);


INSERT INTO Store_employee (store_id, employee_id,  wage_rate)
VALUES
   (160, 9,  15.75),
   (150, 10,  17.25);


SELECT * FROM Store_employee;


--- PRIMARY KEY constraint checking---
```

```
--- when same employee_id was inserted again in Employee table, it showed error.
INSERT INTO Employee (employee_id, first_name, last_name, email, SIN_NO, phone,salary)
VALUES
    (6, 'Aarav', 'Sharma', 'aarav.sharma.com', '123-456-789', '123-456-7890',20000.00);

-- PRIMARY KEY constraint checking
-- when same supplier_id was inserted again in supplier table ,it showed error
INSERT INTO Supplier (supplier_id, first_name, last_name, email)
VALUES
    (88, 'Ramu', 'Pad', 'raaam.prasad@gmail.com');

-- UNIQUE KEY constraint checking
--- When same email address was inserted in email column of Employee table  ,it showed error---
INSERT INTO Employee (employee_id, first_name, last_name, email, SIN_NO, phone,salary)
VALUES(11, 'Aaav', 'Sarma', 'aarav.sharma.com', '168-456-789', '199-456-7890',23001.00);

--- Foreigh key constraint checking---
--- When i enter a foreign key that does not exists in reference table it will so error
--- IN payment table i enter a product_id that doesnot exists ,it showed error
INSERT INTO Payment (payment_id, payment_date, payment_method, customer_id, product_id)
VALUES
    (1011, '2024-04-03', 'Credit Card', 1, 121);

--- Foreigh key constraint checking---
--- IN item table i enter a product_id that doesnot exists, it showed error
INSERT INTO Item (customer_id, product_id, price, discount)
VALUES  (1, 122, 2.99, 0);

--- Foreigh key constraint checking---
--- IN orders table i enter a store_id that doesnot exists ,it showed error
INSERT INTO Orders ( quantity_order, order_date, store_id)
VALUES
```

(2, '2024-04-01', 133);

--- checking constraint CHECK---

--- when i try to enter the salary in the Emplpoyee table which was more or less than the specified value ,it showed error

INSERT INTO Employee (employee_id, first_name, last_name, email, SIN_NO, phone,salary)

VALUES

(12, 'Aarav', 'Sharma', 'aav.sharma.com', '129-456-789', '113-456-7890',200000.00);

--- Checking constraint check--

-- when i enter a eamil without @ or . in email column of supplier table ,showed error.

INSERT INTO Supplier (supplier_id, first_name, last_name, email)

VALUES (2, 'Jane', 'Smith', 'jane@gmailcom');

INSERT INTO Supplier (supplier_id, first_name, last_name, email)

VALUES (2, 'Jane', 'smittth', 'janegmail.com');

--- checking constraint check

-- when i enter a payment_method other than the specified one ,it showed error

INSERT INTO Payment (payment_id, payment_date, payment_method, customer_id, product_id)

VALUES

(3113, '2024-04-02', 'Credit card', 3, 113);