**INF 305 DATABASE MANAGEMENT SYSTEM 2**

**ENDTERM EXAM**

**Introduction**

My project is related to a dishes shop. The dishes shop system is an e-commerce platform that allows customers to browse and purchase a variety of dishes online. The system is designed to provide a seamless shopping experience for customers, from product selection to payment and delivery. The system features several key entities such as customers, orders, products, categories, order items, payments, and deliveries. Customers can browse products by category and add items to their cart, and then proceed to checkout and make payments using various payment methods. The system also tracks order status and delivery information, allowing customers to track their orders and receive timely updates on their delivery status. To ensure data consistency and accuracy, the system follows the principles of 1NF, 2NF, and 3NF. Additionally, the system includes several procedures, functions, triggers, and exceptions to ensure data integrity and enhance system functionality. Overall, the dishes shop system is a comprehensive e-commerce platform that offers a convenient and secure way for customers to purchase their favorite dishes online.

Kalykova Saule worked on the project

**ERD**



TABLES

**// Creating the Customer table**

CREATE TABLE Customer (

customer\_id NUMBER(10) NOT NULL,

name VARCHAR2(100) NOT NULL,

email VARCHAR2(100) NOT NULL,

phone\_number VARCHAR2(20) NOT NULL,

address VARCHAR2(200) NOT NULL,

PRIMARY KEY (customer\_id)

);

**// Creating the Order table**

CREATE TABLE Order\_1 (

order\_id NUMBER(10) NOT NULL,

order\_number VARCHAR2(20) NOT NULL,

order\_date DATE NOT NULL,

customer\_id NUMBER(10) NOT NULL,

payment\_id NUMBER(10) NOT NULL,

delivery\_id NUMBER(10) NOT NULL,

PRIMARY KEY (order\_id),

FOREIGN KEY (customer\_id) REFERENCES Customer (customer\_id),

FOREIGN KEY (payment\_id) REFERENCES Payment (payment\_id),

FOREIGN KEY (delivery\_id) REFERENCES Delivery (delivery\_id)

);

**// Creating the Product table**

CREATE TABLE Product (

product\_id NUMBER(10) NOT NULL,

product\_name VARCHAR2(100) NOT NULL,

description VARCHAR2(1000) NOT NULL,

price NUMBER(10, 2) NOT NULL,

category\_id NUMBER(10) NOT NULL,

PRIMARY KEY (product\_id),

FOREIGN KEY (category\_id) REFERENCES Category (category\_id)

);

**// Creating the Category table**

CREATE TABLE Category (

category\_id NUMBER(10) NOT NULL,

category\_name VARCHAR2(100) NOT NULL,

description VARCHAR2(1000) NOT NULL,

PRIMARY KEY (category\_id)

);

**// Creating the Order Item table**

CREATE TABLE Order\_Item (

order\_item\_id NUMBER(10) NOT NULL,

order\_id NUMBER(10) NOT NULL,

product\_id NUMBER(10) NOT NULL,

quantity NUMBER(10) NOT NULL,

PRIMARY KEY (order\_item\_id),

FOREIGN KEY (order\_id) REFERENCES Order (order\_id),

FOREIGN KEY (product\_id) REFERENCES Product (product\_id)

);

**// Creating the Payment table**

CREATE TABLE Payment (

payment\_id NUMBER(10) NOT NULL,

payment\_type VARCHAR2(100) NOT NULL,

payment\_date DATE NOT NULL,

amount NUMBER(10, 2) NOT NULL,

PRIMARY KEY (payment\_id)

);

**// Creating the Delivery table**

CREATE TABLE Delivery (

delivery\_id NUMBER(10) NOT NULL,

delivery\_date DATE NOT NULL,

delivery\_address VARCHAR2(200) NOT NULL,

status VARCHAR2(100) NOT NULL,

PRIMARY KEY (delivery\_id)

);

DATA

**//Inserting data into Customer table**

INSERT INTO Customer (customer\_id, name, email, phone\_number, address)

VALUES (1, 'John Smith', 'john.smith@example.com', '+1-555-123-4567', '123 Main St'),

(2, 'Jane Doe', 'jane.doe@example.com', '+1-555-987-6543', '456 Elm St'),

(3, 'Bob Johnson', 'bob.johnson@example.com', '+1-555-555-5555', '789 Oak Ave'),

(4, 'Alice Green', 'alice.green@example.com', '+1-555-111-1111', '321 Maple St'),

(5, 'Sam Lee', 'sam.lee@example.com', '+1-555-222-2222', '555 Pine St'),

(6, 'Sarah Brown', 'sarah.brown@example.com', '+1-555-333-3333', '777 Cedar Ave'),

(7, 'Mike Jones', 'mike.jones@example.com', '+1-555-444-4444', '888 Walnut St'),

(8, 'Lisa Taylor', 'lisa.taylor@example.com', '+1-555-777-7777', '999 Birch Ln'),

(9, 'David Kim', 'david.kim@example.com', '+1-555-888-8888', '111 Oak Ln'),

(10, 'Mary Williams', 'mary.williams@example.com', '+1-555-999-9999', '222 Maple Ave');

**//Inserting data into Payment table**

INSERT INTO Payment (payment\_id, payment\_type, payment\_date, amount)

VALUES (1, 'Credit card', TO\_DATE('2022-03-01', 'YYYY-MM-DD'), 100.00),

(2, 'PayPal', TO\_DATE('2022-03-02', 'YYYY-MM-DD'), 50.00),

(3, 'Debit card', TO\_DATE('2022-03-03', 'YYYY-MM-DD'), 75.00),

(4, 'Cash', TO\_DATE('2022-03-04', 'YYYY-MM-DD'), 30.00),

(5, 'Credit card', TO\_DATE('2022-03-05', 'YYYY-MM-DD'), 200.00),

(6, 'PayPal', TO\_DATE('2022-03-06', 'YYYY-MM-DD'), 90.00),

(7, 'Debit card', TO\_DATE('2022-03-07', 'YYYY-MM-DD'), 150.00),

(8, 'Cash', TO\_DATE('2022-03-08', 'YYYY-MM-DD'), 75.00),

(9, 'Credit card', TO\_DATE('2022-03-09', 'YYYY-MM-DD'), 50.00),

(10, 'PayPal', TO\_DATE('2022-03-10', 'YYYY-MM-DD'), 25.00);

**//Inserting data into Delivery table**

INSERT INTO Delivery (delivery\_id, delivery\_date, delivery\_address, status)

VALUES (1, TO\_DATE('2022-01-01', 'YYYY-MM-DD'), '123 Main St, Anytown, USA', 'Shipped');

INSERT INTO Delivery (delivery\_id, delivery\_date, delivery\_address, status)

VALUES (2, TO\_DATE('2022-01-02', 'YYYY-MM-DD'), '456 Oak St, Anytown, USA', 'Delivered');

INSERT INTO Delivery (delivery\_id, delivery\_date, delivery\_address, status)

VALUES (3, TO\_DATE('2022-01-03', 'YYYY-MM-DD'), '789 Pine St, Anytown, USA', 'In Transit');

INSERT INTO Delivery (delivery\_id, delivery\_date, delivery\_address, status)

VALUES (4, TO\_DATE('2022-01-04', 'YYYY-MM-DD'), '321 Elm St, Anytown, USA', 'Shipped');

INSERT INTO Delivery (delivery\_id, delivery\_date, delivery\_address, status)

VALUES (5, TO\_DATE('2022-01-05', 'YYYY-MM-DD'), '654 Maple St, Anytown, USA', 'Delivered');

INSERT INTO Delivery (delivery\_id, delivery\_date, delivery\_address, status)

VALUES (6, TO\_DATE('2022-01-06', 'YYYY-MM-DD'), '987 Cedar St, Anytown, USA', 'In Transit');

INSERT INTO Delivery (delivery\_id, delivery\_date, delivery\_address, status)

VALUES (7, TO\_DATE('2022-01-07', 'YYYY-MM-DD'), '246 Birch St, Anytown, USA', 'Shipped');

INSERT INTO Delivery (delivery\_id, delivery\_date, delivery\_address, status)

VALUES (8, TO\_DATE('2022-01-08', 'YYYY-MM-DD'), '135 Fir St, Anytown, USA', 'Delivered');

INSERT INTO Delivery (delivery\_id, delivery\_date, delivery\_address, status)

VALUES (9, TO\_DATE('2022-01-09', 'YYYY-MM-DD'), '864 Poplar St, Anytown, USA', 'In Transit');

INSERT INTO Delivery (delivery\_id, delivery\_date, delivery\_address, status)

VALUES (10, TO\_DATE('2022-01-10', 'YYYY-MM-DD'), '975 Oak St, Anytown, USA', 'Shipped');

**//Inserting data into Category table**

INSERT INTO Category (category\_id, category\_name, description)

VALUES (1, 'Main Dish', 'Main course dishes'),

(2, 'Appetizers', 'Small dishes to start a meal'),

(3, 'Desserts', 'Sweet dishes to finish a meal'),

(4, 'Drinks', 'Beverages to accompany the meal'),

(5, 'Specials', 'Special dishes for the season or occasion'),

(6, 'Vegetarian', 'Vegetarian dishes'),

(7, 'Gluten-free', 'Gluten-free dishes'),

(8, 'Seafood', 'Seafood dishes'),

(9, 'Meat', 'Meat dishes'),

(10, 'Pasta', 'Pasta dishes'),

(11, 'Pizza', 'Pizza'),

(12, 'Sandwiches', 'Sandwiches'),

(13, 'Salads', 'Salads'),

(14, 'Soups', 'Soups'),

(15, 'Sides', 'Side dishes');

**//Inserting data into Product table**

INSERT INTO Product (product\_id, product\_name, description, price, category\_id)

VALUES (1, 'Spaghetti Bolognese', 'Classic Italian pasta dish with tomato sauce and minced meat', 12.99, 10),

(2, 'Margherita Pizza', 'Traditional Italian pizza with tomato sauce and mozzarella cheese', 10.99, 11),

(3, 'Greek Salad', 'Salad with feta cheese, olives, cucumber, and tomato', 8.99, 13),

(4, 'Tiramisu', 'Classic Italian dessert with layers of coffee-soaked ladyfingers and mascarpone cream', 6.99, 3),

(5, 'Caesar Salad', 'Salad with romaine lettuce, croutons, parmesan cheese, and Caesar dressing', 7.99, 13),

(6, 'Fish and Chips', 'British pub food with battered fish and fries', 14.99, 8),

(7, 'Mushroom Risotto', 'Italian rice dish with mushrooms and Parmesan cheese', 11.99, 10),

(8, 'Beef Burger', 'Classic burger with beef patty, lettuce, tomato, and onion', 9.99, 15),

(9, 'Mango Lassi', 'Indian yogurt-based drink with mango pulp and spices', 3.99, 4),

(10, 'Falafel Wrap', 'Middle Eastern wrap with falafel balls, lettuce, tomato, and tahini sauce', 8.99, 2);

**//Inserting data into Order table**

INSERT INTO Order\_1 (order\_id, customer\_id, delivery\_id, order\_date, payment\_id, order\_number)

VALUES (1, 1, 1, TO\_DATE('2023-04-24', 'YYYY-MM-DD'), 1, 'ORD-001'),

(2, 2, 2, TO\_DATE('2023-04-23', 'YYYY-MM-DD'), 2, 'ORD-002'),

(3, 3, 3, TO\_DATE('2023-04-22', 'YYYY-MM-DD'), 3, 'ORD-003'),

(4, 4, 4, TO\_DATE('2023-04-21', 'YYYY-MM-DD'), 4, 'ORD-004'),

(5, 5, 5, TO\_DATE('2023-04-20', 'YYYY-MM-DD'), 5, 'ORD-005'),

(6, 6, 6, TO\_DATE('2023-04-19', 'YYYY-MM-DD'), 6, 'ORD-006'),

(7, 7, 7, TO\_DATE('2023-04-18', 'YYYY-MM-DD'), 7, 'ORD-007'),

(8, 8, 8, TO\_DATE('2023-04-17', 'YYYY-MM-DD'), 8, 'ORD-008'),

(9, 9, 9, TO\_DATE('2023-04-16', 'YYYY-MM-DD'), 9, 'ORD-009'),

(10, 10, 10, TO\_DATE('2023-04-15', 'YYYY-MM-DD'), 10, 'ORD-010;

**//Inserting data into Order\_Item table**

INSERT INTO Order\_Item (order\_item\_id, order\_id, product\_id, quantity)

VALUES (1, 1, 1, 2),

(2, 1, 3, 1),

(3, 2, 2, 1),

(4, 2, 4, 2),

(5, 3, 5, 1),

(6, 3, 6, 1),

(7, 4, 7, 1),

(8, 4, 8, 1),

(9, 5, 9, 1),

(10, 5, 10, 2);

**Normal Forms**

**Customer:**

1NF: All attributes are atomic and do not contain repeating groups.

2NF: No partial dependencies exist, and the entity is fully dependent on the primary key.

3NF: No transitive dependencies exist.

**Order:**

1NF: All attributes are atomic and do not contain repeating groups.

2NF: No partial dependencies exist, and the entity is fully dependent on the primary key.

3NF: No transitive dependencies exist.

**Product:**

1NF: All attributes are atomic and do not contain repeating groups.

2NF: No partial dependencies exist, and the entity is fully dependent on the primary key.

3NF: No transitive dependencies exist.

**Category:**

1NF: All attributes are atomic and do not contain repeating groups.

2NF: No partial dependencies exist, and the entity is fully dependent on the primary key.

3NF: No transitive dependencies exist.

**Order Item:**

1NF: All attributes are atomic and do not contain repeating groups.

2NF: No partial dependencies exist, and the entity is fully dependent on the primary key.

3NF: No transitive dependencies exist.

**Payment:**

1NF: All attributes are atomic and do not contain repeating groups.

2NF: No partial dependencies exist, and the entity is fully dependent on the primary key.

3NF: No transitive dependencies exist.

**Delivery:**

1NF: All attributes are atomic and do not contain repeating groups.

2NF: No partial dependencies exist, and the entity is fully dependent on the primary key.

3NF: No transitive dependencies exist.

**Coding Part**

**- Function which counts the number of records**

CREATE FUNCTION count\_records\_1

RETURN NUMBER

IS

num\_rows NUMBER;

BEGIN

SELECT COUNT(\*) INTO num\_rows FROM Order\_1;

RETURN num\_rows;

END;

**- Procedure which uses SQL%ROWCOUNT to determine the number of rows affected**

CREATE PROCEDURE row\_count\_procedure\_0

AS

BEGIN

DELETE FROM Product WHERE product\_name LIKE 'Fish%';

DBMS\_OUTPUT.PUT\_LINE('Number of rows affected: ' || SQL%ROWCOUNT);

END;

**- Add user-defined exception which disallows to enter title of item (e.g. book) to be less than 5 characters**

CREATE OR REPLACE TRIGGER check\_title\_length

BEFORE INSERT ON Product

FOR EACH ROW

DECLARE

title\_length INTEGER;

BEGIN

title\_length := LENGTH(:NEW.product\_name);

IF title\_length < 5 THEN

RAISE\_APPLICATION\_ERROR(-20001, 'Product name must be at least 5 characters long.');

END IF;

END;

**- Create a trigger before insert on any entity which will show the current number of rows in the table**

CREATE OR REPLACE TRIGGER show\_row\_count

BEFORE INSERT ON Order\_1

DECLARE

num\_rows NUMBER;

BEGIN

SELECT COUNT(\*) INTO num\_rows FROM Order\_1;

DBMS\_OUTPUT.PUT\_LINE('Current number of rows in Order\_1 table: ' || num\_rows);

END;