**Restaurant Management System**

**Case study of restrauntant manage system :**

"Taste Haven," a well-known restaurant, has made the decision to replace its manual operations with a cutting-edge technology system for successful administration. They want to improve the customer experience, inventory management, and overall business performance by streamlining their procedures. A Restaurant Management System (RMS) is suggested as the approach since it integrates several components to centralise and automate operations.

1. Business Conditions The following are the main specifications provided by Taste Haven for the RMS:

1. Order management: The software must enable clients to place orders through a variety of channels (online, by phone, or in person) and offer real-time information on order status. Staff should be able to effectively manage orders, monitor delivery status, and handle order adjustments thanks to it.
2. Menu and Inventory Management: The RMS need to make it simple to develop and maintain a thorough menu with a range of categories, prices, and promotional offers. It should also keep track of inventory levels, provide warnings when an item is running short, and create automatic buy orders to restock supplies.
3. Table bookings and seating should be facilitated by the system, which should also control waitlists and effectively assign seating to visitors. Staff members ought to be able to look up table availability, follow reservations, and update occupied table statuses.
4. Role and responsibility management, attendance monitoring, and staff scheduling are all duties that should be handled by the RMS. Additionally, it should provide salary records, monitor employee progress, and manage tip distribution.
5. Analytics and Reporting: The system should produce information on sales, inventory, employee productivity, and customer satisfaction. These reports ought to help management make data-driven choices, spot patterns, and boost operational effectiveness.

System Architecture 2. A web-based application with a centralised database will be created for the RMS. The following modules will be included:

1. User Interface: The system will have an intuitive user interface that is accessible to management, employees, and consumers. Online purchasing, reservation management, and reporting dashboards are just a few of the features it will include.
2. Order Processing: This module will take care of placing, amending, and cancelling orders. To maintain a smooth order flow and give consumers real-time information, it will interface with the kitchen display system.
3. Administration of the restaurant's menu, including its products, categories, and prices, is possible through this module. Additionally, it will check inventory levels, create purchase orders, and send warnings for goods with low supply.
4. Table Management: This module is in charge of managing the waitlist, seating plans, and reservations for tables. The ability to change table statuses as guests come and go will allow staff to adjust table availability in real-time.
5. Schedule management, attendance monitoring, and payment processing are all handled by the staff and payroll management module. Additionally, it will have capabilities for tip distribution and performance tracking.
6. Reporting and analytics: This module will produce reports on a variety of operational elements of the restaurant, such as sales, inventory, staff productivity, and client feedback. To assist in decision-making, it will offer graphical representations and data visualisation.

4. Rewards and Anticipated Results Taste Haven anticipates the following advantages from using the RMS:

a) Increased Productivity: Automating order processing, inventory control, and other administrative operations will decrease manual mistakes, free up time, and improve productivity.

The system will offer a smooth ordering procedure, precise order tracking, and speedy table bookings, all of which will improve the client experience.

c) Effective Inventory Management: Stock outs, waste, and optimal inventory management may all be avoided with real-time inventory tracking, automatic purchase order production, and low stock warnings.

d) Simplified Staff Management: The RMS will make payroll computations, personnel scheduling, and attendance monitoring simpler. Staff will be motivated and productivity will be increased by features like tip distribution and performance tracking.

Class Diagram (Example): A class diagram represents the structure and relationships between the classes in a system. In the context of an RMS, the following classes and their relationships could be included:

1. Class: Restaurant
   * Attributes: name, address, contactNumber
   * Operations: addMenu(), removeMenu(), updateMenu()
2. Class: Menu
   * Attributes: name, price, category
   * Operations: getPrice(), getCategory()
3. Class: Order
   * Attributes: orderNumber, orderStatus, totalPrice
   * Operations: calculateTotalPrice(), updateStatus()
4. Class: Customer
   * Attributes: name, contactNumber, email
   * Operations: placeOrder(), makeReservation()
5. Class: Staff
   * Attributes: name, position, salary
   * Operations: manageInventory(), updateOrderStatus()
6. Class: Table
   * Attributes: tableNumber, capacity, availability
   * Operations: checkAvailability(), reserveTable()
7. Class: Inventory
   * Attributes: itemName, quantity, supplier
   * Operations: trackStock(), generatePurchaseOrder()

Test Case (Example): A test case is a detailed description of a specific scenario to verify if a system meets its requirements. Here's an example test case for the RMS:

Test Case: Place Order and Update Status

1. Description: Verify if the system allows customers to place an order and updates the order status correctly.
2. Precondition:
   * The system is running and accessible.
   * Menu items are available with correct pricing and categories.
   * The customer has logged into the system.
3. Steps:
   * Select desired items from the menu.
   * Specify the quantity for each item.
   * Click on the "Place Order" button.
   * Verify that the order is successfully placed and assigned a unique order number.
   * Check the order status.
   * Update the status to "In Progress."
   * Verify that the order status is updated correctly.
4. Expected Results:
   * The order should be successfully placed with a unique order number.
   * The initial order status should be "Received."
   * After updating the status to "In Progress," the order status should reflect the change.
5. Actual Results:
   * The order is placed successfully with an order number.
   * The initial order status is displayed as "Received."
   * After updating the status to "In Progress," the order status is updated correctly.
6. Pass/Fail:
   * The test case passes as all the expected results match the actual results.

**Data Flow Diagram of Restaurant Management System**

**O Level DFD: (Context DFD )**

Resturant Staff

Customer

Restaurant Management System

Ingredrents

Menu items

Reservation

Order items

Payment

Sales

**Over All Structure of RMS**

**Level 1 (DFD):**

Person

+

Email/Password

Full detail

Login

Registraion

Customer

List of dishes

Choose dishes

Order

Give Payment

Confirmed Order

**Level 2 DFD:**

Login

Email/Password

Person

+

Registration

Full detail

Order Management System

List of Dishes

**Menu List**

Food and Menu Management

Menu Database

**Order List**

**Menu List**

**Confirmation**

**Payment Detail**

**Choose Dish**

Customer

Order database

Restaurant Crew

**Reservation detail**

**Reservation detai**l

Reservation Mnanagement System

Reservation database

Payment Management System

Payment database

**Payment Detail**

**Entity Relation Diagram Of Restaurant Management System**

**Object:** Construct an ER diagram for a Restaurant Management System where customers can purchase one or more dishes.

**Step1:**

From the given project the entity sets identified are:

1. Customer
2. Menu items
3. Staff
4. Order
5. Reservation
6. Restaurant

**Step 2:**

The relevant attributes of Customerentity set are:

1. Customer\_id
2. Customer\_name
3. Email

The relevant attributes of Menu item entity set are:

1. Menu\_Item\_id
2. name
3. Price

iv) Description

The relevant attributes of Staff entity set are:

1. Staff\_id
2. Name
3. Salary
4. Position
5. Email
6. Hire date

The relevant attributes of Order entity set are:

1. Order\_id
2. Customer\_id
3. Item\_id

Iv) Order date

1. Order item

The relevant attributes of Reservation entity set are:

1. Reservation\_id
2. Reservation\_date

The relevant attributes of Restaurant entity set are:

1. Restaurant\_id
2. Name
3. Address
4. Contact

**Step 3:**

Identify the Relationship between entity sets

Last\_name

Firsrt\_name

Restaurant\_id(PK)

Staff\_id

address

Contact

name

Staff

Has

Restaurant

name

Salary

1 M

(1,1) (1,N)

Resturant\_id (FK)

hire\_date

position

email

R1

R2

R3

R4

.

.

.

S1

S2

S3

.

.

.

**Relational Schema:**

|  |  |  |  |
| --- | --- | --- | --- |
| Restaurant id | name | email | contact |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

Schema Resturant: (Resturant id, name, email, contact)

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Restaurant id | Staff id | First name | Last name | position | salary | Hire date | email |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |

Schema Staff Relation: (Restaurant id, Staff id, First name, Last name, position, salary, hire date,email)

Note:

Restaurant id determines -----> name, email, contact

Staff id determines ------> First name, Last name , position , salary, hire date, email

Customer\_id

(PK)

Staff\_id(PK)

Last\_name

Firsrt\_name

email

email

Resturant\_id (FK)

position

hire\_date

Customer

Deals

Staff

1 N

(1,N) (1,N)

C1

C2

C3

.

.

.

S1

S2

S3

.

.

.

name

First\_name

Last\_name

**Relational Schema:**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Staff id | First name | Last name | position | salary | Hire date | email |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |

Schema Staff: (Restaurant id, Staff id, First name, Last name, position, salary, hire date,email)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Staff id | Customer id | First name | Last name | email |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

Schema Customer Relation(Staff id, Customer id, First name , Last name, email)

Note:

Resturant id determines ----> Staff id (In the above Reational Schema)

Customer id determines -----> First name , Last name ,email

So;

Staff id is Primary Key and represent customer schema as Forign key

Customer\_id(PK)

Order\_date

Order\_id (PK)

First\_name

Order

O1

O2

O3

.

.

.

.

C1

C2

C3

.

.

.

email

Last\_name

Customer

Gives

1 N

(1,N) (1,N)

j

Order\_items

|  |  |  |  |
| --- | --- | --- | --- |
| Customer id | First name | Last name | email |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

Schema Customer: (Customer id, First name , Last name, email)

|  |  |  |
| --- | --- | --- |
| Order id | Order date | Customer id |
|  |  |  |
|  |  |  |
|  |  |  |

Schema Order( Customer id, Order id, Order date)

**Note:**

Customer id determines----->First name Last,Last name, email,order id

Restaurant id determines-----> Staff id, Restaurant items

Staff id determines -------> customer id

Order id ------> Order date

So;

Resturant id is our Primary Key

**Note 2:**

Restaurant itemcontinously change mean database does not represent historic data.

|  |  |  |
| --- | --- | --- |
| Order id items | Order items1 | Order items2 |
|  |  |  |
|  |  |  |
|  |  |  |

Order id determines -----> Order items because Order items is Multi-value attribute

description

name

Menu\_item\_id(PK)

name

Restaurant\_id(PK)

Resutaurant\_id(FK)

price

Customer\_id(FK)

Restaurant\_id(FK)

Reservation\_id(PK)

Reservation\_date

address

contact

Reservation

Menu\_items

Restaurant

1 N

(1,1) (1,N)

1

(1,1)

Note:

Restaurant id is the primary key and represent Menu items and Reservation as a Forign key

But Reservation has many menu items means it has multi value instance

And Many menu item given for many reservations

|  |  |  |  |
| --- | --- | --- | --- |
| Menu item id | name | price | Description |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

Schema Menu: (Menu item id, name, price, description)

|  |  |
| --- | --- |
| Reservation id | Reservation date |
|  |  |
|  |  |
|  |  |

Schema Reservation: ( Reservation id, Reservation date)

|  |  |  |  |
| --- | --- | --- | --- |
| Restaurant id | Customer id | Reservation id | Menu item id |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

Schema Relationship(Restaurant id, customer id, reservation id, Menu item id)

**Complete ERD**

Last name

First\_namee

Restaurant\_id

name

address

Staff id

Email

Restaurant

Staff

has

position

**(1,1) (1,1)**

hire\_date

contact

name

First name

Customer\_idd

email

Last name

name

Deal

IN/OUT

**(1,N)**

Customer

**(1,N)**

name

Food item id(PK)

Order date

Order id

Order

reservation id

O

Order\_items



Reservation

(0,N)

(1,1)

Reservation date

From

Generate recipt

Give

name

Ingedient no

quantity

(1,N)

(1,1)

Food items

Ingredients

Payment

date

description

price

method

amount

**COMPLETE RELATIONAL SCHEMA:**

|  |  |  |  |
| --- | --- | --- | --- |
| **Restaurant id(PK)** | **Name** | **Address** | **Contact No** |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

Schema Restaurant(Restaurant id, Name, Address, Contact)

**Super Key:**

1. {Contact No}
2. {Contact No, Restaurant id}
3. {Contact No, Name}
4. {Contact No, Address}
5. {Contact No,Name,Address}
6. {Contact No,Name, Restaurant id}
7. {Contact No,Address, Restaurant id}
8. {Contact No, Name, Address, Contact No}
9. {Restaurant id}
10. {Restaurant id,Name}
11. {Restaurant id, Address}
12. {Restaurant id,Contact No}
13. {Restaurant id,Name,Address}
14. {Restaurant id,Name,contact No}
15. Restaurant id, Address ,Contact No}
16. {Restaurant id, Name, Address,Contact No}

**Candidate Key:**

1. {Restaurant id}
2. {Contact No}

**Primary Key:**(Only one selected by system Analyst)

1. {Restaurant id}

**Alternative Key:**

1. {Contact No}

**Composite Primary Key:**

{Restaurant id, Contact No}

**Secondary Key:**

1. {Name}
2. {Address}

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Customer id(PK)** | **First Name** | **Last Name** | **Email** | **Restaurant id(FK)** |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

Schema Customer:(Customer id, First Name, Last Name, Email, Restaurant id)

**Super Key:**

1. {Email}

2) {Email,Firsrt Name}

3) {Email, Last Name}

4) {Email,Customer id}

5) {Email,First Name,Last Name}

6) {Email,First Name,Customerid}

7) {Email, Last name ,Customer id}

8) {Email, First name, Last name,Customer id}

1. {Customer id}
2. {Customer id,Firsrt Name}
3. {Customer id, Last Name}
4. {Customer id,Email}
5. {Customer id,First Name,Last Name}
6. {Customer id,First Name,Email}
7. {Customer id, Last name ,Email}
8. {Customer id, First name, Last name,Email}

**Candidate Key:**

1. {Customer id}
2. {Email}

**Primary Key:**

1) {Customer id}

**Alternative Key:**

1. {Email}

**Composite Key:**

1. {Customer id, email}

**Secondary Key:**

1. {First name, last name}

**Forign Key:**

1. {Restaurant id}

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Staff id(PK)** | **First name** | **Last name** | **Email** | **Positioin** | **Hire date** | **Restaurant id(FK)** | **Customer id(FK)** |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |

Schema staff(Staff id, First name, Last name, Email, Position, Hire date.Restaurant id, Customer id)

Super Key:

1. {staff id}
2. {staff id,First name}
3. {staff id, Last name}
4. {staff id, Email}
5. {staff id, Position}
6. {staff id, Hire date}
7. {staff id,First name, last name }
8. {staff id , last name , Postion}
9. {staff id, Email, hire date}
10. {staff id Email, Position}

Same case as Email

Hence there are 48 super key in this table because (2^n-1) +(2^n-1) - (2^n-2)

**Candidate Key:**

1. {staff if}
2. {email}

**Primary Key:**

1. {staff id}

**Alternative Key:**

1. {email}

**Composite Key:**

1. {staff id, email}

**Secondary Key:**

1. {first name, last name, position, hire date}

**Foreign Key:**

1. {Restaurant id, Staff id}

|  |  |  |
| --- | --- | --- |
| **Order id(PK)** | **Order date** | **Customer id(FK)** |
|  |  |  |
|  |  |  |
|  |  |  |

Schema Order(Order id, Order date, Cumtomer id)

**Super Key:**

1. {Order id}
2. {Order id, Order date}

**Candidate Key:**

1. {Order id}

**Primary Key:**

1. {Order id}

**Secondary Key:**

1. {Order date}

**Forign Key:**

1){Customer id}

|  |  |  |  |
| --- | --- | --- | --- |
| **Order id(FK)** | **Order item 1** | **Order item 2** | **Order item 3** |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

Order Relation(Order id, Order item 1,Order item 2, Order item 3)

**Super Key:**

1. {Order id}
2. {Order id, Order item1}
3. {Order id, Order item 2}
4. {Order id, Order item 3}
5. {Order id, Order item 1, Order item 2}

Hence there are 8 super keys because 2^3

**Candidate Key:**

1. {Order id}

**Secondary Key:**

1. {Order item 1}

2) {Oder item 2}

3) {Oder item 3}

**Foreign Key:**

1. {Order id}

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Food items id(PK)** | **Name** | **Quantity** | **Price** | **Description** | **Order id(FK)** |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |

Schema Food item(Food item id,Name,Quantity,Price, Description,Order id)

**Super key:**

1. {Food item id}
2. {Food items id,Name}
3. {Food item id, Quantity}
4. {Food item id, Price}
5. {Food item id, Description}
6. {Food item id,Name ,Quantity}
7. {Food item id, Name, Price}

Hence there are 16 super keys because 2^4

**Candidate Key:**

1. {Food item id}

**Primary Key:**

1. {Food item id}

**Secondary Key:**

1. {Name}
2. {Quantity}
3. {Description}

**Foreign Key**:

1. {Order id}

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Payment id(PK)** | **Date** | **method** | **amount** | **Order id (FK)** |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

Schema Payment:(Payment id, Date, method, amount,Order id)

**Super Key:**

1. {payment id}
2. {payment id,Date}
3. {payment id, method}
4. {payment id,amount}
5. {payment id, Date,method}
6. {payment id, method, amount}
7. {payment id, Date, amount}
8. {payment id,Date,method,amount}

**Candidate Key:**

1. {payment id}

**Primary key:**

1. {payment id}

**Secondary Key:**

1. {Date}
2. {method}
3. {amount}

**Foreign Key**:

1. {Order id}

|  |  |  |  |
| --- | --- | --- | --- |
| **Reservation id(PK)** | **Date** | **Quantity** | **Customer id(FK)** |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

Schema Reservation(Reservation id, Date,Quantity, Customer id)

**Super Key:**

1. {Reservation id}
2. {Reservation id, Date}
3. {Reservation id, Quantity}
4. {Reservation id,Date,Quantity}

**Candidate Key:**

1. {Reservation id}

**Primary Key:**

1. {Reservation id}

**Secondary Key:**

1. {Date}
2. {Quantity}

**Foreign Key:**

1. Customer id

|  |  |  |  |
| --- | --- | --- | --- |
| **Receipt id (PK)** | **Customer id(FK)** | **Restaurant id(FK)** | **Order id(FK)** |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

Schema Recipt(Recipt id, Customer id,Restaurant id,Order id)

**Super Key:**

1. {Receipt id}

**Candidate Key:**

1. {Receipt id }

**Primary Key:**

1. {Receipt id}

**Foreign Key:**

1. {Customer id}
2. {Restaurant id}
3. {Order id}