Restaurant Management DB

**SAI UDAY VAJRAPU**

<https://github.com/UdayVajram/Restaurant-Database.git>

**May 2, 2022**

# **Week 1**

I have created a business application database called "Dining app" that I created to store all the information that I need to take orders from customers. I need to make a database that keeps track of all the information about customers, employees, and the restaurant's food menu and dining layout. I need to make sure that all of this information is stored in a way that makes it easy for users to find it. In the beginning, my goal is to get into the system. To get into the system, only employees should be able to get in with their login credentials, and then their orders should be kept track of. Later, when an employee logs into the system, the restaurant seating layout should show up. This way, the customer's name, order type, choice of dining, age, and loyalty of the customers can be tracked, so they can get complimentary things. Later, the next table layout should hold the order value, which is where I want to store the entire food menu with the type of cuisine it is. This table should hold the order value. The employee should be able to figure out how old the customers are in order to serve alcohol. It's also important to keep an eye on the food stock option so that customers don't have to wait for the food they want. Finally, the store manager or owner should be able to figure out what to do when there's not enough food.

All these data are tracked to keep an eye of most selling dishes. Once these most selling dishes is listed, it will be easy to recommend it to the customers, when the customers are indecisive to order.

# **Week 2**

In this Week's assignment, I created a database for my Dining App idea. Here, I've created 5 different charts, each with at least 3 or more attributes and at least 3 tuples to display the data records.

Note: In this report, I've used a format that looks something like this**: Name (Datatype (Length))**

In my 1st table called

**Customer\_details**

I have created 4 attributes with some records like

|  |  |  |  |
| --- | --- | --- | --- |
| Customer Name (**Varchar** (**100**)) | Mode of Dining (**Varchar** (**100**)) | Order Type (**Varchar** (**100**)) | Order Number/Bill (**Int** (**255**)) |

**Employment Information:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Employee Name (**Varchar** (**100**)) | Employee Id (**Int** (**255**)) | Age (**Int** (**255**) | Place of Work (**Varchar** (**100**)) | Hours/Week (**Int** (**255**)) |

**Kitchen Backend:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Stock (**Int** (**255**)) | Unit (**Int** (**255**)) | total orders (**Int** (**255**)) | Customer Loyalty (**Boolean(100))** | Veg or Non-Veg (**Int** (**255**)) |

**Restaurant Frontend:**

|  |  |  |  |
| --- | --- | --- | --- |
| Table Number (**Int** (**255**)) | No of Guests (**Int** (**255**)) | Employee Id (**Int** (**255**)) | Choice of Cuisine (**Int** (**255**)) |

**Menu:**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Appetizers (**Int** (**255**)) | Entrée (**Int** (**255**)) | Sides (**Int** (**255**)) | Desserts (**Int** (**255**)) | Beverages (**Int** (**255**)) | Kids Menu (**Int** (**255**)) |

The above tables are created with at least 3 tuples of record to understand the database.

# **Week 3**

In this Week assignment I have established a relationship in between tables by using both vertabelo and UMLET for my Restaurant Management DB.

Diagram

Description automatically generated

**ER Diagram – Crow’s Foot Model with vertabelo:**

Diagram

Description automatically generated

**Week 4**

In this week’s assignment, I have downloaded the SQL script, which is generated by vertabelo and then, load the data in tables into database using csv files.

With the help of SQL query, I have created all the tables into the database RDM in MariaDB. Here, 5 tables have been created “customerdetails, employmentinformation, kitchenbackend, Menu, Restaurantfrontend ”.A screenshot of a computer

Description automatically generatedA computer screen capture

Description automatically generated with medium confidenceA screenshot of a computer

Description automatically generated with medium confidence

Later, Customerdetails data is loaded into customerdetails table in RDM database using customerdetails.csv with the help of the following command

“LOAD DATA LOCAL INFILE 'Z:\LEWIS\Database Systems\Week 2\Project tables\\customerdetails.csv'

INTO TABLE customerdetails

FIELD TERMINATED BY ','

LINES TERMINATED BY '\R\N'

IGNORE 1 LINES

(Customerdetails, ModeofDining, orderType, BillNumber, TableNumber)” for all the 5 tables by replacing the required name and lines and then the tables with data looks like below.

A screenshot of a computer

Description automatically generated with medium confidenceA screenshot of a computer

Description automatically generated

**Data Manipulation Language Scripts**

1. **Insert command**

The 1st command used is insert which will help us to insert additional data into the tables and hence, I have added extra information into CUSTOMERDETAILS by updating an extra row to these (customerName, ModeofDining, orderType, BillNumber and TableNumber) = (‘Sai Uday’, ’Private Party’, ‘Customized Menu’, 10, 8). And also to employmentInformation by adding same details.

A screenshot of a computer

Description automatically generated with medium confidenceA screenshot of a computer

Description automatically generated with medium confidence

1. **Update Command**

This command is used to update the data that already existing in the table. As I have already loaded extra data into customerdetails using insert command. Now, I have updated the same data from CUSTOMIZED MENU to NONE and in the 2nd table I have updated the employmentinformation of Sai Uday’s placeofwork to entire restaurant.

A screenshot of a computer

Description automatically generated with medium confidenceA screenshot of a computer

Description automatically generated with medium confidence

1. **Delete Command**

From the above information we know that Sai Uday data is added into customerdetails and into emloymentinformation, by using the delete command I have removed the data from both the tables.

A screenshot of a computer

Description automatically generated with medium confidence

1. **Select Command**

This simple select command is used to select data from the tables or from the database. So in the below image I have selected both BillNumber and EmployeeName from both the tables customerdetails and employmentinformation.

A screenshot of a computer

Description automatically generated with medium confidence

1. **Join Commands**

This command is used to help join tables when there is common information. I have used this command to join 2 tables restaurantfrontend and employmentinformation.A picture containing text

Description automatically generated

1. Summary Command:

In Summary Command, I have used three queries

1. Max
2. Avg
3. Sum

The above 3 queries I have used to get the max value, avg value and Sum values for the employmentinformation with the column name Age.

A screenshot of a computer

Description automatically generated with medium confidence

1. **Multi Table query**

This query is used to join 2 different tables with different data. So, I have used my Restaurantfrontend and employmentinfomration.

A screenshot of a computer

Description automatically generated with medium confidence

Week 5

**Indexes**

In this topic I have considered two tables i.e., **employmentinformation** and **kitchenbackend**. I chose those tables in particular because, in my database, employment information is critical since it can overwhelm data, which in turn can cause the entire database operation to stall. As I read on a blog, it took them more than seven minutes to load a query that helps them identify information about an employee in their 11M database, and much longer to display the results. At the same time, with a few tweaks to the database queries and the assistance of indexes, it only takes approximately 6.5 seconds to load the results. In a database, the job of INDEX is straightforward; an index is simply a pointer to the data contained within a table. It is extremely comparable to an index in the back of a book when it comes to indexes in databases. The commands I have used to implement Indexes in my database is

1. **create index index1 on employmentinformation (EmployeeName, EmployeeId);**

The above query line is used to retrieve my EmployeeName and EmployeeId quickly, in order to save time.

Text

Description automatically generated

1. **create index index2 on kitchenbackend(TableNo, EmployeeID);**

The above command is used to create an index on my 2nd table Kitchenbackend.

A picture containing graphical user interface

Description automatically generated

**VIEWS**

Views are believed to be a virtual table, with rows and columns in the same manner as a physical table. To create a view, we can choose one or more tables from a database that already exists. When we delete a View from a database, it has no effect on the real tables included within the database. It is possible for database administrators to use views when everyone who has access to the database can only view the information and not make any changes to the information. The data will be more secure as a function of this.

I have applied the view commands on employmentinformation and kitchenbackend.

1. create view VIEW1 as select EmployeeName, EmployeeId from employmentinformation;

Text

Description automatically generated

1. **create view VIEW2 as select TableNumber, EmployeeID from Restaurantfrontend;**

A screenshot of a computer

Description automatically generated with medium confidence

The above command is used to retrieve information only that are needed ignoring all the data, because it hides/reduce the complexity in the SQL server. By using views, we can also implement row and column level security using view in the sql server.