**Restaurant Management System**

**Project Overview:**

Creating a Restaurant Management System SQL Project is a great way to demonstrate your database design, query, and back end integration skills.

Here's a structured outline and a working example you can build upon.

**Objective:**

This SQL-based system manages all operations related to a restaurant, including:

User Loing Page

Customers

Branch Dedailse

Menu Items

Restaurant Owner

Employee

Orders

Bill

Home Delivery

Efficient data querying using:

SELECT, INSERT, UPDATE, ALTER, LIKE, JOIN’S, AGGREGATE FUNCTION, CONCAT, SUBQUERIES, STOREDPROCEDURES ,GROUPBY, TRIGGER FUNCTION, VIEW FUNCTION, STRING FUNCTION

CREATE DATABASE valli\_Restaurant;

USE valli\_Restaurant;

commit;

rollback;

**USER LOING PAGE :**

create table USER(

User\_Id int NOT NULL AUTO\_INCREMENT,

Fname varchar(50) NOT NULL,

Lname varchar(50) NOT NULL,

Password varchar(50) NOT NULL,

PRIMARY KEY (`User\_Id`)

);

**CREATE CUSTOMER DEDAILS:**

CREATE TABLE Customer\_details(

CUSTOMER\_id int PRIMARY KEY AUTO\_INCREMENT,

Customer\_name varchar(30) DEFAULT NULL,

City varchar(20) DEFAULT NULL,

Address varchar(50) DEFAULT NULL,

Email\_id varchar(30) unique);

**CREATE BRANCH DEDAILS:**

CREATE TABLE Branch\_Detail (

branch\_id INT PRIMARY KEY AUTO\_INCREMENT,

branch\_name VARCHAR (100),

menu\_items TEXT,

combo\_price varchar(40));

**VEG MENU**

create table veg(

id int primary key auto\_increment,

veg\_name varchar(60) not null,

price varchar(50));

desc veg;

insert into veg(veg\_name, price) values

('mushroom curry','150'),

('Gobi manchurian dry','180'),

('sampar rice','50'),

('curd rice','60'),

('veg biriyani','130'),

('veg fried rice','150'),

('veg hakka noodels','130'),

('veg 65','110'),

('veg manchurian','150'),

('Gobi manchurian','150'),

('panneer chilly','130'),

('veg cripsy','120'),

('mushroom 65','90'),

('mashroom fried rice','150'),

('babycorn chilly',110);

select count(veg\_name) from veg;

select\*from veg;

**JUICE MENU:**

create table juice(

id int primary key auto\_increment,

juice\_name varchar(60) not null,

price varchar(50));

insert into juice(juice\_name, price) values

('Apple zing','80' ),

('Mango tango','75' ),

('pomegranate','85' ),

('berry bliss','80' ),

('orange','80' ),

('pineapple','100' ),

('fruit juice','120' ),

('sweet lime','50' ),

('mosambi','90' ),

('carrot','80' ),

('watermelon','80' ),

(' fresh lime soda','80' ),

('grap jcie','110' ),

('vegetable juice','90' ),

('fresh lime water','80' );

select count(juice\_name) from juice;

**MENU ITEMS:**

create table MENU(

menu\_id INT PRIMARY KEY AUTO\_INCREMENT,

name VARCHAR (100),

Pric varchar(30),

Rate float DEFAULT NULL,

Catagory varchar(40),

veg\_id int null,

foreign key(veg\_id) references veg(id),

juice\_id int null,

foreign key(juice\_id) references juice(id));

desc Branch\_Detail;

SHOW TABLES;

**INSERT CUSTOMER DEDAILSE:**

INSERT INTO Customer\_details (Customer\_name, City, Address, Email\_id) VALUES

('PremKumar','chennai', '390001','maria@gmail.com'),

('Ashok','chennai', '380001', 'tina@gmail.com'),

('Ram','chennai', '380001', 'mena@gmail.com'),

('Gopal','karaikudi', '380009', 'riya@gmail.com'),

('Arsath','thiricy', '380008', 'raj@gmail.com'),

('Dhoni','chennai', '3800780', 'vijay@gmail.com');

**INSERT BRANCH DEDAILS:**

INSERT INTO Branch\_Detail (branch\_name, menu\_items,combo\_price) VALUES

("kanyakumari", "Biryani, fish curry,muton chukka", 450),

("chennai", "Biryani, Pongal, Idly", 600),

("karaikudi", "Biryani, chettinad spcial,uppukari",650),

("Coimbatore", "Dragon\_Chicken, keema\_dosa, Prawn\_65",500),

("Madurai", "Chicken\_65, parrota, muttoun curry",400),

("Hyderbad", "Kal\_dosa, Naan, Panner\_gravy", 350),

("pondychery", "potato uthappam, mushroom dragon",450);

**INSERT MENU ITEM:**

INSERT INTO MENU (name,Pric,Rate,Catagory,veg\_id,juice\_id) VALUES

("Chicken Tikka",160,4,"Starters",1,1),

("Tandoori Chicken",250,4.5,"Starters",2,2),

("Tandoori King Prawn",200,4.3,"SeaFood",3,3),

("King Prawn Rosun",250,3,"SeaFood",4,4),

("Mushroom Rice",150,4.2,"Rice",5,5),

("Chicken Fry Rice",150,4.2,"Rice",6,6),

("Garlic Naan",130,4.4,"Bread",7,7),

("Kal\_dosa",50,4,"Starters",8,8),

("Chapati",30,4,"Bread",9,9),

("Chicken\_65",210,4.2,"Starters",10,10),

("parrota",30,4.3,"Starters",11,11),

("muttoun curry",180,4,"Starters",12,12),

("Pongal",50,3.3,"Starters",13,13),

("muton chukka",250,4,"Starters",14,14),

("potato uthappam",50,4,"Starters",15,15);

select\*from USER;

select\*from veg;

select\*from juice;

select\*from Customer\_details;

select\*from Branch\_Detail;

select\*from MENU;

select count(name) from MENU;

**JOINS:**

select e.menu\_id,e.name,e.Pric,e.Rate,e.Catagory,e.veg\_id,c.id,c.veg\_name,c.price

from MENU as e inner join veg as c on veg\_id=c.id;

**INNER JOINS:**

To join data from two different tables into one using an SQL INNER JOIN,

you need a common column (usually a foreign key) that exists in both tables. Here's a basic structure

select e.menu\_id,e.name,e.Pric,e.Rate,e.Catagory,

c.id,c.veg\_name,c.price,

d.id,d.juice\_name,d.price

from MENU as e

inner join veg as c on veg\_id=c.id

inner join juice as d on e.juice\_id=d.id;

**CREATE RESTAURANT OWENER:**

create table OWNER(

Fname varchar(15) NOT NULL,

Lname varchar(15) NOT NULL,

Contact varchar(100) NOT NULL,

Rest\_Name varchar(100) NOT NULL,

PRIMARY KEY (`Fname`,`Lname`,`Contact`));

**INSERT RESTAURANT OWENET DEDAILS:**

insert into `OWNER`(`Fname`,`Lname`,`Contact`,`Rest\_Name`)values

("sekar","muniyandi","894056438","valli\_restaurant");

select\*from OWNER;

**ADD THE NEW ONE COLUMN:**

To add one column to an existing SQL table using the ALTERÂ statement

alter table Customer\_details add mobile\_number varchar(50) unique after Address;

**MULTIPLE VALUESE UPDATE:**

You might want to update multiple rows with different values in a single SQL statement.

update Customer\_details set mobile\_number =case

when CUSTOMER\_id=1 then 8940058241

when CUSTOMER\_id=2 then 9840058241

when CUSTOMER\_id=3 then 9840058875

when CUSTOMER\_id=4 then 9887654321

when CUSTOMER\_id=5 then 9865743269

else 9865432768

end;

**DROP THE ONE COLUMN IN COUSTOMER DEDAILS:**

To remove (drop) a column from a table, you use the ALTER TABLE statement with DROP COLUMN

alter table Customer\_details drop column Address;

select\*from Customer\_details;

**AGGREGATE FUNCTION:**

select min(Pric) from MENE;

\*Show the minimum price in menu item

select max(Pric) from MENU;

\*Show the maximum price in menu item

select count(name) from MENU;

\*Show the how many item in menu

**RESTAURANT MANAGER:**

create table MANAGER(

Manager\_Id int NOT NULL AUTO\_INCREMENT,

Fname varchar(15) NOT NULL,

Lname varchar(15) NOT NULL,

Contact varchar(20) NOT NULL,

Address varchar(30) DEFAULT NULL,

Salary varchar(30) DEFAULT NULL,

Sex varchar(10) DEFAULT NULL,

Bdate date DEFAULT NULL,

Join\_Date date NOT NULL,

PRIMARY KEY (`Manager\_Id`));

insert into MANAGER(Fname,Lname,Contact,Address,Salary,Sex,Bdate,Join\_Date)

values

("babu","shankar","9642540626","E-28 OBH , IIIT chennai","30000","Male","1992-09-29","2014-08-01"),

("saravana","kumar","9581871321","E-27 OBH , ILIT karaikudi","40000","Male","1993-11-18","2012-08-01");

select\*from MANAGER;

**CONCAT:**

function is used to join two or more strings (columns) into one.

select Manager\_Id, concat(Fname, ' ',Lname)as Full\_name ,Contact,Address,Salary,Sex,Bdate,Join\_Date from MANAGER;

**CREATE RESTAURANT EMPLOYEES:**

CREATE TABLE employee(

Emp\_Id int primary key auto\_increment,

Emp\_name varchar(10) NOT NULL,

Designation varchar(10) DEFAULT NULL,

Address varchar(50) DEFAULT NULL,

City varchar(10) DEFAULT NULL,

State varchar(20) DEFAULT NULL,

Pincode varchar(6) DEFAULT NULL,

Contact\_no varchar(13) DEFAULT NULL unique,

Emp\_salary float DEFAULT NULL,

salary\_increass varchar(40) null,

hiring\_date date DEFAULT NULL,

check(Emp\_salary >=0)

);

**RESTAURANT EMPLOYEES DEDAILS:**

INSERT INTO employee (Emp\_name,Designation,Address,City,State,Pincode,Contact\_no,Emp\_salary,salary\_increass,hiring\_date) VALUES

( 'Akshat', 'chef','ashram road', 'ahmedabad','gujrat', '380009', '9874561230', 15000, 1000,'2019-04-14'),

( 'Vishal', 'chef','sarma road', 'chennai','Tamilnadu', '630554', '9874561867', 16000, 1000,'2011-03-14'),

( 'Hardik', 'chef', 'gandhiroad', 'chennai','Tamilnadu', '630557', '8974561275', 12000, 1000,'2015-08-24'),

( 'Brijesh', 'cashier','cg road', 'karaikudi','Tamilnadu', '630544', '9774561230', 20000,1000, '2015-01-10'),

( 'Jugal', 'cashier', 'mg road', 'karur','Tamilnadu', 630558,'8974561230', 25000, 1000,'2019-09-14'),

( 'ram', 'cashier', 'kk road', 'thirupur','Tamilnadu', 630559,'8974561287', 25000, 1000,'2019-09-14'),

( 'praven', 'waiter', 'mn road', 'karur','Tamilnadu', 630558,'8974361230', 21000,1000, '2019-06-04'),

( 'kavin', 'waiter', 'hp road', 'vellor','Tamilnadu', 630564,'8974567894', 17000,1000, '2019-03-17'),

( 'keerthi','cleaner', 'ram road', 'karaikudi', 'Tamilnadu', 630765,'8940058765',1000, 11000, '2022-09-14'),

( 'harshitha', 'reception', 'jj road', 'Tirupati','anthra', 649876,'8974568796', 13000,1000, '2020-05-17');

select\*from employee;

**GROUP BY:**

Selects each unique designation and the number of employees with that designation.

select Designation, count(Designation)count from employee group by Designation

having count(Designation)>1;

**TRIGGER FUNCTION:**

create a trigger that automatically updates an employee's salary by adding the salary\_increass value before any update.

create trigger tr

before update on employee

for each row

set new.Emp\_salary =(new.salary\_increass + new.Emp\_salary);

update employee set salary\_increass= 2000 where Emp\_Id=1;

select\*from employee;

**SUB QUERY:**

This query to display each employee's name and salary, along with the minimum and maximum salary across the entire employee table.

select Emp\_name,Emp\_salary,

(select min(Emp\_salary) from employee) as min, (select max(Emp\_salary) from employee) as max from employee;

**STRING FUNCTION:**

This function to display employee salaries with thousand separators and no decimal places. This is commonly used to improve readability.

select format(Emp\_salary,0) as salay from employee;

**VIEW:**

\*HOW MANY CHEF EMPLOYEE IN RESTAURANT

create view chef as select

Emp\_name,Address,City,State,Pincode,Contact\_no,Emp\_salary,hiring\_date from employee where Designation ='chef';

select\*from chef;

\*HOW MANY CASHIER EMPLOYEE IN RESTAURANT

create view cashier as select

Emp\_name,Address,City,State,Pincode,Contact\_no,Emp\_salary,hiring\_date from employee where Designation ='cashier';

select\*from cashier;

\*HOW MANY RECEPTION EMPLOYEE IN RESTAURANT

create view reception as select

Emp\_name,Address,City,State,Pincode,Contact\_no,Emp\_salary,hiring\_date from employee where Designation ='reception';

select\*from reception;

\*HOW MANY WAITER EMPLOYEE IN RESTAURANT

create view waiter as select

Emp\_name,Address,City,State,Pincode,Contact\_no,Emp\_salary,hiring\_date from employee where Designation ='waiter';

select\*from waiter;

**ORDER ITEMS:**

CREATE TABLE order\_items (

Order\_Id int primary key auto\_increment,

Food\_name varchar(20) DEFAULT NULL,

Quantity int(11) DEFAULT NULL);

DROP table order\_items;

**INSERT ORDER VALUES:**

insert into order\_items (Food\_name,Quantity) values

("Mushroom Rice",2),

("Chicken Fry Rice",2) ,

("potato uthappam",2),

("pineapple Rice",2);

select\*from order\_items;

**BILL:**

create table BILL(

Order\_Id varchar(55) NULL ,

Customer\_name varchar (20) NULL,

Name varchar(100) NULL,

Price varchar(20) NULL,

Total\_Amount double NULL

);

insert into BILL(Order\_Id,Customer\_name,Name,Price,Total\_Amount)values

("Order01","PremKumar","Biryani",150,null),

(null,null,"Dragon\_Chicken",120,null),

(null,null,"Kal\_dosa",150,null);

select\*from BILL;

**SUM USED :**

That adds up all values in the Price column in tottal amount.

select sum(Price) as Total from BILL;

create table MENU\_BILL(

Order\_Id int NOT NULL,

Name varchar(100) NOT NULL,

Price varchar(20) NOT NULL

);

drop table MENU\_BILL;

**FOOD ONLINE ORDER:**

create table HOME\_DELIVERY(

Delivery\_Id int NOT NULL AUTO\_INCREMENT,

Address varchar(50) NOT NULL,

Contact varchar(20) NOT NULL unique,

cust\_name varchar(40),

Order\_Id varchar(40) NOT NULL,

PRIMARY KEY(`Delivery\_Id`));

insert into HOME\_DELIVERY (Address,Contact,cust\_name,Order\_Id) values

("kk nagar",9876543212,"prem","ord01"),

("rn nagar",8976543249,"javid","ord07"),

("ram nagar",9876543765,"babu","ord06"),

("mm nagar",9776543278,"frank","ord03");

select\*from HOME\_DELIVERY;

**WILD CARD:**

This query retrieves all rows from the HOME\_DELIVERY table where the

customer name (cust\_name) matches a specific pattern.

SELECT \* FROM HOME\_DELIVERY

WHERE cust\_name LIKE "\_a%";

**STORED PROCEDURE :**

DELIMITER //

create procedure selectEmployeeById(

in p\_id int

)

begin

select\* from Customer\_details where CUSTOMER\_id = p\_id;

end //

DELIMITER //

call selectEmployeeById(1);