DBMS FINAL PROJECT REPORT:

WINTER SEMESTER 2022

CSE250 Database Management System

Project Title: Restaurant Management System

AU2040263 Dhruvi Shah

AU2040235 Kanvi Patel

AU2040162 Madhvendra Jhala

ABSTRACT:

The main aim of this project is to develop an online food ordering system. Through the webpage the users can order from varieties of restaurants that are registered in the website. The restaurant inventories are stored in the oracle database. The customers can access the restaurant web page easily on the world wide web.

DESIGN AND METHODOLOGY:

The technical tools used for the implementation of the project are:

- HTML, CSS are used for the creation and design of web pages.
- PHP is used for connecting the database with the front-end.
- The back-end data is stored in the oracle database server.
- Oracle is used for the creation of tables, as well as for the implementation of procedures and triggers.

WORKING MODEL DESCRIPTION:

The following diagrams describe the working flow model of the entire database into steps.

Description of Project:

Our project “Restaurant Management System”, is aimed to reduce the efforts required to manage the data and accounts of a restaurant. The main user for the database is the admin(manager) is allowed to login and see the entire database. A login page is created for user where the registered user as well as a new user can register and can view different hotels and also view menu and order food items. After selecting the items, the orders go into the cart from where there is a functionality provided for the user to add or remove orders as well as make changes in the quantity of the food items ordered. After finalizing the orders in the cart, the user can place the final order after which the user can select the bill-payment option. Also, the expected time for the arrival of order would also be visible to the user. Also, after placing the order, customers would be able to provide reviews on the food item and also the user would be able to read reviews of the food-item provided by other users. While at the admin level, the manager would be able to view all the orders and add or edit the orders. Along with a variety of restaurants, the user also gets an option to order food of multi-varied cuisine. The admin would also be able to update or add restaurants, items etc. into the menu and also update the price as well as description about the food-items.

IMPLEMENTATION:

DATABASE OVERVIEW:

The online restaurant database is made up of 6 tables, which are,

(1) Users: This contains all the information about the table customer. The attributes present in the table are:

(i) u_id: It is a special identity key, given to each and every customer, so that each customer could be identified uniquely.

(ii) Username: It contains the data of names of all customers. The main purpose of it being to display the name of the customer on the bill.

(iii) F_name: The abbreviation means the first name of the customer. It is used as an identification of the customer.

(iv) L_name: l_name means the last name of the customer, the data could be used to get information if the customers belong to the same family.

(v) Email: The attribute is used as a reference for the hotel management, so that they could connect with their customers in future too.

(vi) Phone: The phone no of customers is used for storing the record details of the customers, to keep them updated and also as a point of contact in future.

(vii) Password: This is created by the users to login into the database and access the services. This enables security for the users and secures their data.

CODE:

```
CREATE TABLE users (

u_id number(10) primary key,
username varchar2(222),
f_name varchar2(222),
```

```
l_name varchar2(222),
email varchar2(222),
phone varchar2(222),
password varchar2(222),
address clob
) ;
```

(2) users_orders: This table contains details about the different orders placed by different customers.

(i) O_id: This stands for order_id , which is the unique identity given to each and every order placed so that the information about the orders could be fetched all at once. This reduces data redundancy as otherwise each and every individual item in the order would store all the information (which would be the same for the order placed by the user).

(ii) Customer_id: The attribute here acts as a foreign key as it would give information about which order is placed by which customer, hence uniquely identifying the customer and order.

(iii) Product_id: It is used to uniquely identify each and every item of the order. As the entire name would consume a lot of space and hence unique keys are given to each and every item.

(iv) Title: This displays the name of the item ordered as it would be useful for displaying it in the final bill.

(v) Quantity: Sometimes the user may want to have the same item in more than one quantity and hence, this functionality provides the user the facility of ordering the same item and the user would not have to order the item again and again, they just need to increase or decrease the quantity of the item.

(vi) Price: The attribute displays the price of the item ordered. It also considers the quantity of item ordered and the price is displayed accordingly.

(vii) Status: This gives the user information about their order, whether it is placed, or ready to be served or delivered etc. This also helps the restaurant staff to get information about the orders.

CODE:

```
CREATE TABLE users_orders (

  o_id number(10) primary key,
  u_id number(10) references users(u_id),
  title varchar2(222),
  quantity number(10),
  price number(10,2),
  status varchar2(222),
  dt date default sysdate
);
```

(3) Dishes: This table gives information about the various food & beverages that are served by the restaurant.

(i) Product_id: This helps to uniquely identify each and every food or drinks that are being offered.

(ii) Rs_id: As there are many restaurants, hence it must also be taken care about which item is offered by which restaurant and hence this helps to develop linkage between restaurant and product.

(iii) Title: The attribute tells the name of the food item. Hence it would be easy for the user to order the item which they like.

(iv) Slogan: This gives the description of items that are offered by the restaurant and also talks about the ingredients that are used in the preparation and vaguely describes the process too.

(v) Price: The attribute describes the prices of the items that are served by the restaurant.

(vi) Img: The attribute shows the picture of the item that the restaurant serves.

CODE:

```
CREATE TABLE dishes (
  d_id number(10),
  rs_id number(10),
  title varchar2(222),
  slogan varchar2(222),
  price number(10,2),
  img clob, primary key(d_id,rs_id) ) ;
```

(4) Restaurant: As the database contains various restaurants and hence this table gives information about each and every restaurant.

(i) Rs_id: This helps in identifying each and every restaurant uniquely.

(ii) C_id: The attribute helps in identifying customers associated with each and every customer.

(iii) Title: This attribute describes the name of the various restaurants that are present in the database.

(iv) Email: The attribute gives information about the email address of the restaurant, as it would be helpful for the user as well as the restaurant to stay connected with the users.

(v) Phone: This facility will be useful as the users could directly contact the restaurant in case of any issues faced or for some other tasks which are related to the restaurant.

(vi) url: It's the individual web-address of the restaurant so that the user could visit it and get more information about the restaurant.

(vii) O_hr: This states the restaurant wise time of when the restaurant would open (i.e., start operating).

(viii) C_hr: This attribute gives information about the closing time of the restaurant, hence the user gets to know about the timings that they could visit the restaurant.

(ix) O_days : It describes the days when the restaurant would be functional.

(x) Address: It gives the restaurant's address, hence the customer could visit the place.

(xi) Image: This attribute shows the image of the restaurant, hence users could get information about the restaurant.

(xii) Date: The attribute helps to get information about the date on which activities would take place at the restaurant.

CODE:

```
CREATE TABLE restaurant (
  rs_id number(10) primary key,
  c_id number(10) references res_category(c_id),
  title varchar2(222),
  email varchar2(222),
  phone varchar2(222),
  url varchar2(222),
  o_hr varchar2(222),
  c_hr varchar2(222),
  o_days varchar2(222),
  address varchar2(222),
  image clob,
  dt date
) ;
```

(5) Res_Category: This table is mainly useful for the customer to decide upon which category they are interested in eating.

(i) C_id: The abbreviation stands for category_id, which uniquely identifies each and every category that is being offered by the restaurant. Also the database provides the facility that if a customer

directly searches for a category, they would get the list of restaurants that serve that category of food.

(ii) C_name: The category name attribute helps the customer to get information about the category of food, so they can order from the category they like.

(iii) Date: This attribute is useful as some of the food-items are only available in a particular season or only during some days and hence the date attribute helps in differentiating which food_items would be available during a particular time, when a user is ordering food.

CODE:

```
CREATE TABLE res_category (
 c_id number(10) primary key,
 c_name varchar2(222),
  dt date
);
```

(6) cart: This table is used to display the dishes that the user orders.

(i) cart_id : cart id associated will be different for each user.

(ii) u_id: It is a special identity key, given to each and every customer, so that each customer could be identified uniquely.

(iii)Title: This attribute describes the name of the various restaurants that are present in the database.

(iv) quantity : It will show the quantity of each dish that the user orders.

(v) Price : It will show the total price of all the dishes that the user orders.

CODE:

```
create table cart(
  cart_id number(10) primary key,
```

```
u_id number(10) references users(u_id),
title varchar2(222),
quantity number(10),
price number(10,2) );
```

(7) track user: This table will track the user's orders and the time of the order placed by the user.

(i) u_id: It is a special identity key, given to each and every customer, so that each customer could be identified uniquely.

(ii) systimestamp : It will display the current date and time including fractional seconds and time zone.

CODE:

```
Create table track_user(u_id references users(u_id), systimestamp date, action char(6));
```

(8) admin : This contains all the information about admin. The attributes present in the table are:

(i) adm_id : It is a special identity key for the admin.

(ii) adm_username : It contains the data of names of admins. It is used for log in.

(iii) adm_password : Admins will have their own passwords to log in.

(iv)adm_email : It is used as a reference for admins.

(v)adm_dt : It is useful to display the log in date of the user.

CODE:

```
CREATE TABLE admin (
   adm_id number(10) primary key,
   adm_username varchar2(222),
```

```
  adm_password varchar2(222),
  adm_email varchar2(222),
  adm_code varchar2(222),
  adm_dt date
 );
```

(9) remark : This contains all the information related to the user's remark about the food.

(i) id : remark id would be different for every user's remark.

(ii) user_id : It is different for every user to identify the user of the given remark.

(iii) order_id : It will help to know that the given remark is for which order.

(iv) remark : it would save the user's feedback of the food.

(v) remarkDate : remark date is useful to display the date when the feedback/ remark is given.

CODE:

```
CREATE TABLE remark (
  id number(10),
  user_id number(10),
  status varchar2(255),
  Order_id number(10),
  remark varchar2(222),
  remarkDate date
);
```

The following is the entire representation of how the database would be helpful, from a customer point of view. But there are many actions that are going on in the background but as that information being confidential can only be viewed by the restaurant manager or some higher-authority. As they have the responsibility of looking upon the entire working of the restaurant.

Also, the database has special login credentials for the manager, and upon logging the manager would be able to view the list of orders that were placed on a day, how many were served. And also based on that they can calculate, what are the food_items that customers are ordering more and hence can suggest it to other customers. The manager would also be able to view the bill_status and accounting details of the restaurant. As what is the amount paid by the users, hence it would also be helpful in keeping an account of the financial details of the restaurant. This also establishes the security of the database, as the data of users is safe and is not being accessed or is not known to anyone.

Thus, with these we conclude, hence the entire project is completed, keeping in mind the user needs, restaurant managers necessity and an entire database of restaurant management systems is created.

Values

```
begin
Insert into dishes values(1,1,’palak paneer’, ‘ A dish of antioxidant served with paneer cubes’ , ‘390.00’, ‘606d72f3cb12f.jpg’);
Insert into dishes values(2, 1, 'veg kofta', 'vegetables kofta served with tangy brown gravy', '250.00', '606d73302ece2.jpg');
Insert into dishes values(3, 1, 'paneer biryani', 'long grained rice like basmati flavored with fragrant spices such as saffron and layered with vegetables and thick gravy.', '470.00', '606d73771366a.jpg');
Insert into dishes values (4, 1, 'Paneer Tikka', 'different marinades- pudina, pahadi, amritsari.', '380.00', '606d73d2d37f4.jpg');
Insert into dishes values (5, 2, ' Spaghetti ', 'Spaghetti pasta, grilled shrimps, parmesan cheese, with our homemade sauce,', '380.00', '606d7491a9d13.jpg');
Insert into dishes values (6, 2, 'Lasagna', 'layers of pasta baked with layers of cheese.', '250.00', '606d74c416da5.jpg');
Insert into dishes values (7, 2, 'Cheese Cigar', 'cheese stuffed rolls', '460.00', '606d74f6ecbbb.jpg');
Insert into dishes values(8, 2, 'Spaghetti And Pasta', 'Pasta and Spaghetti served with your choice of gravy.', '380.00', '606d752a209c3.jpg');
Insert into dishes values (9, 3, ' Fried Rice', 'Fried rice is a dish of cooked rice that has been stir-fried in awok or a frying pan..', '350.00', '606d7575798fb.jpg');
Insert into dishes values (10, 3, 'Chinese Rolls', 'Chinese rolls are rolled appetizer or dim sum commonly found in chinese and other southeast asian cuisines', '120.00', '606d75a7e21ec.jpg');
```

```
Insert into dishes values (11, 3, 'Manchurian', 'Manchurian is a class of Indian chinese dishes made by roughly chopping and deep frying ingredients such as gobi flavored with soy sauce', '470.00', '606d75ce105d0.jpg');
Insert into dishes values (12, 4, 'Cheese Balls', 'Cheese balls are a mixture of soft cheeses shaped into a ball served with crackers', '500.00', '606d7600dc54c.jpg');
Insert into dishes values (13, 4, 'Potato Twister', 'It is a deep fried spiral-cut whole potato on a skewer, brushed with various seasonings such as onion,cheese,or honey', '450.00', '606d765f69a19.jpg');
Insert into dishes values (14, 4, 'Penne Pasta in makhani gravy', 'Makhani sauce pasta is a delicious indian style fusion pasta recipe cooked in a veg makhani gravy sauce with cream and herbs.', '350.00', '606d768a1b2a1.jpg');
end;
/
```

```
begin
Insert into users values (1, 'kp', 'kanvi', 'Patel', 'kp@gmail.com', '0987654321', 'a32de55ffd7a9c4', 'cg Road');
Insert into users values  (2, 'mj', 'Madhvendra', 'Jhala', 'madhvendra@gmail.com', '1234567890', 'bc28715006af20d', 'Vasna');
Insert into users values  (3, 'mis', 'mishti', 'bhavsar', 'mishti@gmail.com', '2738463984', '58b2318af544351', 'gota');
Insert into users values (4, 'sat', 'satyam', 'desai', 'satyami@gmail.com', '2738463984', '58b2318af5443513', 'shivranjani');
end;
/
```

```
begin
insert into res_category values (1, 'Indian', '2022-04-07 08:45:20');
insert into res_category values  (2, 'Italian', '2022-04-07 08:45:23');
insert into res_category values (3, 'Chinese', '2022-04-07 08:45:25');
insert into res_category values (4, 'American', '2022-04-07 08:45:28');
end;
/
```

```
begin
Insert into restaurant values (1, 1, 'Indie Ka Swad ', 'iks@gmail.com', '4312533432', 'www.iks.com', '12pm', '12am', 'Mon-Sat', 'Paldi', '606d71a81ec5d.jpg', '2022-04-07 23:11:45');
Insert into restaurant values (2, 2, 'Eataly', 'eataly@gmail.com', '0557426406', 'www.eataly.com', '11am', '9pm', 'Mon-Sat', 'Goregaon', '606d720b5fc71.jpg', '2022-04-07 23:11:07');
```

```
Insert into restaurant values (3, 3, 'China Wala Chinese', 'cwc@china.com', '4326538776',
'www.cwc.com', '8am', '9pm', 'Mon-Fri', 'Malad', '606d72653306f.jpg', '2022-04-07 23:11:16');
Insert into restaurant values (4, 4, 'India se America', 'isai@gmail.com', '2342353325', 'www.isa.com',
'9am', '9pm', 'Mon-Sat', 'Lower Parel', '606d72a49503a.jpg', '2022-04-07 23:11:27');
end;
```

PROCEDURES:

1.

```
create or replace procedure your_order(u_i number) as
o number(10);
q number(10);
p number(10,2);
begin
select o_id, quantity, price into o,q,p from users_orders where u_id=u_i;
end;
/
```

2.

```
create or replace procedure desc_restaurants as
r_id number(10);
c_id number(10);
t varchar2(222);
m varchar2(222);
p varchar2(222);
begin
select rs_id,c_id,title,email,phone into r_id,c_id,t,m,p from restaurant;
end;
/
```

3.

```
create or replace procedure new_registration(u_i number,us_n varchar2,f_n varchar2,l_n
varchar2,mail varchar2,phn varchar2,pw varchar2,add varchar2) as
begin
insert into users values( u_i,us_n, f_n, l_n, mail, phn, pw,add);
end;
/
```

4.

```
create or replace procedure add_quantity(u_i varchar2, t varchar2,n number) as
q number(10);
a number(10);
begin
select quantity into q from users_orders;
a:=q+n;
update users_orders set quantity=a where u_id= u_i and title=t;
end;
/
```

5.

```
create or replace procedure delete_dish(dish varchar2) as
begin
delete from dishes where dish=title;
end;
/
```

6.

```
create or replace procedure restaurantwise_menu(rest varchar2) as
r number(10);
t varchar2(222);
s varchar2(222);
```

```
p number(10,2);
i varchar2(222);
begin
select rs_id into r from restaurant where title=rest;
select title,slogan,price,img into t,s,p,i from dishes where r=rs_id;
dbms_output.put_line(t||‘ ’||s||‘ ’||p||‘ ’||i);
end;
/
```

7.

```
create or replace procedure dlt_from_cart(us_id varchar2,name varchar2) as
begin
delete from cart where u_id=us_id and name=title;
end;
/
```

8.

```
create or replace procedure login(us_id varchar2,pw varchar2) as
u varchar2(222);
p varchar2(222);
begin
select username,password into u,p from users;
if (us_id=u and p=pw) then
  dbms_output.put_line(‘Login successfully’);
else
  dbms_output.put_line(‘Login failed’);
end if;
end;
/
```

9.

```
create or replace procedure deleteuser(us_id varchar2) as
Begin
delete from users where us_id=u_id;
End;
/
```

FUNCTION:

```
create or replace function createbill(us_id  number) return int as
t int;
p int;
n varchar2(222);
begin
select sum(quantity*price), title into t,n from users_orders group by title having us_id=u_id;
p:=t;
return p;
end;
/
```

TRIGGERS:

1.

```
create or replace trigger tr_insert after insert on users_orders
for each row
declare
                value int;
begin
        update users_orders o set o.price = value where o.o_id= :new.o_id;
end;
/
```

2.

```
 create or replace trigger restrict_data_manipulation before insert or update or delete on
users_orders
begin
        if(trim(to_char(sysdate, 'day')) not in ('monday', 'tuesday','wednesday', 'thursday','friday')
and
if extract(hour from datetime_value_expression) >= 20) then
                raise_application_error(-20000,'Orders are not taken on weekdays after 8 pm');
        end if;
end;
/
```

3.

```
create or replace trigger review_anlyz after insert on
    reviews
 begin
          if(reviews=’%quality’ and reviews=’%compromise’) then
          delete user.orders;
          raise_application_error(-20050,'Sorry for the inconvenience! Please collect your payment
back from the counter.');
          end if;
end;
/
```

4.

```
create or replace trigger tr_del before delete on users_orders
for each row
begin
        delete from cart where u_id=:old.u_id;
end;
/
```

5.

```
create or replace trigger tr_user after insert on users_orders for each row
declare
d char(10);
Begin
select (d:=trim(to_char(sysdate,'day'));
       if d='monday' or d='tuesday' or d='wednesday' or d='thursday' or d='friday') into dual
  and
extract((hour from sysdate)>=20) then
        if inserting then
           insert into track_user values(:new.u_id,systimestamp,'Insert');
     end if;
 End if;
end;
/
```