**TERM PROJECT**

**UDUPI RESTAURANT MANAGEMENT**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **We declare that we have completed this assignment completely and entirely on our own, without any consultation with others. We have read the UAB Academic Honor Code and understand that any breach of the Honor Code may result in severe penalties.**  **We also declare that the following percentage distribution faithfully represents individual group**  **members’ contributions to the completion of the assignment** | | | | |
| **Name** | **Overall Contribution (%)** | **Major Work Items completed By** | **Signature Or Initials** | **Date** |
| **Pooja Rajeev Maheria** | **33.33** | **Database Schema Design, E R Diagram, Triggers, Stored**  **Procedures, Views, Relational Schema, Constraints, Indexes.** | **Pooja M** | **12/03/2021** |
| **Priyanka Bhauji Wankhede** | **33.34** | **Database Schema Design, E R Diagram, Triggers, Stored**  **Procedures, Views, Relational Schema, Constraints, Indexes.** | **Priyanka W** | **12/03/2021** |
| **Sri Ram Kaushik Karnati** | **33.33** | **Database Schema Design, E R Diagram, Triggers, Stored**  **Procedures, Views, Relational Schema, Constraints, Indexes.** | **Ram K** | **12/03/2021** |

**The web username and passwords for all of us is as below:**

|  |  |  |
| --- | --- | --- |
| NAME | WEBUSERNAME | WEB PASSWORD |
| Pooja Maheria | pmaheriaweb | 382424 |
| Priyanka Bhauji Wankhede | Pw2web | 461994 |
| Sri Ram Kaushik Karnati | skarnatiweb | 123456 |

**Application Requirements**

1. At least 5 entity sets (ES). NOTE: ESs in a IS-A relationship are counted as ONE ES.  
2. There is AT LEAST one IS-A relationship in the E-R diagram.  
3. DBMS server for the project: CS PostgreSQL (do NOT implement this on your home computer/server)

**Deliverables**

**A.1 (10 pts). Application background, requirements (use cases), and assumptions. Assumptions have to be reasonable and do not deviate too much from real world applications. Do NOT use people name as Keys. The applications from the textbook are NOT qualified.**

**Ans:**

1.The employee \_id is given by the authority that we have not taken into picture for our project.

2.The customer orders from a tablet on his/her table and so there is no waiter required to take the order.

3.The manager monitors the chef and chef prepares orders as per the order given by the customer.

4.The manager and chef have different shifts allotted to them by shift\_days.

5.The item to order is a one-to-one relationship as the order can have single item with different quantity at once and the item can be ordered only once with required quantity. If a customer wants to order more than one item, he/she should place a new order with its required quantity and, he will be given a new cust\_id.

6.The customer and order have one to one relationship as one customer can create only one order as per our project. Also, an order corresponds to a unique cust\_id.

7.In short, a customer is expected to place a new order for each item and pay for it where he is allotted a payment\_id to track his payment and keep for future aspects.

8.Each customer on placing an order for an item with the needed quantity, gets a customer id and after he pays, he gets a payment\_id as well.

9.The payment\_id of customers is for storing records for the financial department, who are responsible to see if the payment with the corresponding payment\_id id completed or not, which is not included in this project.

10.The database for customer\_id, payment\_id, mode of payment and amount is kept only for reference for the restaurant. It does not share the customer id with the financial department.

11.The price in order is price/quantity.

12. The amount in payment\_mode is inclusive of all taxes and utility services.

**A.2 (30 points). E-R diagram (with detailed explanation of assumptions and database constraints)**

**Ans:**

**Diagram

Description automatically generated**

**B (15 points). Relational Schema – must be in 3NF (you need to formally prove it). The actual SQL create table commands are required. Include primary key and referential integrity constraints.**

**Ans:**

🡪 Each table is in 2NF and no table have transitive dependency (The Non-prime attribute of any table cannot uniquely determine the other attribute).Hence all the relations are in 3NF

CREATE TABLE Manager (

Manager\_id varchar(5) PRIMARY KEY,

Shift\_Day varchar(20) NOT NULL,

Name varchar(30));

CREATE TABLE Chef(

Chef\_id varchar(5) PRIMARY KEY,

Shift\_Day varchar(20) NOT NULL,

Name varchar(30));

CREATE TABLE Item(

Item\_id varchar(5) PRIMARY KEY,

Name varchar(30) UNIQUE,

Calories int,

Price int,

Spicy\_Level SL);

CREATE TABLE Payment\_Mode(

Payment\_id varchar(5) PRIMARY KEY,

Mode M,

Amount int);

CREATE TABLE Orders(

Order\_id varchar(5) PRIMARY KEY,

Quantity int,

Order\_item\_id varchar(5),

Order\_Cust\_id varchar(5),

FOREIGN KEY (Order\_item\_id) REFERENCES Item(Item\_id),

FOREIGN KEY(Order\_Cust\_id) REFERENCES Customer(Cust\_id));

CREATE TABLE Customer(

Cust\_id varchar(5) PRIMARY KEY,

Name varchar(30),

City varchar(30),

Mobile\_No int NOT NULL);

Manger and chef:

A screenshot of a computer screen

Description automatically generated

Customer:

Text

Description automatically generated

Order:

Text

Description automatically generated

Items:



Text

Description automatically generated

Payment\_mode:



Text

Description automatically generated

Tables:

Text

Description automatically generated

Text

Description automatically generated

**C (15 points). Sample Data Populate database with sample data. Insert at least 10 tuples per relation. Make sure that all queries from Deliverable D (below) have a non-empty answer.**

Sample data for Managers, Chefs, Orders:

Text

Description automatically generated

Sample data for Item, Customer, Payment\_Mode:

Graphical user interface, text

Description automatically generated

**D (15 points). Create Views Create (write English descriptions and SQL syntax) at least 6 views that a user of the database would find useful. Do NOT include views that involve one single relation only. Attach screenshots of the query results from each created view, without them 0 points will be awarded.**

View1:

To get frequently visiting customer information from “ORDERS” and “CUSTOMERS”:

**CREATE VIEW Freq\_Cust AS SELECT C.CUST\_ID, C.NAME, C.MOBILE\_NO, O.ORDER\_ID FROM ORDERS O , CUSTOMER C WHERE O.ORDER\_CUST\_ID = C.CUST\_ID;**

Text

Description automatically generated

View2:

To get most ordered item information and calorie count from “ORDER” and “ITEM”:

**CREATE VIEW most\_ordered AS SELECT I.ITEM\_ID, I.NAME, I.CALORIES FROM ORDERS O , ITEM I WHERE O.ORDER\_ITEM\_ID = I.ITEM\_ID;**

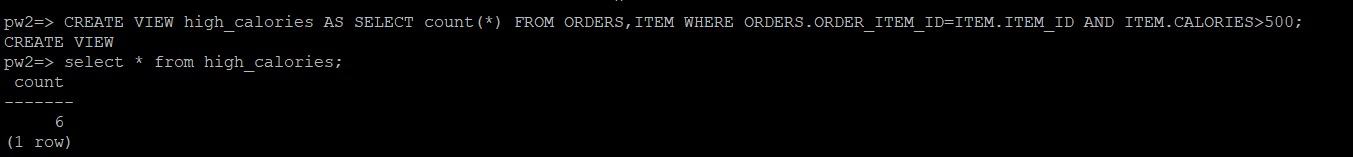
Text

Description automatically generated

View3:

To get the information of the food which has calories higher than 500.

**CREATE VIEW high\_calories AS SELECT count(\*) FROM ORDERS , ITEM WHERE ORDERS.ORDER\_ITEM\_ID = ITEM.ITEM\_ID AND ITEM.CALORIES>500;**



View4:

Text

Description automatically generated

View5:

To get the average quantity of items :

**CREATE VIEW AVGQ AS SELECT avg(QUANTITY) FROM ORDERS, ITEM WHERE ORDERS.ORDER\_ITEM\_ID = ITEM.ITEM\_ID;**

Text

Description automatically generated

View6:

Text

Description automatically generated

**E (10 points). Indexes Create 3 indexes to support some of the queries (each view corresponds to a query) of Deliverable D. For each index, briefly explain (one sentence) how it will support a query (or queries). Do NOT create indexes for primary keys.**

Index1:



Index2:



Index3 & Index4:

Text

Description automatically generated with medium confidence

Index5:



**F (10 points). Constraints Create at least 3 non-key/non-foreign-key constraints on the database. Do not create unique or not null constraints for the primary key.**

**Ans:**

Customers and orders constraint:

Text

Description automatically generated

Item and payment\_mode constraint:

Text

Description automatically generated

Manager and Chef constraint:

Shape

Description automatically generated with medium confidence

**G (15 points). Triggers Create at least 2 Triggers on the database. Should you need to implement triggers to enforce constraints, you should choose to implement such trigger functions that cannot be readily replaced by existing constraint mechanisms provided by DBMS (e.g., do NOT create triggers that simply return an error message when there is a violation of the domain constraints, not null, unique, or a check constraint.) Include in the submission a statement of purpose for each trigger, the create trigger and trigger function statements, and the screenshots showing an actual triggering process as well as the results afterwards (you may also need to show the tuples in the table to show that the triggering condition is met.) At least 50% of points will be deducted if no such screenshots submitted.**

**Ans:**

Trigger1:

When a customer’s mobile number is updated in the customer table then we need to update that customer’s registered mobile number in Payment table.

**Text

Description automatically generated**

Trigger2:

When price of an item is revised in the item table then we need to display the same in order table for that corresponding item.

Text

Description automatically generated

Text

Description automatically generated

**H (Bonus 5 points). Stored Procedures Create at least 1 stored procedure on the database that is NOT tied to a trigger.**

**Ans:**

The stored procedure we have implemented is a function calculating the sum of the amount paid to the restaurant for all the orders.

Shape

Description automatically generated with medium confidence