

**COURSE PROJECT REPORT**

**Design Phase Report**

**Title-Restaurant Management System**

**TEAM MEMBERS DETAILS:**

|  |  |  |
| --- | --- | --- |
| **NAME** | **USN** | **ROLL NO** |
| GAYATRI B | 01FE21BCS029 | 304 |
| GAURI T | 01FE21BCS098 | 314 |
| KRISHNA H | 01FE21BCS100 | 315 |
| AISHWARYA P | 01FE21BCS112 | 318 |

**RESPONSIBILITIES:**

|  |  |
| --- | --- |
| RESPONSIBILITIES | NAME |
| E-R MODEL | AISHWARYA, KRISHNA, GAYATRI, GAURI |
| RELATIONAL MAPPING | GAYATRI AND GAURI |
| INSERTION OF VALUES & NORMALIZATION | AISHWARYA AND KRISHNA |
| REPORT | GAURI, GAYATRI, KRISHNA, AISHWARYA |
| QUERIES | GAYATRI, GAURI, KRISHNA, AISHWARYA |

**PROBLEM DESCRIPTION:**

The Restaurant Management System is a web application. This system is to automate day to day activity of a restaurant. Restaurant is a kind of business that serves people all over world with ready-made food. This system is to provide service facility to restaurant and also to the customer. This restaurant management system can be used by employees in a restaurant to handle the clients, their orders and can help them easily find free tables or place orders. The services that are provided is food ordering and reservation table management by the customer through the system, customer information management and waiter information management, menu information management and report. After a successful login the customer can access the menu page with the items listed according to the desired time. The main point of this system is to help restaurant administrator manage the restaurant business and help customer for online ordering and reserve table. The user can search for a menu according to his choice i.e. according to price range and category of food and later he can order a meal.

**REQUIREMENTS:**

**Manager:**

* Should be able to update information about my restaurant.
* Should be able to view weekly sales for my outlet.
* Should be able to view the current inventory position of the restaurant.

**Customer:**

* Should be able to browse through the menu and look at the various food options available in the restaurant along with the price for each item.
* Should be able to select items from the menu and add them to my order.

**Chef:**

* Should be able to view the current queue to see the ready orders for pickup.
* Should be able to mark the orders as picked up on the common queue.
* Should be able to update the order status to ready once the order is prepared.

**Waiter:**

* Should be able to view the current queue to see the ready orders for pickup.
* Should be able to mark the orders as picked up on the common queue.

**Supplier:**

* Should be able to view the inventory items so as to supply the lacking ingredients to the chef.
* Should be able to update the food items as available when supplied.
* Should be able to update information about the ingredients and some of their advantages when consumed.

**QUESTIONS TO BE ANSWERED:**

**Question 1:** From the problem description, identify the entities that need to be represented in the database, the attributes of each entity, the relationships between the entities, and the cardinality ratios of each relationship.

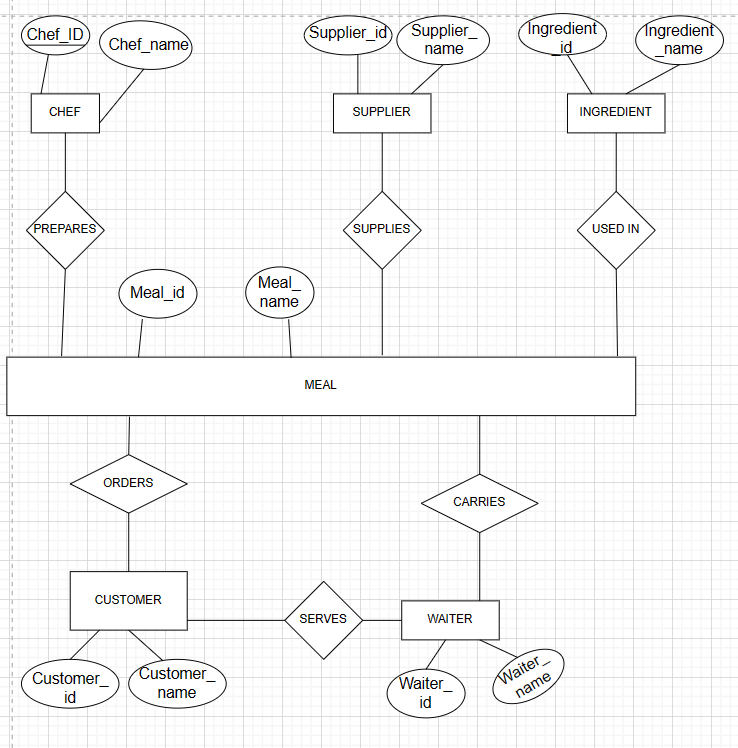
ENTITIES AND ATTRIBUTES:

|  |  |  |
| --- | --- | --- |
| **CHEF** | Chef\_id | Chef\_name |
| **MEAL** | Meal\_ id | Name |
| **ORDERS** | Customer\_customer\_id | Meal\_meal\_id |
| **SUPPLIER** | Supplier\_id | Supplier\_name |
| **WAITER** | Waiter\_id | Waiter\_name |
| **INGREDIENT** | Ingredient\_id | Ingredient \_name |
| **PROVIDES** | Supplier\_id | Ingredient\_id |
| **CUSTOMER** | Customer\_id | Customer\_name |

RELATIONSHIP BETWEEN ENTITIES AND CARDINALITY RATIO:

|  |  |  |  |
| --- | --- | --- | --- |
| **ENTITY 1** | **RELATION NAME** | **ENTITY 2** | **CARDINALITY RATIO** |
| SUPPLIER | SUPPLIES TO | CHEF | N:1 |
| SUPPLIER | PROVIDES | INGREDIENT | M:N |
| MEAL | CONSISTS OF | INGREDIENT | N:1 |
| CUSTOMER | ORDERS | MEAL | M:N |
| WAITER | ATTENDS TO | CUSTOMER | 1:N |

**Question 2:** Draw an Entity-Relationship Diagram illustrating the information you have identified in Question 1



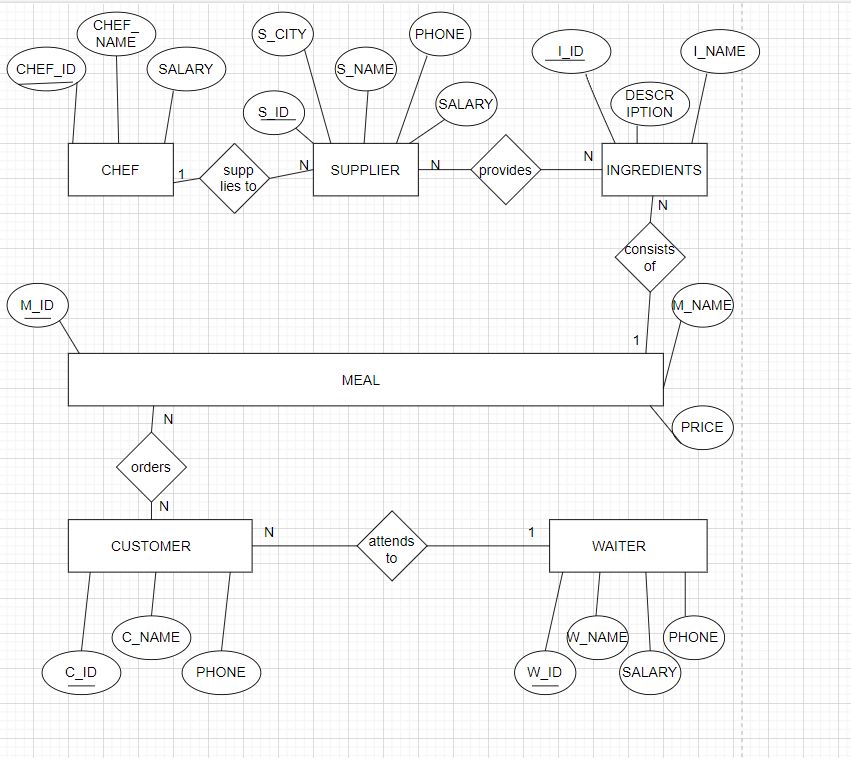
**Question 3:** Draw an alternate Entity-Relationship Diagram illustrating the information you have identified in Question 1 that you think is most likely to occur.

a) We have identified some more entities which are important and related to the system.

b) We have added some more attributes to the previous entities.

c) This is the optimal one because it has almost all the information which the system needs for any information

d) We have mostly covered all the information which will be needed



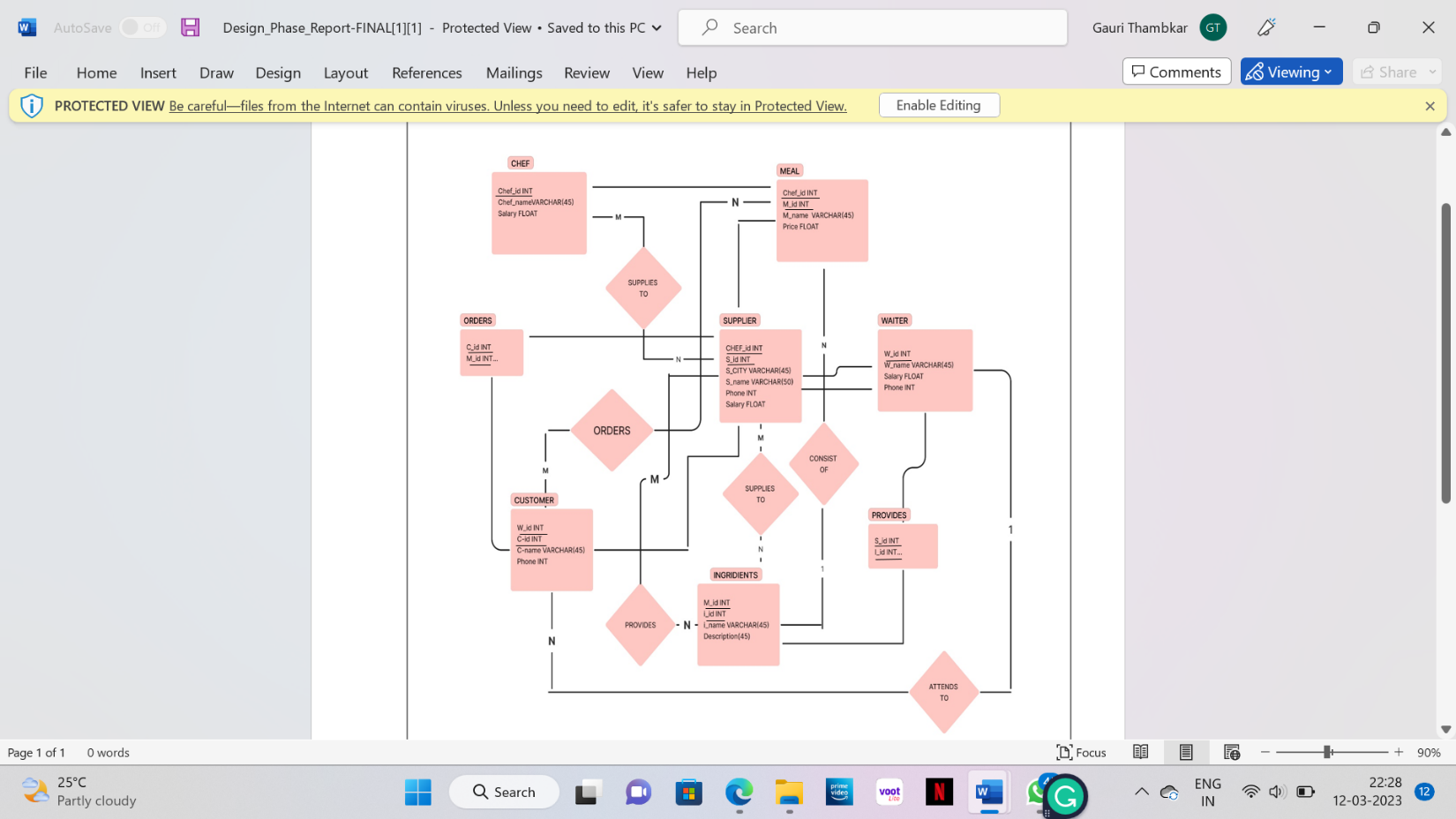
Entity and Attributes

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **CHEF** | Chef\_id | Chef\_name | Salary |  |  |
| **MEAL** | Meal\_ id | Meal\_name | Price |  |  |
| **SUPPLIER** | Supplier\_id | Supplier\_name | Supplier\_city | Phone | Salary |
| **WAITER** | Waiter\_id | Waiter\_name | Salary | Phone |  |
| **INGREDIENT** | Ingredient\_id | Ingredient \_name | Description |  |  |
| **CUSTOMER** | Customer\_id | Customer\_name | Phone |  |  |

RELATIONSHIP BETWEEN ENTITIES AND CARDINALITY RATIO:

|  |  |  |  |
| --- | --- | --- | --- |
| **ENTITY 1** | **RELATION NAME** | **ENTITY 2** | **CARDINALITY RATIO** |
| SUPPLIER | SUPPLIES TO | CHEF | N:1 |
| SUPPLIER | PROVIDES | INGREDIENT | M:N |
| MEAL | CONSISTS OF | INGREDIENT | N:1 |
| CUSTOMER | ORDERS | MEAL | M:N |
| WAITER | ATTENDS TO | CUSTOMER | 1:N |

**Question 4:** Draw an object model illustrating the information you have identified in Question 3.



**Question 5**: Draw an ER to Relation Mapping illustrating the information you have identified

|  |  |  |  |
| --- | --- | --- | --- |
| CHEF | | | |
| Chef\_id | Chef\_name | Salary |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| SUPPLIER | | | | | |
| Chef\_id | **S\_id** | S\_name | S\_city | Phone | Salary |

|  |  |  |  |
| --- | --- | --- | --- |
| MEAL | | | |
| Chef\_ id | **M\_id** | M\_name | Price |

|  |  |  |  |
| --- | --- | --- | --- |
| INGREDIENT | | | |
| M\_id | **I\_id** | I\_name | Description |

|  |  |  |  |
| --- | --- | --- | --- |
| WAITER | | | |
| W\_id | W\_name | Salary | Phone |

|  |  |  |  |
| --- | --- | --- | --- |
| CUSTOMER | | | |
| W\_id | **C\_id** | C\_name | Phone |

|  |  |
| --- | --- |
| ORDERS | |
| C\_id | M\_id | |

|  |  |
| --- | --- |
| PROVIDES | |
| S\_id | I\_id | |

**Question 6:** Draw a Data Dictionary illustrating the information you have identified in Question 5.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **OBJECT**  **(ENTITY)** | **NAME**  **(ATTRIBUTE)** | **TYPE**  **(DATA TYPE)** | **DESCRIPTION** | **PRIMARY KEY** | **FOREIGN KEY** |
| CHEF | CHEF\_ID | INTEGER | UNIQUE IDENTIFICATION FOR EVERY CHEF | YES | NO |
| CHEF\_NAME | VARCHAR | NAME OF CHEF | NO | NO |
| SALARY | FLOAT | SALARY OF CHEF | NO | NO |
|  |  |  |  |  |  |
| SUPPLIER | S\_ID | INTEGER | UNIQUE IDENTIFICATION FOR SUPPLIER | YES | NO |
| S\_NAME | VARCHAR | NAME OF SUPPLIER | NO | NO |
| S\_CITY | VARCHAR | CITY OF THE SUPPLIER | NO | NO |
|  | SALARY | FLOAT | SALARY OF THE SUPPLIER | NO | NO |
|  | PHONE | INTEGER | PHONE NUMBER OF SUPPLIER | NO | NO |
|  | CHEF\_ID | INTEGER | UNIQUE IDENTIFICATION FOR EVERY CHEF | NO | YES |
|  |  |  |  |  |  |
| MEAL | M\_ID | INTEGER | UNIQUE IDENTIFICATION FOR EACH MEAL | YES | NO |
| M\_NAME | VARCHAR | NAME OF THE MEAL | NO | NO |
| PRICE | FLOAT | PRICE OF THE MEAL | NO | NO |
| CHEF\_ID | INTEGER | UNIQUE IDENTIFICATION FOR CHEF | NO | YES |
|  |  |  |  |  |  |
| INGREDIENTS | I\_ID | INTEGER | UNIQUE IDENTIFICATION FOR INGREDIENTS | YES | NO |
| I\_NAME | VARCHAR | NAME OF THE INGREDIENT | NO | NO |
| DESCRIPTION | VARCHAR | QUANTITY OF THE INGREDIENT | NO | NO |
|  | M\_ID | INTEGER | UNIQUE IDENTIFICATION FOR EACH MEAL | NO | YES |
|  |  |  |  |  |  |
| WAITER | W\_ID | INTEGER | UNIQUE IDENTIFICATION FOR WAITER | YES | NO |
|  | W\_NAME | VARCHAR | NAME OF THE WAITER | NO | NO |
|  | SALARY | FLOAT | SALARY OF THE WAITER | NO | NO |
|  | PHONE | INTEGER | PHONE NUMBER OF THE WAITER | NO | NO |
|  |  |  |  |  |  |
| CUSTOMER | C\_ID | INTEGER | UNIQUE IDENTIFICATION FOR CUSTOMER | YES | NO |
| C\_NAME | VARCHAR | NAME OF CUSTOMER | NO | NO |
| PHONE | INT | PHONE NUMBER OF CUSTOMER | NO | NO |
|  | W\_ID | INTEGER | UNIQUE IDENTIFICATION FOR WAITER | NO | YES |
|  |  |  |  |  |  |
| ORDERS | C\_ID | INTEGER | UNIQUE IDENTIFICATION FOR CUSTOMER | NO | YES |
|  | M\_ID | INTEGER | UNIQUE IDENTIFICATION FOR MEAL | NO | YES |
|  |  |  |  |  |  |
| PROVIDES | S\_ID | INTEGER | UNIQUE IDENTIFICATION FOR SUPPLIER | NO | YES |
|  | I\_ID | INTEGER | UNIQUE IDENTIFICATION FOR INGREDIENTS | NO | YES |
|  |  |  |  |  |  |

**Question 7**: **Normalization**: Are all the relations in your chosen schema in 3NF? Are they in BCNF? Explain your answers. If any of your relations are not in BCNF, normalize them to BCNF. If you choose to normalize your relations only till 2NF or 3NF, explain your reasons (e.g., the amount of redundancy introduced is limited or some other valid reason).

1.CHEF

(CHEF\_ID#, CHEF\_NAME, SALARY)

⇒ The Relation is in 1NF as it has no attributes that can hold only multiple values. The attributes in the relation can hold only atomic values.

⇒ The Relation is in 2NF as the primary key contains only one attribute. Hence, every attribute is fully functionally dependent on the key.

⇒ We observe that there is no transitivity in functional dependencies for the given relation. Hence the relation is in 3NF.

⇒ Similarly, it is in BCNF as there exists no non-key attribute that determines another non key attribute.

2.SUPPLIER

(S\_ID#, S\_CITY, S\_NAME, PHONE, CHEF\_ID)

⇒ The Relation is in 1NF as it has no attributes that can hold only multiple values. The attributes in the relation can hold only atomic values.

⇒ The Relation is in 2NF as the primary key contains only one attribute. Hence, every attribute is fully functionally dependent on the key.

⇒ We observe that there is no transitivity in functional dependencies for the given relation. Hence the relation is in 3NF.

⇒ Similarly, it is in BCNF as there exists no non-key attribute that determines another non key attribute.

3.MEAL

(M\_ID#, M\_NAME, PRICE, CHEF\_ID)

⇒ The Relation is in 1NF as it has no attributes that can hold only multiple values. The attributes in the relation can hold only atomic values.

⇒ The Relation is in 2NF as the primary key contains only one attribute. Hence, every attribute is fully functionally dependent on the key.

⇒ We observe that there is no transitivity in functional dependencies for the given relation. Hence the relation is in 3NF.

⇒ Similarly, it is in BCNF as there exists no non-key attribute that determines another non key attribute.

4.INGREDIENTS

(I\_ID#, I\_NAME, DESCRIPTION, M\_ID)

⇒ The Relation is in 1NF as it has no attributes that can hold only multiple values. The attributes in the relation can hold only atomic values.

⇒ The Relation is in 2NF as the primary key contains only one attribute. Hence, every attribute is fully functionally dependent on the key.

⇒ We observe that there is no transitivity in functional dependencies for the given relation. Hence the relation is in 3NF.

⇒ Similarly, it is in BCNF as there exists no non-key attribute that determines another non key attribute.

5.WAITER

(W\_ID#, W\_NAME, SALARY, PHONE)

⇒ The Relation is in 1NF as it has no attributes that can hold only multiple values. The attributes in the relation can hold only atomic values.

⇒ The Relation is in 2NF as the primary key contains only one attribute. Hence, every attribute is fully functionally dependent on the key.

⇒ We observe that there is no transitivity in functional dependencies for the given relation. Hence the relation is in 3NF.

⇒ Similarly, it is in BCNF as there exists no non-key attribute that determines another non key attribute.

6.CUSTOMER

(C\_ID#, C\_NAME, PHONE, W\_ID)

⇒ The Relation is in 1NF as it has no attributes that can hold only multiple values. The attributes in the relation can hold only atomic values.

⇒ The Relation is in 2NF as the primary key contains only one attribute. Hence, every attribute is fully functionally dependent on the key.

⇒ We observe that there is no transitivity in functional dependencies for the given relation. Hence the relation is in 3NF.

⇒ Similarly, it is in BCNF as there exists no non-key attribute that determines another non key attribute.

7.ORDERS

(C\_ID, M\_ID)

⇒ This Relation was a Multivalued attribute in the customer relation due to which it was not satisfying 1NF, thus we removed that multivalued attribute from the Customer Relation and created Orders Relation with the Primary key of the Customer relation and Meal relation.

⇒ The Relation is in 1NF as it has no attributes that can hold only multiple values. The attributes in the relation can hold only atomic values.

⇒ The Relation is in 2NF since the non-Prime attribute is fully functionally dependent on the Primary Key.

⇒ We observe that there is no transitivity in functional dependencies for the given relation. Hence the relation is in 3NF.

8.PROVIDES

(S\_ID, I\_ID)

⇒ This Relation was a Multivalued attribute in the customer relation due to which it was not satisfying 1NF, thus we removed that multivalued attribute from the Supplier Relation and created Provides Relation with the Primary key of the Supplier relation and Ingredients relation.

⇒ The Relation is in 1NF as it has no attributes that can hold only multiple values. The attributes in the relation can hold only atomic values.

⇒ The Relation is in 2NF as the primary key contains only one attribute. Hence, every attribute is fully functionally dependent on the key.

⇒ We observe that there is no transitivity in functional dependencies for the given relation. Hence the relation is in 3NF.

⇒ Similarly, it is in BCNF as there exists no non-key attribute that determines another non key attribute.

**Question 8**: Choose the optimal normalized schema from Question 9 and justify why you think this is an optimal solution.

We have removed the redundant entities from the relations and we have normalized the relations till 3NF for most of the relations. For some relations we have normalized till 1NF or 2NF because of some constraints.

|  |  |
| --- | --- |
| CHEF | (CHEF\_ID#, CHEF\_NAME, SALARY) |
| SUPPLIER | (S\_ID#, S\_CITY, S\_NAME, PHONE, SALARY, CHEF\_ID) |
| MEAL | (M\_ID#, M\_NAME, PRICE, CHEF\_ID) |
| INGREDIENTS | (I\_ID#, I\_NAME, DESCRIPTION, M\_ID) |
| WAITER | (W\_ID#, W\_NAME, SALARY, PHONE) |
| CUSTOMER | (C\_ID#, C\_NAME, PHONE, W\_ID) |
| ORDERS | (C\_ID, M\_ID) |
| PROVIDES | (S\_ID, I\_ID) |

REFERENCES:

1. Elmasri R. and Navathe S., Fundamentals of Database Systems, 7thedition, Pearson Education, 2016.
2. Draw.io tool to draw all the E R Diagram, Relational Schema and Object model illustration.

**Implementation Phase Report:**

**Creation Of Tables:**

1)Table Name: CHEF

create table chef (

chef\_id int constraint pk\_chef primary key,

chef\_name varchar (45),

salary float

);

2) Table Name: SUPPLIER

create table supplier (

chef\_id int,

s\_id int constraint pk\_supplier primary key,

s\_city varchar (45),

s\_name varchar (50),

phone int,

salary float,

foreign key (chef\_id) references chef(chef\_id)

);

3) Table Name: MEAL

create table meal (

chef\_id int,

m\_id int constraint pk\_meal primary key,

m\_name varchar (45),

price float,

foreign key (chef\_id) references chef(chef\_id)

);

4) Table Name: INGREDIENTS

create table ingredients (

m\_id int,

i\_id int constraint pk\_ingredients primary key,

i\_name varchar (45),

description varchar (45),

foreign key (m\_id) references meal(m\_id)

);

5) Table Name WAITER

create table waiter (

w\_id int constraint pk\_waiter primary key,

w\_name varchar (45),

salary float,

phone int

);

6)Table Name: CUSTOMER

create table customer (

w\_id int,

c\_id int constraint pk\_customer primary key,

c\_name varchar (45),

phone int,

foreign key (w\_id) references waiter(w\_id)

);

7)Table Name: ORDER

create table orders (

c\_id int,

m\_id int,

foreign key (c\_id) references customer(c\_id),

foreign key (m\_id) references meal(m\_id)

);

8) Table Name: PROVIDES

create table provides (

s\_id int,

i\_id int,

foreign key (i\_id) references ingredients(i\_id),

foreign key (s\_id) references supplier(s\_id)

);

**Inserting Values into tables:**

1)Table Name: CHEF

insert into chef values('1','BILL','5000');

insert into chef values('2','SINCHAN','30000');

insert into chef values('3','ABHIJNA','35000');

insert into chef values('4','RAJANI','21000');

insert into chef values('5','KATYAYANI','22000');

insert into chef values('6','KRISHNA','23000');

insert into chef values('7','ABHISHEK','24000');

insert into chef values('8','AKSHATHA','25000');

insert into chef values('9','GAURI','26000');

insert into chef values('10','GAYATRI','27000');

insert into chef values('11','PUNEET','28000');

insert into chef values('12','SATWIK','29000');

insert into chef values('13','ABHINAV','31000');

insert into chef values('14','AISHWARYA','32000');

insert into chef values('15','CHETAN','50000');

insert into chef values('16','JEEVAN','34000');

insert into chef values('17','RUBEENA','18000');

insert into chef values('18','PRANAV','19000');

insert into chef values('19','ESHAAN','17000');

insert into chef values('20','ANIRUDH','15000');

2) Table Name: SUPPLIER

insert into supplier values('1','100','HUBLI','JOHN','9445789987','10000');

insert into supplier values('2','110','DHARWAD','SMITH','9445789986','15000');

insert into supplier values('3','120','DANDELI','BILL','9445789985','20000');

insert into supplier values('4','130','DAVANGERE','EMILY','9445789981','25000');

insert into supplier values('5','140','BANGALORE','CHANDLER','9446789981','30000');

insert into supplier values('6','150','MANGALURU','MONIKA','9445789983','31000');

insert into supplier values('7','160','UDAPI','XAVIER','9445789984','25000');

insert into supplier values('8','170','MANIPAL','ROSS','9445789982','16000');

insert into supplier values('9','180','MYSORE','RACHEL','9445989917','20000');

insert into supplier values('10','190','DELHI','RAMESH','9445782987','34000');

insert into supplier values('11','200','HUBLI','TOM','9442789987','11000');

insert into supplier values('12','210','DHARWAD','JERRY','9445779987','14000');

insert into supplier values('13','220','DANDELI','KEVIN','9445719987','19000');

insert into supplier values('14','230','DAVANGERE','EMMA','9445789937','24000');

insert into supplier values('15','240','BANGALORE','ISABELLE','9445789187','31000');

insert into supplier values('16','250','MANGALORE','OLIVIA','8445789987','26000');

insert into supplier values('17','260','UDAPI','ELIJAH','9427789987','27000');

insert into supplier values('18','270','MANIPAL','ELLA','9444789987','17000');

insert into supplier values('19','280','MYSURE','NOAH','9645789987','21000');

insert into supplier values('20','290','DELHI','ELA','9545789987','36000');

insert into supplier values('1','300','HUBLI','JOHN','9445789987','12000');

insert into supplier values('2','310','DHARWAD','SMITH','9445789986','17000');

insert into supplier values('3','320','DANDELI','BILL','9445789985','15000');

insert into supplier values('4','330','DAVANGERE','EMILY','9445789981','23000');

insert into supplier values('5','340','BANGALORE','JOEY','9445789982','32000');

insert into supplier values('6','350','MANGALORE','MONIKA','9445789983','28000');

insert into supplier values('7','360','UDAPI','XAVIER','9445789984','23000');

insert into supplier values('8','370','MANIPAL','ROSS','9445789982','18000');

insert into supplier values('9','380','MYSURU','RACHEL','9445989917','20000');

insert into supplier values('10','390','DELHI','CHLOE','9445782987','33000');

insert into supplier values('11','400','HUBLI','TOM','9442789987','13000');

insert into supplier values('12','410','DHARWAD','JERRY','9445779987','16000');

insert into supplier values('13','420','DANDELI','KEVIN','9445719987','14000');

insert into supplier values('14','430','DAVANGERE','EMMA','9445789937','24000');

insert into supplier values('15','440','BANGALORE','ISABELLE','9445789187','31000');

insert into supplier values('16','450','MANGALORE','OLIVIA','8445789987','29000');

insert into supplier values('17','460','UDAPI','ELIJAH','9427789987','22000');

insert into supplier values('18','470','MANIPAL','ELLA','9444789987','25000');

insert into supplier values('19','480','MYSURU','NOAH','9645789987','20000');

insert into supplier values('20','490','DELHI','SURESH','9545789987','34000');

3) Table Name: Series

insert into meal values ('2','1','MANCHOW SOUP','45');

insert into meal values ('2','2','MIXED VEG CLEAR SOUP','40');

insert into meal values ('3','3','TOMATO SOUP','55');

insert into meal values ('4','4','BOTTLE GAURD AND ONION SOUP','70');

insert into meal values ('5','5','BABY CORN SOUP','50');

insert into meal values ('6','6','LEMON AND CORIANDER SOUP','40');

insert into meal values ('7','7','ALOO PARATHA','50');

insert into meal values ('8','8','POORI BHAJI','70');

insert into meal values ('9','9','VEG MANCHURIAN','100');

insert into meal values ('10','10','GOBI MANCHURIAN','100');

insert into meal values ('11','11','BABY CORN MANCHURIAN','120');

insert into meal values ('12','12','PANEER MANCHURIAN','130');

insert into meal values ('13','13','PANEER CHILY','100');

insert into meal values ('14','14','MASALA PAPAD ','45');

insert into meal values ('15','15','CRISPY VEG','90');

insert into meal values ('16','16','MASHROOM CHILY','120');

insert into meal values ('17','17','FINGER FRIES','75');

insert into meal values ('18','18','PAV BHAJI','65');

insert into meal values ('19','19','DAL FRY','60');

insert into meal values ('20','20','DAL MAKHANI','50');

insert into meal values ('4','21','CHILY PANEER','150');

insert into meal values ('2','22','MALAI KOFTA','120');

insert into meal values ('3','23','ALOO DUM','70');

insert into meal values ('4','24','DAL TADKA','55');

insert into meal values ('5','25','PANEER BUTTER MASALA','130');

insert into meal values ('6','26','TOMATO MASALA','70');

insert into meal values ('7','27','CHANA MASALA','85');

insert into meal values ('8','28','SAHI PANEER','160');

insert into meal values ('9','29','KADAI PANEER','160');

insert into meal values ('10','30','ALOO CHANA','70');

insert into meal values ('11','31','KADI PAKODA','80');

insert into meal values ('12','32','PLAIN NAAN','40');

insert into meal values ('13','33','BUTTER NAAN','50');

insert into meal values ('14','34','LACHAA PARATHA','60');

insert into meal values ('15','35','TANDOORI ROTI','25');

insert into meal values ('16','36','ROOMALI ROTI','30');

insert into meal values ('17','37','PANEER NAAN','80');

insert into meal values ('18','38','GARLIC NAAN','70');

insert into meal values ('19','39','BUTTER ROTI','30');

insert into meal values ('20','40','MIX VEG NAAN','70');

insert into meal values ('8','41','ONION NAAN','80');

insert into meal values ('2','42','PUDINA PARATHA','70');

insert into meal values ('3','43','VEG BIRIYANI','120');

insert into meal values ('4','44','JEERA RICE','100');

insert into meal values ('5','45','CURD RICE','70');

insert into meal values ('6','46','DAL KHICHIDI','110');

insert into meal values ('7','47','VEG HYDRABADI','130');

insert into meal values ('8','48','VEG KASHMIRI PULAV','120');

insert into meal values ('9','49','VEG PULAV','90');

insert into meal values ('10','50','VEG SCHEZWAN RICE','100');

insert into meal values ('11','51','VEG FRIED RICE','95');

4) Table Name: INGREDIENTS

insert into ingredients values ('1','1','BOK CHOY','10KG');

insert into ingredients values ('2','2','SNAKE BEANS','10KG');

insert into ingredients values ('3','3','ROCKET LEAVES','10KG');

insert into ingredients values('4','4','TOMATO','30KG');

insert into ingredients values('5','5','LEMON','100');

insert into ingredients values('6','6','POTATO','50KG');

insert into ingredients values('7','7','ONION','60KG');

insert into ingredients values('8','8','SPINACH','1KG');

insert into ingredients values('9','9','MUSHROOM','10KG');

insert into ingredients values('10','10','CHILI','50KG');

insert into ingredients values ('11','11','SPRING ONION','5KG');

insert into ingredients values ('12','12','GREEN PEAS','10KG');

insert into ingredients values('13','13','GARLIC','5KG');

insert into ingredients values('14','14','CUCUMBUR','20kg');

insert into ingredients values('15','15','RADDISH','5kg');

insert into ingredients values('16','16','CORN','4kg');

insert into ingredients values('17','17','CAULIFLOWER','5kg');

insert into ingredients values('18','18','CARROT','10kg');

insert into ingredients values('19','19','CAPSICUM','10kg');

insert into ingredients values('20','20','BROCOLI','5kg');

insert into ingredients values ('21','21','BOTTLE GUARD','7kg');

insert into ingredients values ('22','22','LADY FINGER','10kg');

insert into ingredients values ('23','23','BELL PEPPER','10kg');

insert into ingredients values('24','24','BEETROOT','10kg');

insert into ingredients values('25','25','CABBAGE','10kg');

insert into ingredients values ('26','26','BABY CORN','10kg');

insert into ingredients values('27','27','GINGER','5kg');

insert into ingredients values('28','28','FLOUR','5kg');

insert into ingredients values('29','29','JOWER','5kg');

insert into ingredients values ('30','30','KIDNEY BEANS','5kg');

insert into ingredients values ('31','31','GREEN GRAM','5kg');

insert into ingredients values ('32','32','BLACK GRAM','5kg');

insert into ingredients values('33','33','RICE','10kg');

insert into ingredients values('34','34','BREAD','3kg');

insert into ingredients values ('35','35','BLACK EYED BEANS','5kg');

insert into ingredients values ('36','36','CHIK PEAS','5kg');

insert into ingredients values ('37','37','BASMATI RICE','20kg');

insert into ingredients values ('38','38','WHEAT FLOUR','10kg');

insert into ingredients values ('39','39','TOOR DAL','5kg');

insert into ingredients values ('40','40','BEAN SPROTS','5kg');

insert into ingredients values('41','41','VINEGER','5kg');

insert into ingredients values('42','42','PANEER','20kg');

insert into ingredients values('43','43','SAUCE','10kg');

insert into ingredients values ('44','44','WHITE PEPPER','5kg');

insert into ingredients values ('45','45','RICE FLOUR','5kg');

insert into ingredients values ('46','46','VEGETABLE OIL','30kg');

insert into ingredients values ('47','47','TARTARIC ACID','3kg');

insert into ingredients values ('48','48','SOYA SAUCE','5kg');

insert into ingredients values('49','49','PAPAD','1kg');

insert into ingredients values('50','50','PAAN','2kg');

insert into ingredients values ('51','51','CORN FLOUR','5kg');

insert into ingredients values ('1','52','BAKING SODA','5kg');

5)Table Name: WAITER

insert into waiter values('1','JAI','10000','9998887766');

insert into waiter values('2','ISHAAN','11000','9998887761');

insert into waiter values('3','AMAR','12000','9998887762');

insert into waiter values('4','DHRUV','13000','9998887763');

insert into waiter values('5','KANAN','14000','9998887764');

insert into waiter values('6','NAYA','10000','9998887765');

insert into waiter values('7','PIYUSH','10000','9998887766');

insert into waiter values('8','SHIVA','11000','9998887767');

insert into waiter values('9','RITHVIK','13000','9998887768');

insert into waiter values('10','ARJUNA','12000','9998887716');

insert into waiter values('11','KARAN','14000','9998887726');

insert into waiter values('12','AARUSH','9000','9998887736');

insert into waiter values('13','NARAYAN','10000','9998887746');

insert into waiter values('14','HRITHIK','12000','9998887756');

insert into waiter values('15','SALMAN','9000','9998887776');

6) Table Name: CUSTOMER

insert into customer values('1','1','AADI','9988776655');

insert into customer values('2','2','AARAV','9988776651');

insert into customer values('3','3','AAYUSH','9988776652');

insert into customer values('4','4','BALVAN','9988776653');

insert into customer values('5','5','CHAITANYA','9988776654');

insert into customer values('6','6','DARSH','9988776656');

insert into customer values('7','7','DEV','9988776657');

insert into customer values('8','8','GAGAN','9988776658');

insert into customer values('9','9','GURMEET','9988776659');

insert into customer values('10','10','HARSH','9988776615');

insert into customer values('11','11','YASH','9988776625');

insert into customer values('12','12','MADHUSUDAN','9988776635');

insert into customer values('13','13','KIRAN','9988776645');

insert into customer values('14','14','SUNIL','9988776665');

insert into customer values('15','15','RAMESH','9988776675');

insert into customer values('1','16','NEEL','9988776685');

insert into customer values('2','17','KIRTI','9988776695');

insert into customer values('3','18','GIRISH','9988776155');

insert into customer values('4','19','ABHINAY','9988776255');

insert into customer values('5','20','GANDHARV','9988776355');

insert into customer values('6','21','MAYUR','9988776455');

insert into customer values('7','22','PRAJWAL','9988776555');

insert into customer values('8','23','SHASHANK','9988776755');

insert into customer values('9','24','SANKET','9988776855');

insert into customer values('10','25','VIVEK','9988776955');

insert into customer values('11','26','LAVNIK','9988771655');

insert into customer values('12','27','NEERAJ','9988772655');

insert into customer values('13','28','BHUSHAN','9988773655');

insert into customer values('14','29','SHANTANU','9988774655');

insert into customer values('15','30','TUSHAR','9988775655');

7) Table Name: ORDERS

insert into orders values ('1','1');

insert into orders values ('2','2');

insert into orders values ('3','3');

insert into orders values ('4','4');

insert into orders values ('5','5');

insert into orders values ('6','6');

insert into orders values ('7','7');

insert into orders values ('8','8');

insert into orders values ('9','9');

insert into orders values ('10','10');

insert into orders values ('11','11');

insert into orders values ('12','12');

insert into orders values ('13','13');

insert into orders values ('14','14');

insert into orders values ('15','15');

insert into orders values ('16','16');

insert into orders values ('17','17');

insert into orders values ('18','18');

insert into orders values ('19','19');

insert into orders values ('20','20');

insert into orders values ('21','21');

insert into orders values ('22','22');

insert into orders values ('23','23');

insert into orders values ('24','24');

insert into orders values ('25','25');

insert into orders values ('26','26');

insert into orders values ('27','27');

insert into orders values ('28','30');

insert into orders values ('29','31');

insert into orders values ('30','32');

insert into orders values ('1','33');

insert into orders values ('2','34');

insert into orders values ('3','35');

insert into orders values ('4','36');

insert into orders values ('5','37');

insert into orders values ('6','38');

insert into orders values( '7','39');

insert into orders values ('8','40');

insert into orders values ('9','41');

insert into orders values ('10','42');

insert into orders values ('11','43');

insert into orders values ('12','44');

insert into orders values ('13','45');

insert into orders values ('14','46');

insert into orders values ('15','47');

insert into orders values ('16','48');

insert into orders values ('17','49');

insert into orders values ('18','50');

insert into orders values ('19','51');

insert into orders values ('20','1');

insert into orders values ('21','2');

insert into orders values ('22','3');

insert into orders values ('23','4');

insert into orders values ('24','5');

insert into orders values ('25','6');

insert into orders values ('26','7');

insert into orders values ('27','8');

insert into orders values ('28','9');

insert into orders values ('29','10');

insert into orders values ('30','11');

8) Table Name: PROVIDES

insert into provides values ('100','1');

insert into provides values ('110','2');

insert into provides values ('120','3');

insert into provides values ('130','4');

insert into provides values ('140','5');

insert into provides values ('150','6');

insert into provides values ('160','7');

insert into provides values ('170','8');

insert into provides values ('180','9');

insert into provides values ('190','10');

insert into provides values ('200','11');

insert into provides values ('210','12');

insert into provides values ('220','13');

insert into provides values ('230','14');

insert into provides values ('240','15');

insert into provides values ('250','16');

insert into provides values ('260','17');

insert into provides values ('270','18');

insert into provides values ('280','19');

insert into provides values ('290','20');

insert into provides values ('300','21');

insert into provides values ('310','22');

insert into provides values ('320','23');

insert into provides values ('330','24');

insert into provides values ('340','25');

insert into provides values ('350','26');

insert into provides values ('360','27');

insert into provides values ('370','28');

insert into provides values ('380','29');

insert into provides values ('390','30');

insert into provides values ('400','31');

insert into provides values ('410','32');

insert into provides values ('420','33');

insert into provides values ('430','34');

insert into provides values ('440','35');

insert into provides values ('450','36');

insert into provides values ('460','37');

insert into provides values ('470','38');

insert into provides values ('480','39');

insert into provides values ('490','40');

insert into provides values ('100','41');

insert into provides values ('110','42');

insert into provides values ('120','43');

insert into provides values ('130','44');

insert into provides values ('140','45');

insert into provides values ('150','46');

insert into provides values ('160','47');

insert into provides values ('170','48');

insert into provides values ('180','49');

insert into provides values ('190','50');

insert into provides values ('200','51');

insert into provides values ('210','52');

insert into provides values ('220','1');

insert into provides values ('230','2');

insert into provides values ('240','3');

insert into provides values ('250','4');

insert into provides values ('260','5');

insert into provides values ('270','6');

insert into provides values ('280','7');

insert into provides values ('290','8');

insert into provides values ('300','9');

insert into provides values ('310','10');

insert into provides values ('320','11');

insert into provides values ('330','12');

insert into provides values ('340','13');

insert into provides values ('350','14');

insert into provides values ('360','15');

insert into provides values ('370','16');

insert into provides values ('380','17');

insert into provides values ('390','18');

insert into provides values ('400','19');

insert into provides values ('410','20');

QUERIES

AGGREGATE FUNCTIONS:

1) CALCULATE THE AVERAGE SALARY OF ALL CHEFS

SELECT AVG(salary) AS avg\_salary

FROM chef;

2) RETRIEVE PRICE OF MEALS WHOSE COUNT IS GREATER THAN 4 IN EACH PRICE.

SELECT price,COUNT(\*)

FROM meal

GROUP BY price

HAVING COUNT(\*)>4;

3)CALCULATE THE TOTAL NUMBER OF ORDERS PLACED BY EACH CUSTOMER.

SELECT c.c\_name, COUNT(\*) AS num\_orders

FROM customer c

JOIN orders o ON c.c\_id = o.c\_id

GROUP BY c.c\_name;

4)FIND THE HIGHEST AND LOWEST PRICES OF ALL MEALS

SELECT MAX(price) AS max\_price, MIN(price) AS min\_price

FROM meal;

5) RETRIEVE THE AVG SALARY AND COUNT OF SUPPLIER IN EACH CITY.

SELECT ROUND(AVG(salary)) as AVG\_SALARY,COUNT(\*)

FROM supplier

GROUP BY s\_city

HAVING AVG(salary)>10000;

COMPLEX QUERIES

1) RETRIEVE THE NAMES OF THE SUPPLIER AND ALSO THE CHEF ID WHOSE SALARY IS MORE THAN 30000.

SELECT DISTINCT s.s\_name,c.chef\_id

FROM supplier s,chef c

WHERE s.chef\_id=c.chef\_id AND c.salary>30000;

2) RETRIEVE THE CUSTOMER NAMES AND ALSO THE WAITER NAMES WHOSE SALARY IS MORE THAN 12000.

SELECT DISTINCT c.c\_name,w.w\_name,salary

FROM customer c,waiter w

WHERE c.w\_id=w.w\_id AND salary>12000;

3) RETRIEVE THE CUSTOMER NAMES WHO ORDERS ALOO PARATHA.

SELECT c.c\_name

FROM customer c,orders o,meal m

WHERE c.c\_id=o.c\_id AND o.m\_id=m.m\_id AND m\_name='ALOO PARATHA';

4) RETRIEVE THE INGREDIENTS ID AND NAMES OF THE MEAL WHOSE WHOSE PRICE IS MORE THAN 100.

SELECT i.i\_id,m.m\_name

FROM ingredients i,meal m

WHERE i.m\_id=m.m\_id AND price>100;

5) RETRIEVE THE SUPPLIER ID AND SUPPLIER NAME WHO PROVIDES WHICH INGREDIENTS.

SELECT s.s\_id,s.s\_name,i.i\_name

FROM supplier s,provides p,ingredients i

WHERE s.s\_id=p.s\_id AND p.i\_id=i.i\_id;

ALL SIX CLAUSES

1)RETRIEVE THE NAMES OF ALL MEALS AND THEIR INGREDIENTS

SELECT m.m\_name, i.i\_name

FROM meal m

JOIN ingredients i ON m.m\_id = i.m\_id;

2)RETRIEVE THE NAME OF EACH CUSTOMER, THE NAME OF EACH MEAL THEY ORDERED, AND THE PRICE OF EACH MEAL OF CUSTOMERS WITH PHONE NUMBERS THAT CONTAIN THE DIGITS '555'.

SELECT c.c\_name, m.m\_name, m.price

FROM customer c

JOIN orders o ON c.c\_id = o.c\_id

JOIN meal m ON o.m\_id = m.m\_id

WHERE c.phone LIKE '%555%';

3) RETRIEVE TOTAL SALARY OF SUPPLIER WHOSE COUNT IS GREATER THAN 2 IN EACH CITY.

SELECT SUM(salary) AS SUM\_SALARY,COUNT(\*),s\_city

FROM supplier

GROUP BY s\_city

HAVING COUNT(\*)>2;

4) RETRIEVE DESCRIPTION OF INGREDIENTS WHOSE COUNT IS GREATER THAN 3 IN EACH DESCRIPTION.

SELECT description,COUNT(\*)

FROM ingredients

GROUP BY description

HAVING count(\*)>3;

5)GROUP THE SUPPLIERS BY CITY AND COUNT THE NUMBER OF SUPPLIERS IN EACH CITY AND SORT THE RESULTS BY THE NUMBER OF SUPPLIERS IN EACH CITY, IN DESCENDING ORDER.

SELECT s.s\_city, COUNT(\*) AS num\_suppliers

FROM supplier s

GROUP BY s.s\_city

ORDER BY num\_suppliers DESC;

SIMPLE NESTED AND CORRELATED QUERIES

1) RETRIEVE THE NAMES OF ALL CHEFS WHO DONT MAKE ANY MEALS.

SELECT \*

FROM chef

WHERE chef\_id NOT IN(SELECT chef\_id

FROM meal);

2) DISPLAY NAMES AND SALARY OF ALL SUPPLIERS IN THE SAME CITY AS SUPPLIER WITH HIGHEST SALARY.

SELECT s\_name,salary,s\_id

FROM supplier

WHERE s\_city=(SELECT s\_city

FROM supplier

WHERE salary=(SELECT MAX(salary)

FROM supplier));

3)RETRIEVE ALL MEALS PROVIDED BY SUPPLIERS LOCATED IN A SPECIFIC CITY

SELECT m.m\_name, s.s\_name

FROM meal m

JOIN ingredients i ON m.m\_id = i.m\_id

JOIN provides p ON i.i\_id = p.i\_id

JOIN supplier s ON p.s\_id = s.s\_id

WHERE s.s\_city = 'HUBLI';

4)RETRIEVE THE NAME OF THE CHEF WHO PREPARES THE MOST EXPENSIVE MEAL

SELECT chef\_name

FROM chef c

WHERE c.chef\_id in (

SELECT chef\_id

FROM meal

WHERE price = (

SELECT MAX(price)

FROM meal

)

);

5)RETRIEVE ALL CUSTOMERS WHO HAVE PLACED ORDERS FOR MEALS PREPARED BY A SPECIFIC CHEF

SELECT c.c\_name

FROM customer c

JOIN orders o ON c.c\_id = o.c\_id

JOIN meal m ON o.m\_id = m.m\_id

WHERE m.chef\_id = (

SELECT chef\_id

FROM chef

WHERE chef\_name = 'ABHINAV'

);

PLSQL

1.RETRIEVE ALL CHEF NAMES AND THEIR SALARIES

DECLARE

v\_chef\_name chef.chef\_name%TYPE;

v\_salary chef.salary%TYPE;

BEGIN

FOR chef\_rec IN (SELECT chef\_name, salary FROM chef) LOOP

v\_chef\_name := chef\_rec.chef\_name;

v\_salary := chef\_rec.salary;

DBMS\_OUTPUT.PUT\_LINE('Chef Name: ' || v\_chef\_name || ', Salary: ' || v\_salary);

END LOOP;

END;

/

2.RETRIEVE ALL MEALS WITH THEIR PRICES AND THE NAME OF THE CHEF WHO CREATED THEM

DECLARE

v\_chef\_name chef.chef\_name%TYPE;

v\_m\_name meal.m\_name%TYPE;

v\_price meal.price%TYPE;

BEGIN

FOR meal\_rec IN (SELECT chef.chef\_name, meal.m\_name, meal.price FROM meal

JOIN chef ON meal.chef\_id = chef.chef\_id) LOOP

v\_chef\_name := meal\_rec.chef\_name;

v\_m\_name := meal\_rec.m\_name;

v\_price := meal\_rec.price;

DBMS\_OUTPUT.PUT\_LINE('Meal: ' || v\_m\_name || ', Price: ' || v\_price || ', Chef: ' || v\_chef\_name);

END LOOP;

END;

/

3.RETRIEVE ALL CHEFS FROM THE 'chef' TABLE

DECLARE

TYPE chef\_rec IS RECORD (

chef\_id chef.chef\_id%TYPE,

chef\_name chef.chef\_name%TYPE,

salary chef.salary%TYPE

);

chef\_record chef\_rec;

CURSOR chef\_cursor IS SELECT \* FROM chef;

BEGIN

OPEN chef\_cursor;

LOOP

FETCH chef\_cursor INTO chef\_record;

EXIT WHEN chef\_cursor%NOTFOUND;

DBMS\_OUTPUT.PUT\_LINE(chef\_record.chef\_id || ' ' || chef\_record.chef\_name || ' ' || chef\_record.salary);

END LOOP;

CLOSE chef\_cursor;

END;

/

4.AUTOMATICALLY UPDATE THE SALARY OF A SUPPLIER WHEN THEY PROVIDE INGREDIENTS FOR A MEAL WITH A HIGH PRICE

CREATE OR REPLACE TRIGGER update\_supplier\_salary

AFTER INSERT ON provides

FOR EACH ROW

DECLARE

v\_m\_id provides.m\_id%TYPE := :NEW.m\_id;

v\_price meal.price%TYPE;

v\_supplier\_salary supplier.salary%TYPE;

BEGIN

SELECT price INTO v\_price FROM meal WHERE m\_id = v\_m\_id;

SELECT salary INTO v\_supplier\_salary FROM supplier WHERE s\_id = :NEW.s\_id;

IF v\_price > 50 AND v\_supplier\_salary < 1000 THEN

UPDATE supplier SET salary = salary \* 1.1 WHERE s\_id = :NEW.s\_id;

DBMS\_OUTPUT.PUT\_LINE('Supplier salary updated');

END IF;

END;

/

5.RETRIEVE ALL MEALS FOR A SPECIFIC CHEF

DECLARE

TYPE meal\_rec IS RECORD (

chef\_id meal.chef\_id%TYPE,

m\_id meal.m\_id%TYPE,

m\_name meal.m\_name%TYPE,

price meal.price%TYPE

);

meal\_record meal\_rec;

CURSOR meal\_cursor (p\_chef\_id IN NUMBER) IS SELECT \* FROM meal WHERE chef\_id = p\_chef\_id;

BEGIN

OPEN meal\_cursor(1);

LOOP

FETCH meal\_cursor INTO meal\_record;

EXIT WHEN meal\_cursor%NOTFOUND;

DBMS\_OUTPUT.PUT\_LINE(meal\_record.chef\_id || ' ' || meal\_record.m\_id || ' ' || meal\_record.m\_name || ' ' || meal\_record.price);

END LOOP;

CLOSE meal\_cursor;

END;

/

6.CALCULATE THE TOTAL REVENUE GENERATED FROM ALL ORDERS

CREATE OR REPLACE FUNCTION calculate\_revenue RETURN NUMBER IS

v\_revenue NUMBER := 0;

BEGIN

FOR order\_rec IN (SELECT price FROM meal JOIN orders ON meal.m\_id = orders.m\_id) LOOP

v\_revenue := v\_revenue + order\_rec.price;

END LOOP;

RETURN v\_revenue;

END;

/

7.CALCULATE THE TOTAL COST OF ALL INGREDIENTS FOR A SPECIFIC MEAL

CREATE OR REPLACE PROCEDURE calculate\_ingredients\_cost (

p\_m\_id IN ingredients.m\_id%TYPE,

p\_total\_cost OUT NUMBER

) IS

BEGIN

SELECT SUM(cost) INTO p\_total\_cost FROM (

SELECT i.cost \* i.quantity AS cost FROM ingredients i

JOIN meal m ON i.m\_id = m.m\_id

WHERE i.m\_id = p\_m\_id

);