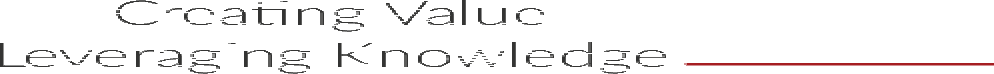
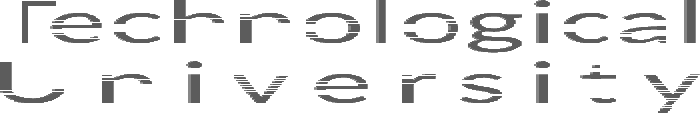
Earlier knownas



B. V. B. College of Engineering &Technology

School of Computer Science and Engineering

DBA Open ended experiment project Report on

***Food Delivery Management System.***

### List of Team Members:

### **ROLLNO NAME**

417 Michael Swaminathan

429 Nikhil dhupadal

410 Uttkarsh Khot

421 Prasannakumar Roogi

406 Naveen

|  |  |  |  |
| --- | --- | --- | --- |
|  | | **Content** |  |
| 1 |  | Introduction | 3 |
|  | 1.1 | Food Delivery system | 3 |
|  | 1.2 | Database Management System | 4 |
|  | 1.3 | Problem Statement | 5 |
|  | 1.4 | Objectives | 6 |
|  | 1.5 | Motivation | 6 |
| 2 |  | Requirement Collection and Analysis | 7 |
|  | 2.1 | Introduction | 7 |
|  | 2.2 | Data requirement | 8 |
|  | 2.3 | Functional Requirement | 9 |
|  | 2.4 | Non Functional Requirement | 10 |
| 3 |  | Database Design | 11 |
|  | 3.1 | Introduction | 11 |
|  | 3.2 | Conceptual Design | 11 |
| 4 |  | Implementation and Results | 16 |
|  | 4.1 | Introduction | 16 |
|  | 4.2 | Database Tables | 16 |
|  | 4.3 | Results | 19 |
|  |  | References | 20 |

# Chapter 1: Introduction

* 1. **Food Delivery System:**

**Online food ordering** is the process of ordering food from a website or other application. The product can be either ready-to-eat food (e.g., direct from a certified home-kitchen, restaurant, or a [ghost kitchen](https://en.wikipedia.org/wiki/Ghost_kitchen" \o "Ghost kitchen)) or food that has not been specially prepared for direction consumption (e.g., vegetables direct from a farm/garden, frozen meats. etc).

In a situation like the one which we are social gatherings are a matter of concern.

But people can’t simply stop eating their favorite cuisines as such pandemics may be frequent in near future.

The Online Food Delivery System is one of the fastest growing marketing strategy for most of the Business people, in away to gain the more profits. As we all know that the food is the basic need in every human life, for which he/she is struggling. But even after their struggles, if people are still not happy with their sustenance, then the effort given behind it is useless. The reasons might be the people are busy with their schedules, they either can’t able to cook in right time or not having time to go and order their food from outside. Do you want to freeze up your starving…? Then here is the way, where Spicy and Delicious Restaurants and other food stuff points have now started to deliver their food through the Apps as. Upon which, people can stop starving and could start to eat healthier food, just by clicking on the menus served on the apps. There are many more benefits offered by the Takeaway Food Delivery system, below are some

**Customer’s satisfaction:**

Since the customers need not to take risks of going outside, stand in queues, one could order from Nearby Indian, Chinese, Thai Takeaway Delivery, by sitting anywhere on the earth. This also tends the customers to fit their budget while ordering, by reducing the travelling expenses and other unnecessary taxes.

**No bothersome works:**

The previous way of food ordering through phone calls included many problems like, the staff of the Restaurant should speak to the different people talking with different accents, sometimes there would be the background disturbances. By all these activities, food ordering might go false. But now because of Order Online Food for Delivery, all the above difficulties are solved.

**Easy trade for Restaurants:**

If the people have started a new Restaurant or the Hotel, through their food point app, one can promote their Business leading to more customers. Also, customers can Find Dining Restaurants Nearby. Services are 24/7: Customers will be happy for the services catered by the Restaurant people at all time. Nowadays, Pizza, Sushi Delivery Online Orders are more, which are expected to have fast transference.

**Effortless Maintenance:**

With the ready posts of Chinese, Indian, Thai Food Restaurant Menu Online, there is no need to get printouts in case of updating the price, additional Items and other cases. Also, customers can compare the price and services offered by the different food points.

### Database managementSystem:

Formally, a "database" refers to a set of related data and the way it is organized. Access to this data is usually provided by a "database management system" (DBMS) [3] consisting of an integrated set of computer software that allows users to interact with one or more databases and provides access to all of the data contained in the database (although restrictions may exist that limit access to particular data). The DBMS provides various functions that allow entry, storage and retrieval of large quantities of information and provides ways to manage how that information is organized.

DBMSs provide various functions that allow management of a database and its data which can be classified into four main functional groups:

* **Data definition** – Creation, modification and removal of definitions that define the organization of thedata.
* **Update** – Insertion, modification, and deletion of the actual data
* **Retrieval** – Providing information in a form directly usable or for further processing by other applications. The retrieved data may be made available in a form basically the same as it is stored in the database or in a new form obtained by altering or combining existing data from the database.
* **Administration** – Registering and monitoring users, enforcing data security, monitoring performance, maintaining data integrity, dealing with concurrency control, and recovering information that has been corrupted by some event such as an unexpected system failure

### Relational database management system(RDBMS)

RDBMS [3] is a database management system (DBMS) based on the relational model of data. Most databases in widespread use today are based on this model. RDBMSs have been a common option for the storage of information in databases used for financial records, manufacturing and logistical information, personnel data, and other applications since the 1980s. Relational databases have often replaced legacy data models like hierarchical databases and network databases because they were easier to implement and administer. Nonetheless, relational databases received continued, unsuccessful challenges by object database management systems in the 1980s and 1990s, (which were introduced in an attempt to address the so-called object-relational impedance mismatch between relational databases and object-oriented application programs), as well as by XML database management systems in the 1990s.However, due to the expanse of technologies, such as horizontal scaling of computer clusters, NoSQL databases have recently become popular as an alternative to RDBMS databases.

### Problem Statement

### The project aims to design and Implement the database for Food delivery system to maintain the following.

### Each restaurant has the name and serves certain dishes of different cuisines, every restaurant has a special dish. Every restaurant is resided in different cities.The restaurant takes orders from the customer and the food is prepared and given to the delivery employee who will deliver the order. It keeps track of the orders that it handled.

### The delivery employee will take the food from the restaurant and deliver to the given address. Each delivery employee has a salary and rating for his skills from the customers to whom he had delivered before. He handles the orders and where the food is to be delivered.

### The customer will order from any restaurant which he likes provided he should order when the restaurant is open.

### Every restaurant has some deals to offer on specific days. The customer can avail the offers.

### The information of the types of dishes is to be maintained and which restaurant serves what dishes is also to be maintained. Even the information should include the dish price, its cuisine. Also certain dish is prepared at a certain time only.

### The food delivered to the given address, who delivered the order and what was the feedback should be maintained.

### 

### Objectives of the Project

Following are the main objectives of the study of this project work.

* To study and implement the basic database concepts.
* To understand the database design process.
* To Explore the activities of Food delivery procedures for designing the database.
* To study and familiar about the Structured Query Languages

### Motivation

The amount of data we produce every day is truly mind-boggling. There are about 2.5 quintillion bytes of data created each day at our current pace from the large organizations like Banking sector, Educational sector, reservation sector, health care sector and many other business applications. Storing, Maintaining and using data for Decision making are the challenging issues. These issues motivate us to design and develop database application for storing and managing the daily activities of The Food delivery system.

# Chapter 2: Requirement Collection and Analysis

## Introduction

The most critical aspect of specification is the gathering and compilation of system and user requirements. This process is normally done in conjunction with managers and users. The major goal in requirements gathering process is to:

* + - Collect the data used by the organization,
    - Identify relationships/conditions to be applied on the data,
    - Identify future data needs, and
    - Determine how the data is used and generated.
    - Identify the functions that are performed on the data

The starting place for data collection is gathering existing forms and reviewing policies and systems. Then, ask users what the data means, and determine their daily processes

Following subsections discuss the data requirements and functional & non functional requirements identified based on the following activities collected from the Food delivery system users.

* The customer has a name, Address, Mobile number which is unique to each customer. The address comprises of House no., Street name, city name, zipcode/pincode. Every customer is assigned a unique Customer id.
* The Menu contains Dish name, dish timings(i.e. from what time till what hour the dish is available), price of the dish, cuisine of the dish. Every dish has a unique id assigned to it.
* The restaurant has a name, address and mobile number which is unique to itself. The address comprises of the street name, city name and the pincode/zipcode, every restaurant has ratings maximum upto 5, also the restaurant will be open at specific timings. Every restaurant also has a special dish and unique restaurant id.
* There are deals for each day of given discount for a specific pay method. Every deal has a deal code.
* The Food is delivered by a delivery employee who has a name, mobile number unique to himself, Salary and shifts (has specific working hours). There are ratings given to each delivery employee also each delivery employee has a unique id.

## Data Requirement

Data requirement describes the data to be stored in the database pertaining to activities of Food delivery system requirement as described in section 2.1. Details of the data stored in the database is shown in the table 2.1 and Table2.2.

Table 2.1: Data to be stored in the database

|  |  |  |
| --- | --- | --- |
| Sl.  No. | Group | Data related to each group |
| 1 | Customer | ID, address, mobile number |
| 2 | Menu | Dish name, Dish cuisine, Dish id, price, timings |
| 3 | Restaurant | Id, name, address, ratings, timings, mobile number specialty |
| 4 | Deals | Deal code, discount in %, valid days, pay methods |
| 5 | Delivery employee | Name, id, salary, ratings, working hours. |

Table 2.2: Conditions on Data

|  |  |
| --- | --- |
| Sl. No. | Conditions |
| 1 | A customer can order only one type of dish in one order but quantity can be more than one. |
| 2 | A restaurant only serves few dishes which are mentioned in the menu |
| 3 | One delivery employee can deliver to many addresses. But one order should be handled by only one employee. |

## Functional Requirement

Functional requirements are product features or functions that developers must implement to enable users to accomplish their tasks. So, it’s important to make them clear both for the development team and the stakeholders (clients). Table 2.3 shows the different types users of Food delivery database application and their respective responsibilities (tasks). Table 2.4 shows the different functions and user can perform on the database

Table 2.3: Categories of Users and their tasks

|  |  |  |
| --- | --- | --- |
| Sl. No. | Users | Responsibilities (tasks) |
| 1. | Restaurant owners | Responsible for Viewing, modifying the data of restaurants |

Table 2.4: Functions of each user

|  |  |  |
| --- | --- | --- |
| Sl.  No. | Functions | User |
| 1 | Inserting data into the database | Restaurant owners, customers |
| 2 | Deleting the data from the database | Restaurant owners. Delivery System owner |
| 3 | Generating various results like order details. | Customer, restaurant owners, Delivery system owner |

## Non Functional Requirement

Nonfunctional Requirements (NFRs) define system attributes such as security, reliability, performance, maintainability, scalability, and usability. They serve as constraints or restrictions on the design of the system across the different backlogs. In this project we are addressing the security in order to secure the data and accessibility.

# Chapter 3: Database Design

## Introduction

The requirements gathering and specification provides you with a high-level understanding of the organization, its data, and the processes that you must model in the database. Database design involves constructing a suitable model of for the information. Since the design process is complicated, especially for large databases, database design is divided into three phases:

* + - Conceptual database design
    - Logical database design
* Physical database design.

In our project work we are addressing the conceptual database design using ER modeling and logical database design using the implementation data model called Relational model.

### Conceptual Database Design

Conceptual database design involves modelling the collected information at a high-level of abstraction without using a particular data model or DBMS. This model allows for easy communication between end-users and database developers and has a clear method to convert from high-level model to relational model. The most popular model for conceptual database design is the Entity Relationship model which describes data as attribute, entity and relationship.

Table 3.1 shows the list of attributes Table 3.2 shows the list of entity types and table

3.3 shows the list of relationship types identified for the requirement discussed in the section 2.1.

Table 3.1: List of attributes

|  |  |  |  |
| --- | --- | --- | --- |
| Sl.  No. | Attribute Name | Attribute Type | Justification |
| 1 | DISH\_NAME | Simple |  |
| 2 | DISH\_ID | Simple |  |
| 3 | DISH\_TYPE | Simple |  |
| 4 | DISH\_CUISINE | Simple |  |
| 5 | DISH\_PRICE | Simple |  |
| 6 | C\_ID | Simple |  |
| 7 | C\_address | Composite | Since address has 4 components |
| 8 | C\_Mob\_No | Simple |  |
| 9 | C\_Name | Composite | Since name has 3 components |
| 10 | D\_ID | Simple |  |
| 11 | D\_Mob\_No | Simple |  |
| 12 | D\_Salary | Simple |  |
| 13 | D\_Ratings | Simple |  |
| 14 | D\_Name | Composite | Since name has 3 components |
| 15 | D\_Working\_hours | Composite | Since working hours has 2 components |
| 16 | Discount | Simple |  |
| 17 | Valid\_day | Simple |  |
| 18 | Deal\_code | Simple |  |
| 19 | Valid\_pay\_method | Simple |  |
| 20 | R\_ID | Simple |  |
| 21 | R\_name | Simple |  |
| 22 | R\_address | Composite | Since address has 3 components |
| 23 | R\_raitngs | Simple |  |
| 24 | R\_timings | Composite | Since Timings has 2 components |
| 25 | R\_Mob\_No | Simple |  |
| 26 | R\_specialty | Simple |  |
| 27 | Order\_date | Simple |  |
| 28 | Feedback | Composite | Since it has two components |
| 29 | Orders\_ID | Simple |  |
| 30 | Orders\_Quantity | Simple |  |
| 31 | Order\_time | Simple |  |

Table 3.2: List of Entity Types

|  |  |  |  |
| --- | --- | --- | --- |
| Sl. No. | Entity Type Name | Type of Entity Type | Justification |
| 1 | Customer | Strong | It has key attribute |
| 2 | Restaurant | Strong | It has key attribute |
| 3 | Delivery Employee | Strong | It has key attribute |
| 4 | Menu | Strong | It has key attribute |

Table 3.3: List of Relationship Types

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Sl.  No. | Relationship Type  Name | Type of  Relationship Type | Justification | Participating Entity Type with cardinality ratio | Participation |
| 1 | Offers | Simple |  | Deals | Partial |
| Restaurant | Partial |
| 2 | Orders | Quaternary relationship | Four entity types are participating | Customer | Total |
| Menu | Partial |
| Delivery employee | Partial |
| Restaurant | Partial |
| 3 | Served by | Simple |  | Menu | Partial |
| Restaurant | Total |

* + 1. E-R Diagram, Schema Diagram and Normalization

Entity relationship diagram of the proposed system as described in the requirement analysis is shown in the figure 3.1and figure 3.2 shows the schema diagram obtained after converting ER diagram to relational model.

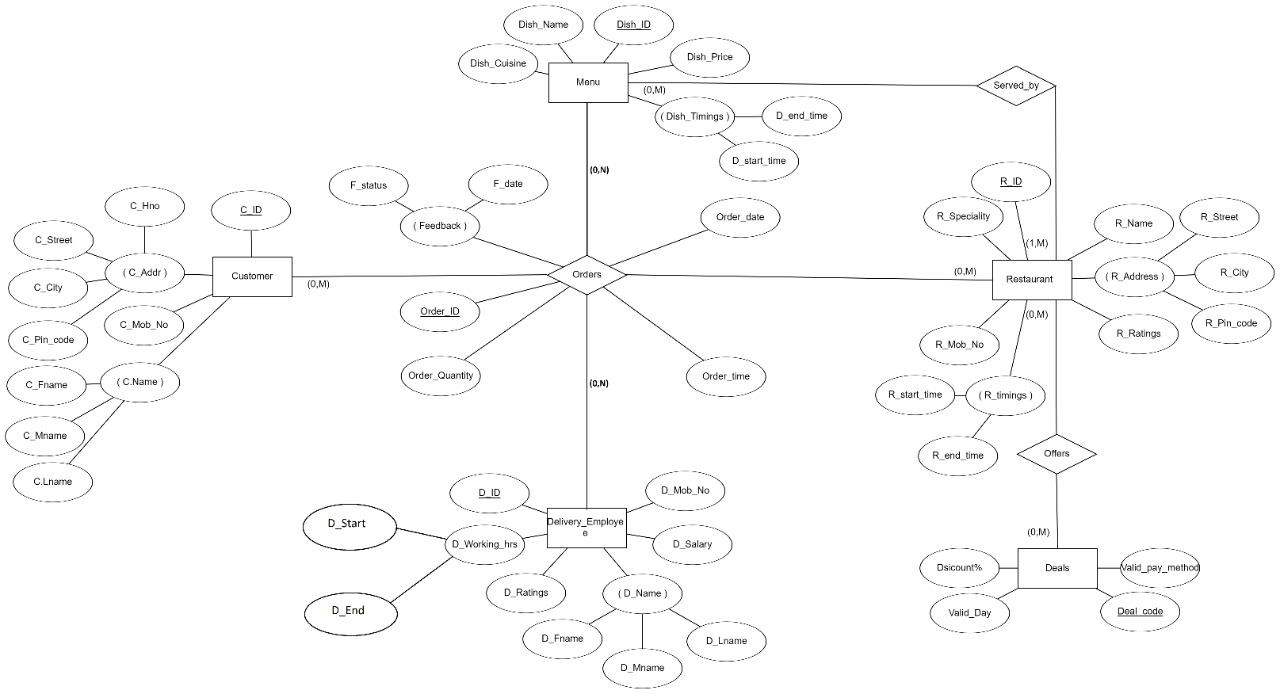
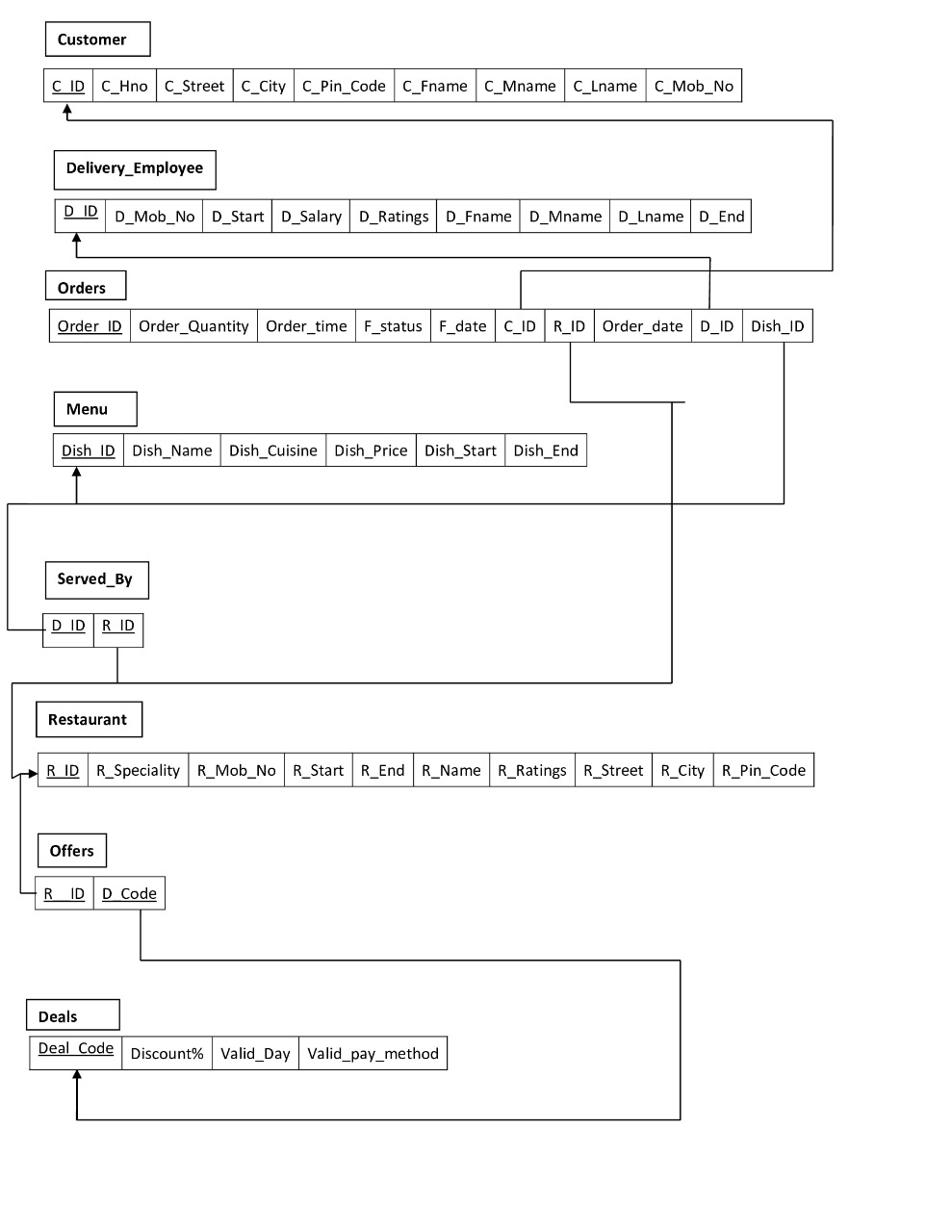


Fig.3.1: ER Diagram

Fig 3.2: Schema diagram of the Fig.3.1

CUSTOMER

(C\_ID,C\_Hno,C\_Street,C\_City,C\_Pin\_code,C\_Fname, C\_Mname, C\_Lname,C\_Mob\_No)

* + - * The Relation is in 1NF as it has atomic valued attributes
      * The Relation is in 2NF since, 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 in3NF.

DELIVERY\_EMPLOYEE

(D\_ID,D\_Mob\_No,D\_Start,D\_End,D\_Salary,D\_Ratings,D\_Fname,D\_Mname,D\_Lname)

* + - * The Relation is in 1NF as it has atomic valued attributes
      * The Relation is in 2NF since, 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 in3NF.

MENU

(Dish\_ID,Dish\_name,Dish\_cuisine,Dish\_Price,Dish\_Start,Dish\_End)

* + - * The Relation is in 1NF as it has atomic valued attributes
      * The Relation is in 2NF since, 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 in3NF.

RESTAURANT

(R\_ID,R\_specialty, R\_Mob\_No, R\_Start, R\_End, R\_Name, R\_Ratings, R\_Street, R\_City, R\_Pin\_code)

* + - * The Relation is in 1NF as it has atomic valued attributes
      * The Relation is in 2NF since, 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 in3NF.

DEALS

(Deal\_code,Discount,Valid\_day,Valid\_pay\_method)

* + - * The Relation is in 1NF as it has atomic valued attributes
      * The Relation is in 2NF since, 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 in3NF

ORDERS

(Order\_id,Order\_Quantity,Order\_time,F\_status,F\_date,C\_ID,R\_ID,Order\_date,D\_ID,Dish\_ID)

* + - * The Relation is in 1NF as it has atomic valued attributes
      * The Relation is in 2NF since, 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 in3NF

SERVED\_BY

(D\_ID, R\_ID)

* + - * The Relation is in 1NF as it has atomic valued attributes
      * The Relation is in 2NF since, 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 in3NF

Offers

(R\_\_ID,D\_CODE)

* + - * The Relation is in 1NF as it has atomic valued attributes
      * The Relation is in 2NF since, 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 in3NF

# Chapter 4. Implementation and Results

### Introduction

Implementation involves the construction of a database according to the specification of a logical schema. This will include the specification of an appropriate storage schema, security enforcement, external schema and so on. Implementation is influenced by the choice of available DBMSs, database tools and operating environment. There are additional tasks beyond simply creating a database schema and implementing the constraints such as data must be entered into the tables, issues relating to the users and user processes need to be addressed, and the management activities associated with wider aspects of corporate data management need to be supported.

In practice, implementation of the logical schema in a given DBMS requires a very detailed knowledge of the specific features and facilities that the DBM has to offer. In an ideal world, and in keeping with good software engineering practice, the first stage of implementation would involve matching the design requirements with the best available implementing tools and then using those tools for the implementation. In database terms, this might involve choosing vendor products with DBMS and SQL variants most suited to the database which is to be implemented. There are many relational DBMSs, available such as Oracle Database, Microsoft SQL Server, MySQL, IBM DB2, IBM Informix and Microsoft Access, use SQL. In this project we used Oracle SQL developer create the following tables of Food delivery system database.

* 1. **Database Tables**

Following tables table 4.1 to table 4.8 are the tables created for the schema diagram shown in figure 3.2.

Table 4.1: Customer table description

|  |  |  |
| --- | --- | --- |
| Attribute | Type | Constraints |
| C\_ID | NUMBER | Primary key |
| C\_HNO | NUMBER | NOT NULL |
| C\_STREET | VARCHAR2(20) | NOT NULL |
| C\_CITY | VARCHAR2(20) | NOT NULL |
| C\_PIN\_CODE | NUMBER | NOT NULL |
| C\_FNAME | VARCHAR2(20) | NOT NULL |
| C\_MNAME | VARCHAR2(20) | NOT NULL |
| C\_LNAME | VARCHAR2(20) | NOT NULL |
| C\_MOB\_NO | VARCHAR2(20) | UNIQUE |

Table 4.2: Delivery\_Employee table description

|  |  |  |
| --- | --- | --- |
| Attribute | Type | Constraints |
| D\_ID | NUMBER | Primary key |
| D\_MOB\_NO | VARCHAR2(20) | UNIQUE |
| D\_START | NUMBER | NOT NULL |
| D\_END | NUMBER | NOT NULL |
| D\_SALARY | FLOAT | NOT NULL |
| D\_RATINGS | NUMBER | NOT NULL |
| D\_FNAME | VARCHAR2(20) | NOT NULL |
| D\_MNAME | VARCHAR2(20) | NOT NULL |
| D\_LNAME | VARCHAR2(20) | NOT NULL |

Table 4.3: Menu table description

|  |  |  |
| --- | --- | --- |
| Attribute | Type | Constraints |
| DISH\_ID | NUMBER | Primary key |
| DISH\_NAME | VARCHAR2(30) | NOT NULL |
| DISH\_CUISINE | VARCHAR2(30) | NOT NULL |
| DISH\_PRICE | FLOAT | NOT NULL |
| DISH\_START | NUMBER | NOT NULL |
| DISH\_END | NUMBER | NOT NULL |

Table 4.4: Restaurant table description

|  |  |  |
| --- | --- | --- |
| Attribute | Type | Constraints |
| R\_ID | NUMBER | Primary key |
| R\_SPECIALTY | VARCHAR2(30) | NOT NULL |
| R\_MOB\_NO | VARCHAR2(30) | UNIQUE |
| R\_START | NUMBER | NOT NULL |
| R\_END | NUMBER | NOT NULL |
| R\_NAME | VARCHAR2(30) | NOT NULL |
| R\_RATINGS | NUMBER | NOT NULL |
| R\_STREET | VARCHAR2(30) | NOT NULL |
| R\_CITY | VARCHAR2(30) | NOT NULL |
| R\_PIN\_CODE | NUMBER | NOT NULL |

Table 4.5: Orders table description

|  |  |  |
| --- | --- | --- |
| Attribute | Type | Constraints |
| ORDER\_ID | NUMBER | Primary key |
| ORDER\_QUANTITY | NUMBER | NOT NULL |
| ORDER\_TIME | NUMBER | NOT NULL |
| ORDER\_DATE | DATE | NOT NULL |
| F\_STATUS | VARCHAR2(10) | F\_STATUS = ‘Good’ or ‘average’ or ‘bad’ |
| F\_DATE | DATE | NOT NULL |
| C\_ID | NUMBER | Foreign key references to Customer |
| R\_ID | NUMBER | Foreign key references to Restaurant |
| D\_ID | NUMBER | Foreign key references to Delivery\_Employee |
| DISH\_ID | NUMBER | Foreign key references to Menu |

Table 4.6: Deals table description

|  |  |  |
| --- | --- | --- |
| Attribute | Type | Constraints |
| DEAL\_CODE | NUMBER | Primary Key |
| DISCOUNT | FLOAT | NOT NULL |
| VALID\_DAY | VARCHAR2(10) | NOT NULL |
| VALID\_PAY\_METHOD | VARCHAR2(10) | ‘COD’ or ‘ONLINE’ only |

Table 4.7: Offers table description

|  |  |  |
| --- | --- | --- |
| Attribute | Type | Constraints |
| R\_\_ID | NUMBER | Foreign key references to Restaurant |
| D\_CODE | NUMBER | Foreign key references to Deals |

Table 4.8: Served\_by table description

|  |  |  |
| --- | --- | --- |
| Attribute | Type | Constraints |
| R\_ID | NUMBER | Foreign key references to Restaurant |
| D\_ID | NUMBER | Foreign key references to menu |

Following syntax shows for creating database table shown with an example for the Table4.1

### Table Creation:

Create Table Customer

(

C\_ID int NOT NULL,

C\_Hno int NOT NULL,

C\_Street varchar(20) NOT NULL,

C\_City varchar(20) NOT NULL,

C\_Pin\_code int NOT NULL,

C\_Fname varchar(20) NOT NULL,

C\_Mname varchar(20) NOT NULL,

C\_Lname varchar(20) NOT NULL,

C\_Mob\_No varchar(20) NOT NULL,

primary key(C\_ID),

unique(C\_Mob\_No)

);

### 4.3. Results

Figures Fig.3.3 and Fig.3.4 shows the sample reports of the proposed system. Fig.3.3 Customer Details who are from Piedra Aguza only if they have ordered something and Fig.3.4 Restaurant name along with it's special dish and it's price.

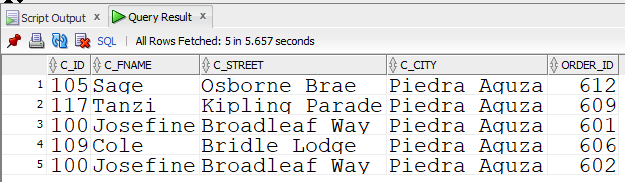


Fig.3.3: Customer details

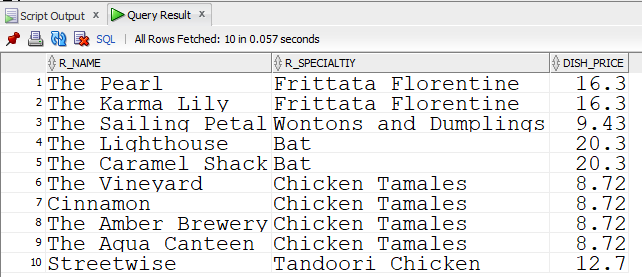


Fig.3.4: Restaurant Details

# References

1. <http://derf.org.uk/history.html>
2. <http://www.oldclassiccar.co.uk/britishschoolmotoring.html>
3. https://[www.oracle.com/database/technologies/appdev/sql-developer.html](http://www.oracle.com/database/technologies/appdev/sql-developer.html)
4. Ramez Elmasri, Shamkant B. Navathe “Fundamentals of Database Systems, 7th Edition “pearson,2016
5. Raghu Ramakrishnan, Johannes Gehrke, “Database Management Systems” 3rd Edition, McGraw Hill Higher Education,2003