CHAPTER 3

DATABASE DESIGN

3.1-Requirements And Constraints

3.1.1 Functional Requirements:

1-Distributed Database:

Distributed Database implies that a single application should be able to operate transparently on data that is spread across a variety of different databases and connected by a Communication Network.

2-Client/Server System

The term client/server refers primarily to an architecture or logical division of responsibilities, the client is the application (also known as the front-end), and the server is the DBMS (also known as the back-end).

A client/server system is a distributed system in which,

- Some sites are client sites and others are server sites.
- All the data resides at the server sites.
- All applications execute at the client sites.

3-User Interfaces

Front-End Software: HTML, CSS, JAVASCRIPT, BOOTSTRAP.

Back-End Software: MySQL

4-Hardware Interfaces

- Windows.
- A browser which supports php and html

5-Security Requirements

- Security systems need database storage just like many other applications. However, the special requirements of the security market mean that vendors must choose their database partner carefully.

3.1.2 Constraints

Constraints on The Relational Database Are Of 4 Types:

1. Domain Constraints
2. Key Constraints
3. Entity Integrity Constraints
4. Referential Integrity Constraints

1. Domain Constraints:

1. Every domain must contain atomic values (smallest indivisible units) it means composite and multi-valued attributes are not allowed.
2. We perform datatype check here, which means when we assign a data type to a column, we limit the values that it can contain.
   E.g., If we assign the datatype of attribute age as int, we can't give it values other than int datatype.

2. Key Constraints Or Uniqueness Constraints:

1. These are called uniqueness constraints since it ensures that every tuple in the relation should be unique.
2. A relation can have multiple keys or candidate keys(minimal super key), out of which we choose one of the keys as primary key, we don't have any restriction on choosing the primary key out of candidate keys, but it is suggested to go with the candidate key with less number of attributes.
3. Null values are not allowed in the primary key, hence not null constraint is also a part of key constraint.

3. Entity Integrity Constraints:

1. Entity integrity constraints says that no primary key can take null value, since using primary key we identify each tuple uniquely in a relation.

4. Referential Integrity Constraints:

1. The referential integrity constraints is specified between two relations or —— tables and used to maintain the consistency among the tuples in two relations.
2. This constraint is enforced through foreign key, when an attribute in the foreign key of relation R1 have the same domain(s) as the primary key of relation R2, then the foreign key of r1 is said to reference or refer to the primary key of relation R2.
3. The values of the foreign key in a tuple of relation R1 can either take the values of the primary key for some tuple in relation R2, or can take null values, but can't be empty.

3.2 Entities and Attribute

1) ADLOGIN
    i) Email
    ii) Password
    iii) A_Id
2) ADMIN
    i) A_Id
    ii) Email
    iii) Name
    iv) Pnumber

3) CONTACTS
    i) Id
    ii) Email
    iii) Subject
    iv) Msg

4) COURIER
    i) C_Id
    ii) U_Id
    iii) Semail

iv) Remail
v) Sname
vi) Rname
vii) Sphone
viii) Rphone
xi) Saddress
x) Raddress
xi) Weight
xii) Billno
xiii) Image
xiv) Date

5) LOGIN
   i) Email
   ii) Password
   iii) U_Id

6) LOGSS
   i) User_Id
   ii) Action_Time
   iii) Action_Performed
   iv) Action_Performed_By

7) USERS
   i) U_Id
   ii) Email
   iii) Name
   iv) Pnumber

3.3 ENTITY RELATIONSHIP DIAGRAM

An **Entity–relationship model (ER model)** describes the structure of a database with the help of a diagram, which is known as **Entity Relationship Diagram (ER Diagram)**. An ER model

is a design or blueprint of a database that can later be implemented as a database. The main components of E-R model are: entity set and relationship set.

What is an Entity Relationship Diagram (ER Diagram)?

An ER diagram shows the relationship among entity sets. An entity set is a group of similar entities and these entities can have attributes. In terms of DBMS, an entity is a table or attribute of a table in database, so by showing relationship among tables and their attributes, ER diagram shows the complete logical structure of a database.

The geometric shapes and their meaning in an E-R Diagram. We will discuss these terms in detail in the next section (Components of a ER Diagram) of this guide so don't worry too much about these terms now, just go through them once.

Rectangle: Represents Entity sets.[**adlogin, login, admin, user, courier, contacts**]
Ellipses:Attributes[**(password,email),(email,password),(email,a_id,phone,name), (c_id,sname,rname,semail,remail,sphone,rphone,saddress,raddress,billno,date), (email,comments,date,title)]**
Diamonds: Relationship Set[**has, takes care, manage, have, can.**]
Lines: They link attributes to Entity Sets and Entity sets to Relationship Set
Double Ellipses: Multivalued Attributes
Dashed Ellipses: Derived Attributes
Double Rectangles: Weak Entity Sets
Double Lines: Total participation of an entity in a relations

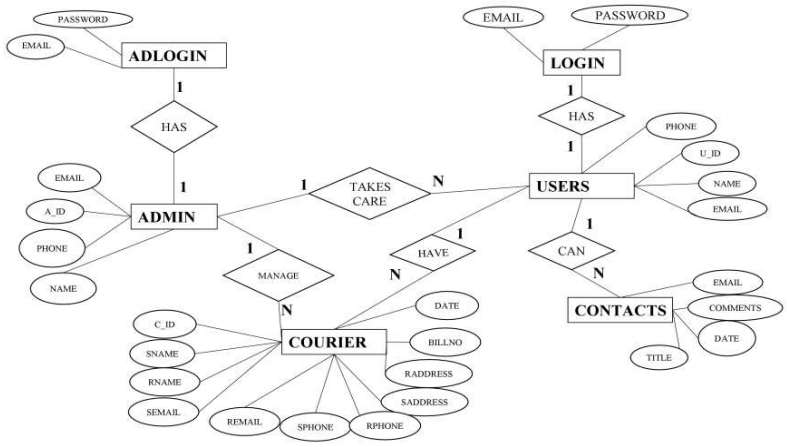

Fig 3.1 ER Diagram For Courier Management System

3.4 SCHEMA DIAGRAM

A database schema is the skeleton structure that represents the logical view of the entire database. It defines how the data is organized and how the relations among them are associated. It formulates all the constraints that are to be applied on the data.
A database schema defines its entities and the relationship among them. It contains a descriptive detail of the database, which can be depicted by means of schema diagrams.

An Entity-Relationship Model (ERM) is an abstract and conceptual representation of data. Entity-relationship modelling is a database modelling method, used to produce a type of conceptual schema or semantic data model of a system, often a relational database, and its requirements in a top-down fashion.

In order to create an ER schema you must know three main concepts: entity, attribute and relationship.

Entity

The entity is the central concept of the Entity-Relationship model. An entity represents a description of the common features a of the set of objects in the real world. Examples of entities are Person, Car, Artist, and Album.

Attribute

An Attribute represents the properties of real world objects that are relevant for the application purposes. Attributes are associated with the concept of Entity, with the meaning that all the instances of the entity are characterized by the same set of attributes. In other words, the entity is a descriptor of the common properties of a set of objects, and such properties are expressed as attributes.

Relationship

A Relationship represents semantic connections between entities, like the association between an artist and his/her album, or between an artist and his/her reviews.

The possible values are one and many. Based on their maximum cardinality constraints, relationships are called

1."one-to-one", if both relationships roles have maximum cardinality 1,

2."one-to-many", if one relationship role has maximum cardinality 1 and the other role has maximum cardinality N,

3."many-to-many", if both relationships roles have maximum cardinality N.

DATAFLOW

The data flow shows the flow of information from a source to its destination. Data flow is represented by a line, with arrowheads showing the direction of flow. Information always flows to or from a process and may be written, verbal or electronic. Each data flow may be referenced by the processes or data stores at its head and tail, or by a description of its contents

DATA STORE

A data store is a holding place for information within the system: It is represented by an open ended narrow rectangle. Data stores may be long-term files such as sales ledgers or may be short-term accumulations: for example batches of documents that are waiting to be processed. Each data store should be given a reference followed by an arbitrary number

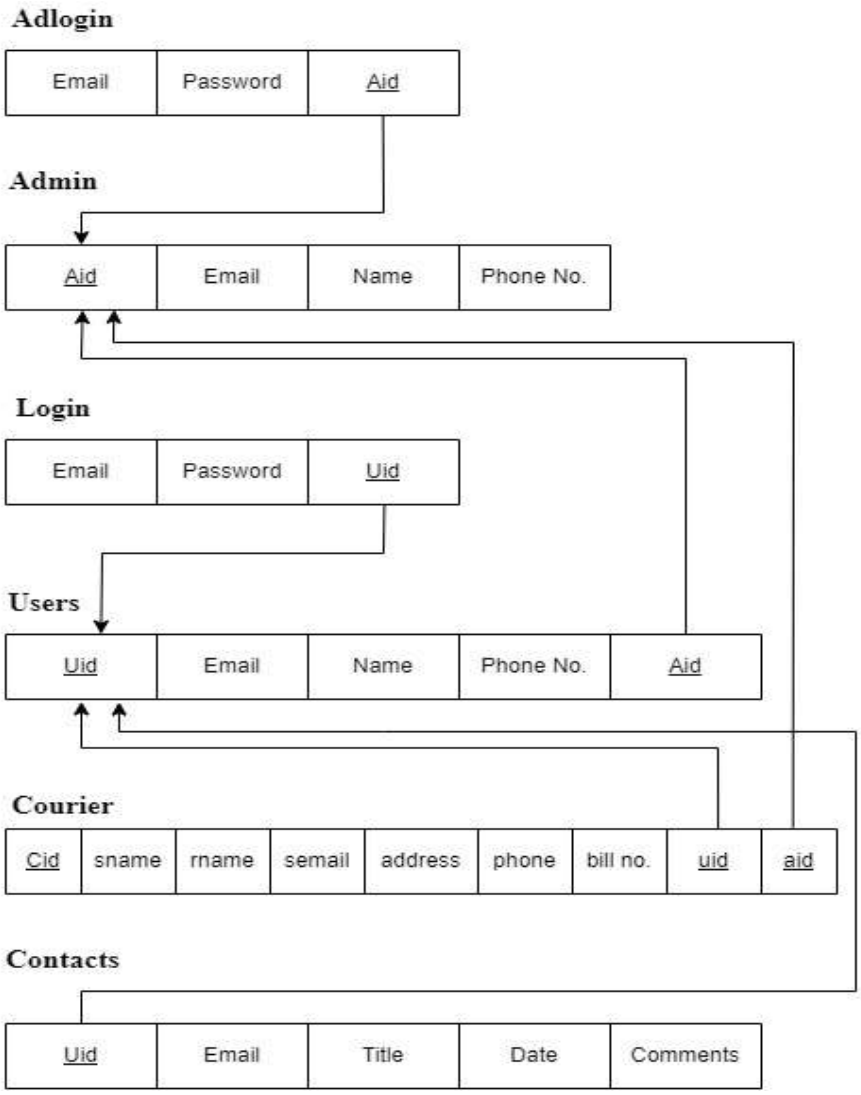

Fig 3.2 Schema Diagram For Courier Management System

FIG 3.2 Is the schema diagram, schema diagram gives us the idea of what are the keys in the database, in fig 3.2 we can see that adlogin has a primary key as a_id and a_id has been referenced by a few entities naming, courier, user, admin