**Chapter 5**

**Database Design**

**5.1 Introduction**

Database Design maintains the data required by the system. One of the key design issues involved in the database to design is the distribution of data in a way that minimizes transaction traffic. Another key design issue is the choice of the database management system. Database Tables used are described in the following sections.

**Database**

A database is a collection of data organized to allow easy access of retrievals, additions, modifications and deletions.

A typical database consists of different parts as shown in the following figure. The Database server (Sqlite3 Server) in the top level is software that can be accessed by multiple users. Within the database server, number of databases can be stored.

Each database stored data in the series of tables that can be related to each other in different ways. The most widely used approach for structuring the data is called Relational database system (RDBMS). A RDBMS is a collection of tables of data.

Database design is a process of organizing the data in an orderly manner so as to provided easy to the required information.

It is difficult to Maintain the database if lot of repetitive data stored in the table. If one instance of the data undergoes a change, that undergoes a change has to be made for all occurrences of the data. To eliminate duplication and easy maintenance of data, it is recommended to create a table of repeated values in one table called master table. The reference remains same throughout the other tables.

**5.2 Schema Description**

**Database: CoolLens**

## Table name: Auth\_User

|  |  |  |  |
| --- | --- | --- | --- |
| **Field Name** | **Type** | **Constraints** | **Description** |
| Id | Int (11) | Primary Key | User ID |
| Password | Varchar (128) | Not null | Password |
| First Name | Varchar (150) | Not null | UserFirst Name |
| Last Name | Varchar (150) | Not null | UserLast Name |
| Email | Varchar (254) | Not null | User Email |
| Is staff | bool | Not null | Staff |
| Is Active | bool | Not null | Activation |
| Date joined | Date Time | Not null | Date joined |
| Last login | Date Time | Not null | Last login |
| Is superuser | bool | Not null | Is superuser |

**Table name: Django\_admin\_log**

|  |  |  |  |
| --- | --- | --- | --- |
| **Field Name** | **Type** | **Constraints** | **Description** |
| *User Id* | int (11) | Primary Key | User ID |
| Dob | Date | null | Dob |
| Photo | varchar (100) | Not null | Photo |
| Mobile | varchar (10) | Not null | Mobile Number |
| Pincode | Varchar (6) | Not null | Pincode |
| Statere325 | Varchar (50) | Not null | State |
| Address | Text | Not null | Address |

## Table name: customer

|  |  |  |  |
| --- | --- | --- | --- |
| **Field Name** | **Type** | **Constraints** | **Description** |
| Customerid | int (11) | Primary Key | Customer ID |
| Customername | varchar (100) | Not null | Customer name |
| Emailid | varchar (200) | Not null | Customer email ID |
| Password | varchar (100) | Not null | Customer password |
| Address | Varchar (250) | Not null | Customer Address |
| Pincode | varchar (10) | Not null | Customer PIN code |
| City | varchar (25) | Not null | Customer city |
| State | varchar (25) | Not null | Customer state |
| Contactno | varchar (20) | Not null | Customer contact no |
| Status | varchar (10) | Not null | Active / Inactive |

**Table name: product**

|  |  |  |  |
| --- | --- | --- | --- |
| **Field Name** | **Type** | **Constraints** | **Description** |
| *Productid* | int(11) | Primary Key | Product ID |
| Categoryid | int(11) | Foreign Key | Category ID |
| shop\_id | int(11) | Foreign Key | Shop ID |
| Product\_name | varchar (35) | Not null | Product name |
| Description | Text | Not null | Product description |
| GST | int(11) | Not null | GST |
| Price | float(10,2) | Not null | Product Price |
| Subcategoryid | int(11) | Foreign Key | Subcategory ID |
| Image | varchar (35) | Not null | Product Image |

## Table name: product\_order

|  |  |  |  |
| --- | --- | --- | --- |
| **Field Name** | **Type** | **Constraints** | **Description** |
| *\_id* | int (11) | Primary Key | ID |
| Usertid | int (11) | Foreign Key | Product ID |
| Products | varchar (200) | Not null | Products |
| Saler | Text | Not null | Saler |
| Status | varchar (11) | Not null | Active / Inactive |

## Table name: saler\_order

|  |  |  |  |
| --- | --- | --- | --- |
| **Field Name** | **Type** | **Constraints** | **Description** |
| *Purchaseid* | int (11) | Primary Key | Purchase ID |
| Customerid | int (11) | Foreign Key | Customer ID |
| Order id | int (11) | Foreign Key | Order ID |
| Size | Varchar (10) | Not null | Size |
| Cost | float (10,2) | Not null | Product cost |
| Status | varchar (25) | Not null | Pending / Active |
| Companyid | int (11) | Foreign Key | Seller ID |
| cancellationreason | Varchar (200) | Not null | Cancellation reason |
| Sellerpayment | varchar (20) | Not null | Paid / Not paid |
| Purchasedate | Date | Not null | Purchase date |
| Deliverystatus | Text | Not null | Delivery status |

**Table name: seller\_detail**

|  |  |  |  |
| --- | --- | --- | --- |
| **Field Name** | **Type** | **Constraints** | **Description** |
| *user\_id* | int (11) | Primary Key | Company ID |
| Shopname | varchar (500) | Not null | Seller’s shop name |
| Address | text | Not null | Seller address |
| State | varchar (50) | Not null | Seller state |
| City | varchar (100) | Not null | Seller city |
| Pincode | varchar (6) | Not null | Seller pincode |
| Landmark | varchar (500) | Not null | Seller landmark |
| Account\_holder | varchar (50) | Not null | Seller Account Holder |
| Account\_Number | varchar (20) | Not null | Seller account number |
| Ifsc\_Code | varchar (11) | Not null | Seller Ifsc code |

**5.3 Entity Relationship diagram:**

ER diagram is a visual representation of data that describes how data is related to each other.

**Symbols and Notations:**

**Entity**

**Relationship**

**Attribute**

**Weak Entity**

**Weak Entity Relationship**

**Multivalued Attribute**

**Key attribute**

**Composite Attribute**

**Entities:**

Entities are objects or concepts that represent important data. They are typically nouns. E.g., costumer, supervisor, location, or promotion.

**Strong Entities:** Strong Entities exist independently from other entity types. They always possess one or more attributes that uniquely distinguish each occurrence of the way.

**Weak Entities:** Weak Entities depend on some other entity type. they don’t possess unique attribute (also known as a primary key) and have no meaning in the diagram without depending on another entity. This other entity is known as the owner.

**Associative Entities:** Associative Entities are entities that associate the instances of one or more entity types. They also contain attributes that are unique to the relationship between those entity instances.

**Relationship:**

**Strong Relationship:** Strong Relationship are meaningful associations between or among entities. They are usually verbs, e.g., assign, associate, or track. A relationship provides useful information that could not be discerned with just the entity types.

**Weak Relationships:** Weak Relationship or Identifying Relationship, are connection that exist between a weak entity type and its owner.

**Keys:**

**Primary Key:** Primary key is a uniquely identify each column in the table.

**Foreign Key:** Foreign Key is used to point to a Primary Key of another table.

**Attributes:**

**Attributes** are characteristics of either an entity. A many-to- many relationship, or A one-to-one relationship.

**Multivalued Attributes:** Multivalued Attributes whose value can be calculated from related attribute values.

**Key Attribute:**  Key Attribute represents the main characteristic of an entity it is used to represent primary key ellipse with underlying lines represent key attribute.

**Composite Attribute:** An attribute can also have their own attribute; these attributes are known as composite attribute.