**CHAPTER 4**

**IMPLEMENTATION**

**DONOR**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Field | Type | Null | Key | Default | Extra |
| DID | Int(4) | NO | PRI | NULL | Auto increment |
| Dname | varchar(15)Char(9) | YES |  | NULL |  |
| Sex | int(11) | YES |  | NULL |  |
| Age | varchar(60) | YES |  | NULL |  |
| Address | Varchar(11) | YES |  | NULL |  |
| Phno | Varchar(11) | YES |  | NULL |  |

Table no 4.1.1 DONOR

**BLOOD**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Field | Type | Null | Key | Default | Extra |
| Btype | Char(3) | NO | PRI | NULL |  |

Table no 4.1.2 BLOOD

**BBANK**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Field | Type | Null | Key | Default | Extra |
| BID | Int(4) | YES |  | NULL |  |
| Btype | Char(3) | YES | MUL | NULL |  |
| Quantity | float | YES |  | NULL |  |
| MID | Int(4) | YES | MUL | NULL |  |

Table no 4.1.3 BBANK

**MANAGER**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Field | Type | Null | Key | Default | Extra |
| Mname | Varchar(15) | YES |  | NULL |  |
| MID | Int(4) | NO | PRI | NULL |  |
| Mphno | Varchar(11) | YES |  | NULL |  |

Table no 4.1.4 MANAGER

**RECEPTIONIST**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Field | Type | Null | Key | Default | Extra |
| Rname | Varchar(15) | YES |  | NULL |  |
| RID | Int(4) | YES |  | NULL |  |
| Rphno | Varchar(11) | YES |  | NULL |  |
| MID | Int(4) | YES | MUL | NULL |  |

Table no 4.1.5 RECEPTIONIST

**ORDERS**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Field | Type | Null | Key | Default | Extra |
| Ord no | int(4) | YES |  | NULL |  |
| Ord date | date | YES |  | NULL |  |
| Btype | char(3) | YES |  | NULL |  |
| Quantity | float | YES |  | NULL |  |
| HID | Char(4) | YES | MUL | NULL |  |

Table no 4.1.6 ORDERS

**HOSPITAL**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Field | Type | Null | Key | Default | Extra |
| HID | char(4) | NO | PRI | NULL |  |
| Hname | varachar (20) | YES |  | NULL |  |
| haddress | varchar(60) | YES |  | NULL |  |
| Hphno | varchar(11) | YES |  | NULL |  |

Table no 4.1.7 HOSPITAL

**4.2 ODBC Connection**

An ODBC driver uses the open database connectivity interface by Microsoft that allows applications to access the data in database management systems using SQL as a standard for accessing the data. ODBC permits maximum interoperability, which means an application can access different DBMS.

**Connection Statement**

String MyConnection=

"server=localHost;userid=root;password=rootmysql;database=BLOODBANK";

MySqlConnection MyConn = new MySqlConnection(MyConnection);

MySqlCommand MyCommand = new MySqlCommand(Query, MyConn);

MySqlDataReader MyReader;

MyReader.ExecuteReader();

MyConn.Open();

MyConn.Close();

**4.3 Triggers and Stored Procedures**

**TRIGGERS**

A SQL trigger is a set of SQL statements stored in the database catalog. A SQL trigger is executed or fired whenever an event associated with a table occurs e.g., insert, update or delete.

A SQL trigger is a special type of stored procedure. It is special because it is not called directly like a stored procedure. The main difference between a trigger and a stored procedure is that a trigger is called automatically when a data modification event is made against a table whereas a stored procedure must be called explicitly.

Eg:

|  |
| --- |
| #TRIGGER  DELIMITER $$  CREATE TRIGGER Olderrcrd  AFTER INSERT ON BBANK  FOR EACH ROW  DELETE FROM ORDERS  WHERE Ord\_date<date\_sub(now(),interval 30 day);  DELIMITER ; |

**Figure 4.3.1** Trigger table

Trigger is used to delete the Orders from the ORDERS table.Trigger is fired when values are inserted into BBANK table after which Orders from the ORDERS table are deleted which are more than 30 days old.

**STORED PROCEDURE**

A stored procedure is a segment of declarative SQL statements stored inside the database catalog. A stored procedure can be invoked by triggers, other stored procedures, and applications such as Java, Python, PHP.

A stored procedure that calls itself is known as a recursive stored procedure. Most database management systems support recursive stored procedures.

Eg:

|  |
| --- |
| #STORED PROCEDURES  DELIMITER //  CREATE PROCEDURE Bloodsearch  (in Bty char(3),in Qty float(10))  SELECT \* FROM BBANK WHERE  Btype = Bty AND Quantity >= Qty;  DELIMITER ; |

**Figure 4.3.2** Stored Procedure table

The stored procedure is used to display a table containing entered Bloodtype and Quantity