**INDEX**

1. Abstract…………………….………………………………………………………………..2

2. Introduction……………….………………………………………………………................3

3. History………………….…………………………………………………………………....6

4. E-R Diagram……..……………………………………………………………………….....9

5. Queries with Output……………..…………………………………………………………..10

6. Final Database….……………………………………………………………………………16

7. Applications………………………………………………………………………………….18

8. Conclusion………………………………………………………………………………..….19

**Abstract**

This paper is scrutinizes the use of different terms and syntaxes in Database Management System, enabling viewer to get the complete concept of different aspects of Database Management System. To satisfy this we created a database managed for Blood Bank System. Use of various syntaxes was used as a reference to the output, satisfying every need of a perfect Database Management.

**Introduction**

1. **Database Management System:**

A database is an organized collection of data, generally stored and accessed electronically from a computer system. Where databases are more complex they are often developed using formal design and modeling techniques. The database management system (DBMS) is the software that interacts with end users, applications, and the database itself to capture and analyze the data. The DBMS software additionally encompasses the core facilities provided to administer the database. The sum total of the database, the DBMS and the associated applications can be referred to as a "database system". Often the term "database" is also used to loosely refer to any of the DBMS, the database system or an application associated with the database.

Computer scientists may classify database-management systems according to the database models that they support. Relational databases became dominant in the 1980s. These model data as rows and columns in a series of tables, and the vast majority use SQL for writing and querying data. In the 2000s, non-relational databases became popular, referred to as NoSQL because they use different query languages.

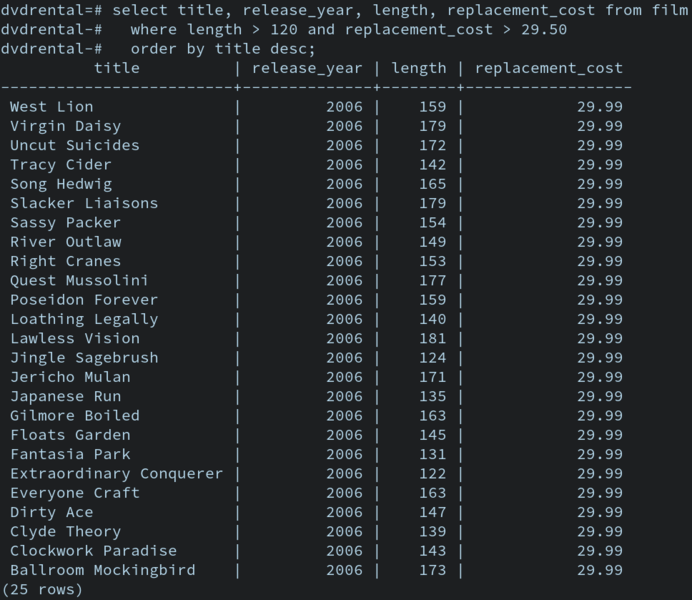


Fig.1. A Database

1. **SQL:**

SQL (Structured Query Language) is a domain-specific language used in programming and designed for managing data held in a relational database management system (RDBMS), or for stream processing in a relational data stream management system (RDSMS). It is particularly useful in handling structured data, i.e. data incorporating relations among entities and variables. SQL offers two main advantages over older read–write APIs such as ISAM or VSAM. Firstly, it introduced the concept of accessing many records with one single command. Secondly, it eliminates the need to specify how to reach a record, e.g. with or without an index.

Originally based upon relational algebra and tuple relational calculus, SQL consists of many types of statements,which may be informally classed as sublanguages, commonly: a data query language (DQL), a data definition language (DDL), a data control language (DCL), and a data manipulation language (DML). The scope of SQL includes data query, data manipulation (insert, update and delete), data definition (schema creation and modification), and data access control. Although SQL is essentially a declarative language (4GL), it also includes procedural elements.



Fig.2. Structured Query Language (SQL)

1. **RDBMS:**

A relational database (RDBMS) is a digital database based on the relational model of data, as proposed by **E. F. Codd** in 1970. A software system used to maintain relational databases is a relational database management system (RDBMS). Many relational database systems have an option of using the SQL (Structured Query Language) for querying and maintaining the database. The most common definition of an RDBMS is a product that presents a view of data as a collection of rows and columns, even if it is not based strictly upon relational theory. By this definition, RDBMS products typically implement some but not all of Codd's 12 rules.

A second school of thought argues that if a database does not implement all of Codd's rules it is not relational. This view, shared by many theorists and other strict adherents to Codd's principles, would disqualify most DBMSs as not relational. For clarification, they often refer to some RDBMSs as truly-relational database management systems (TRDBMS), naming others pseudo-relational database management systems (PRDBMS).As of 2009, most commercial relational DBMSs employ SQL as their query language.Alternative query languages have been proposed and implemented, notably the pre-1996 implementation of Ingres QUEL.

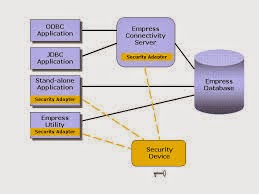


Fig.3. A RDBMS Structure

**History**

1. **DBMS:**

The sizes, capabilities, and performance of databases and their respective DBMSs have grown in orders of magnitude. These performance increases were enabled by the technology progress in the areas of processors, computer memory, computer storage, and computer networks. The concept of a database was made possible by the emergence of direct access storage media such as magnetic disks, which became widely available in the mid 1960s; earlier systems relied on sequential storage of data on magnetic tape. The relational model, first proposed in 1970 by Edgar F. Codd, departed from this tradition by insisting that applications should search for data by content, rather than by following links. By the early 1990s, however, relational systems dominated in all large-scale data processing applications, and as of 2018 they remain dominant: IBM DB2, Oracle, MySQL, and Microsoft SQL Server are the most searched DBMS.The dominant database language, standardised SQL for the relational model, has influenced database languages for other data models.

The next generation of post-relational databases in the late 2000s became known as NoSQL databases, introducing fast key-value stores and document-oriented databases. A competing "next generation" known as NewSQL databases attempted new implementations that retained the relational/SQL model while aiming to match the high performance of NoSQL compared to commercially available relational DBMSs.

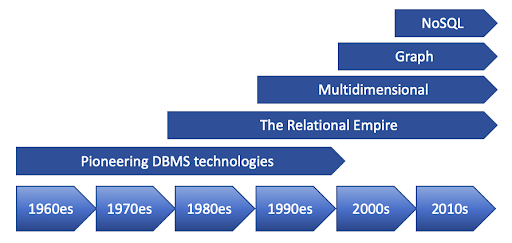


Fig.4. History

1. **SQL:**

SQL was initially developed at IBM by Donald D. Chamberlin and Raymond F. Boyce after learning about the relational model from Edgar F. Codd in the early 1970s. This version, initially called SEQUEL (Structured English Query Language), was designed to manipulate and retrieve data stored in IBM's original quasi-relational database management system, System R, which a group at IBM San Jose Research Laboratory had developed during the 1970s. Chamberlin and Boyce's first attempt at a relational database language was Square, but it was difficult to use due to subscript notation. After moving to the San Jose Research Laboratory in 1973, they began work on SEQUEL.

In the late 1970s, Relational Software, Inc. (now Oracle Corporation) saw the potential of the concepts described by Codd, Chamberlin, and Boyce, and developed their own SQL-based RDBMS with aspirations of selling it to the U.S. Navy, Central Intelligence Agency, and other U.S. government agencies. In June 1979, Relational Software, Inc. introduced the first commercially available implementation of SQL, Oracle V2 (Version2) for VAX computers.

By 1986, ANSI and ISO standard groups officially adopted the standard "Database Language SQL" language definition. New versions of the standard were published in 1989, 1992, 1996, 1999, 2003, 2006, 2008, 2011 and, most recently, 2016.



Fig.5. Oracle Corporations

1. **RDBMS:**

The term "relational database" was invented by E. F. Codd at IBM in 1970. Codd introduced the term in his research paper "A Relational Model of Data for Large Shared Data Banks". In this paper and later papers, he defined what he meant by "relational". One well-known definition of what constitutes a relational database system is composed of Codd's 12 rules. However, no commercial implementations of the relational model conform to all of Codd's rules, so the term has gradually come to describe a broader class of database systems, which at a minimum.

The most common definition of an RDBMS is a product that presents a view of data as a collection of rows and columns, even if it is not based strictly upon relational theory. By this definition, RDBMS products typically implement some but not all of Codd's 12 rules.

As of 2009, most commercial relational DBMSs employ SQL as their query language.

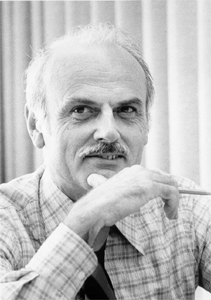


Fig.6. E.F. Codd

**E-R Diagram**

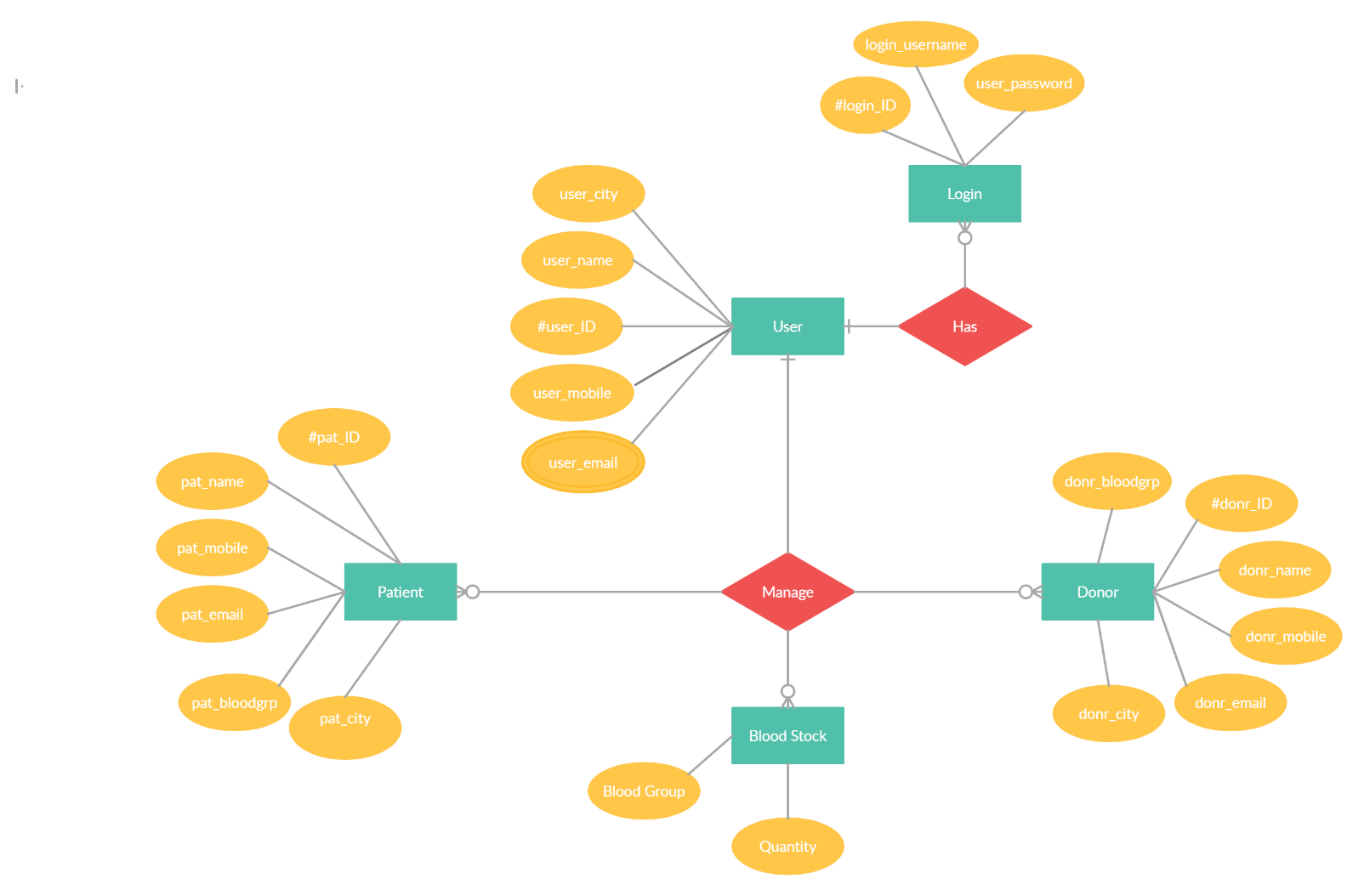
****

Fig. 7. E-R Diagram

**Queries With Output**

CREATE TABLE Login (

login\_ID int,

login\_username varchar(255),

user\_password varchar(255)

);

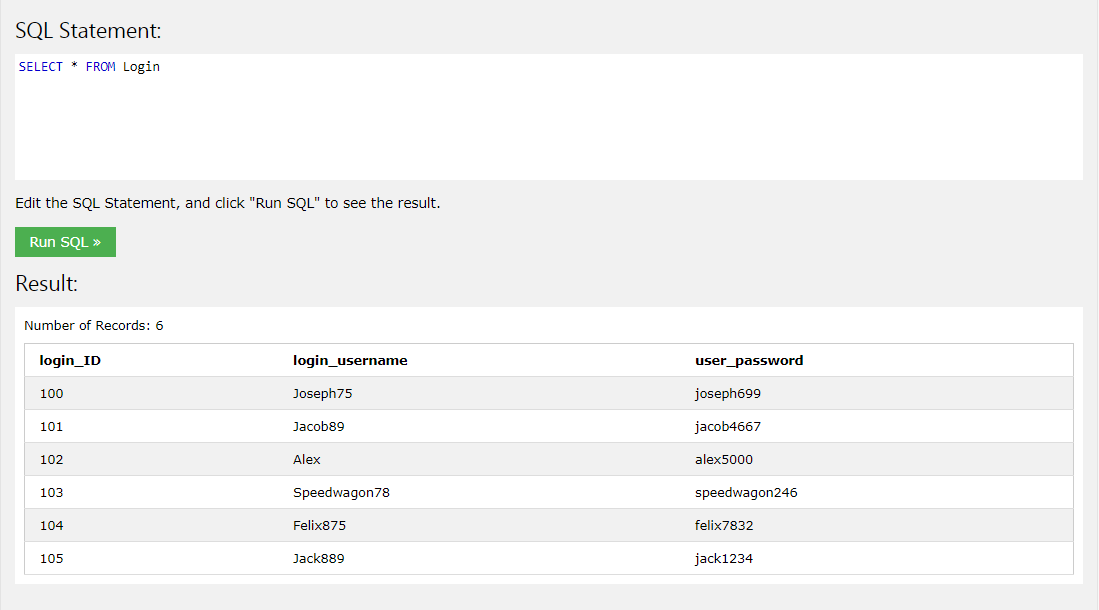
INSERT INTO Login (login\_ID,login\_username,user\_password)

VALUES ('100','Joseph75','joseph699');



/\* After 5 statements \*/

Select \* from Login



CREATE TABLE User (

user\_ID int,

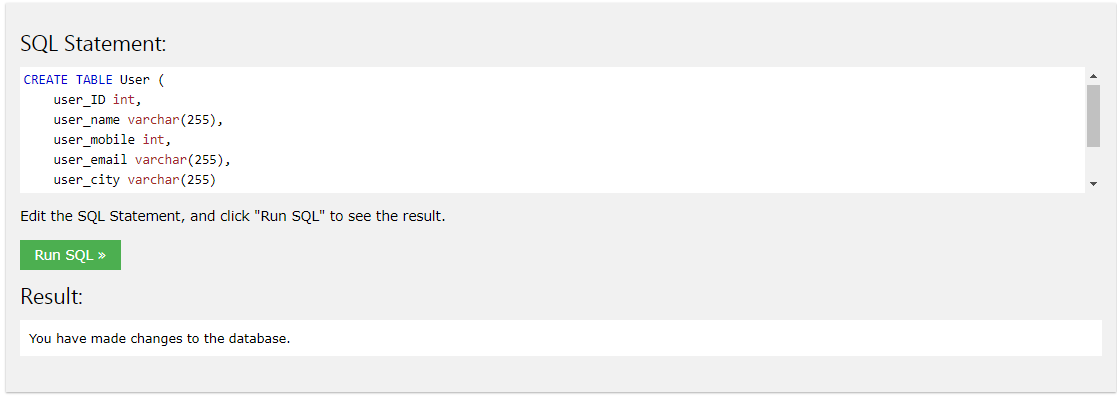
user\_name varchar(255),

user\_mobile int,

user\_email varchar(255),

user\_city varchar(255)

);



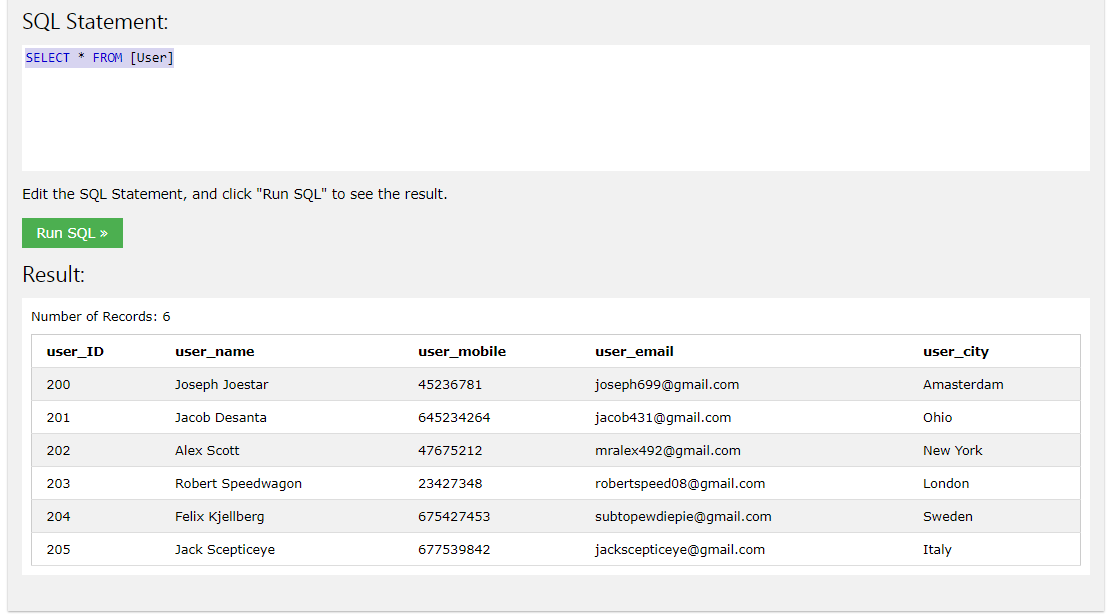
INSERT INTO User (user\_ID,user\_name,user\_mobile,user\_email,user\_city)

VALUES ('200','Joseph Joestar','45236781','joseph699@gmail.com','Amasterdam');



/\* After 5 statements \*/

SELECT \* FROM [User]



CREATE TABLE Patient (

pat\_ID int,

pat\_name varchar(255),

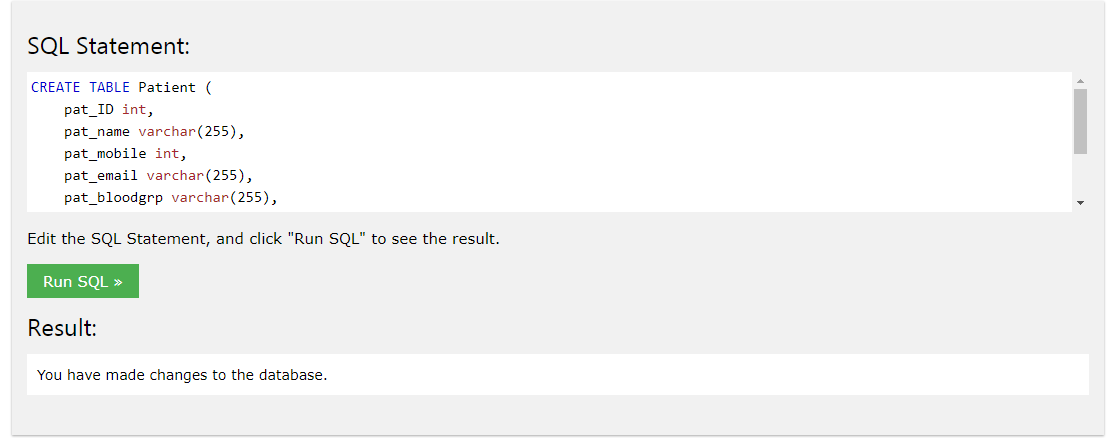
pat\_mobile int,

pat\_email varchar(255),

pat\_bloodgrp varchar(255),

pat\_city varchar(255)

);



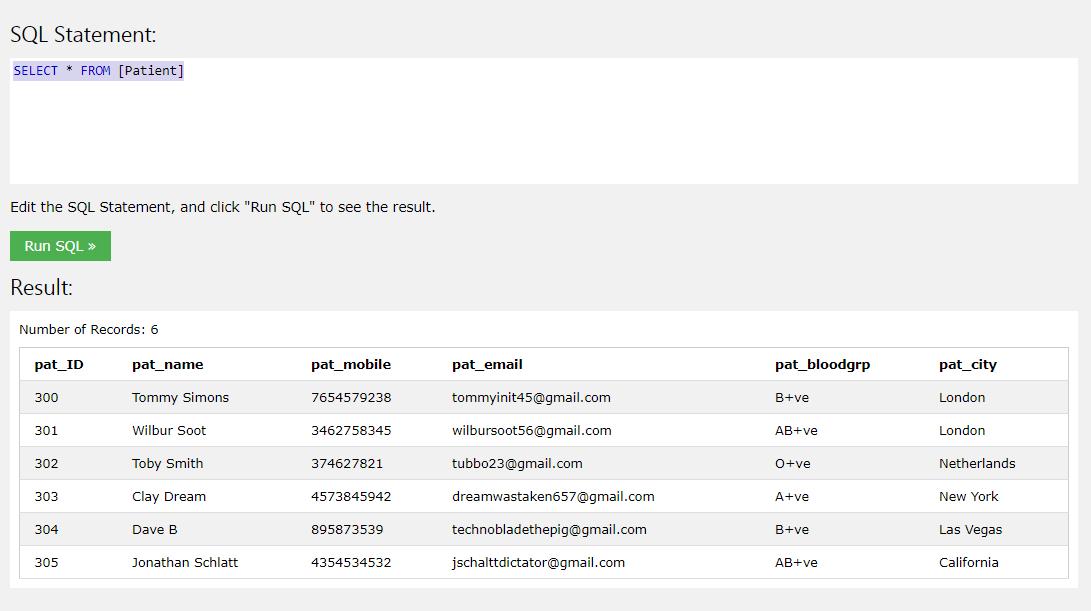
INSERT INTO Patient (pat\_ID,pat\_name,pat\_mobile,pat\_email,pat\_bloodgrp,pat\_city)

VALUES ('300','Tommy Simons','7654579238','tommyinit45@gmail.com','B+ve','London');



/\* After 5 statements \*/

SELECT \* FROM [Patient]



CREATE TABLE Donor (

donr\_ID int,

donr\_name varchar(255),

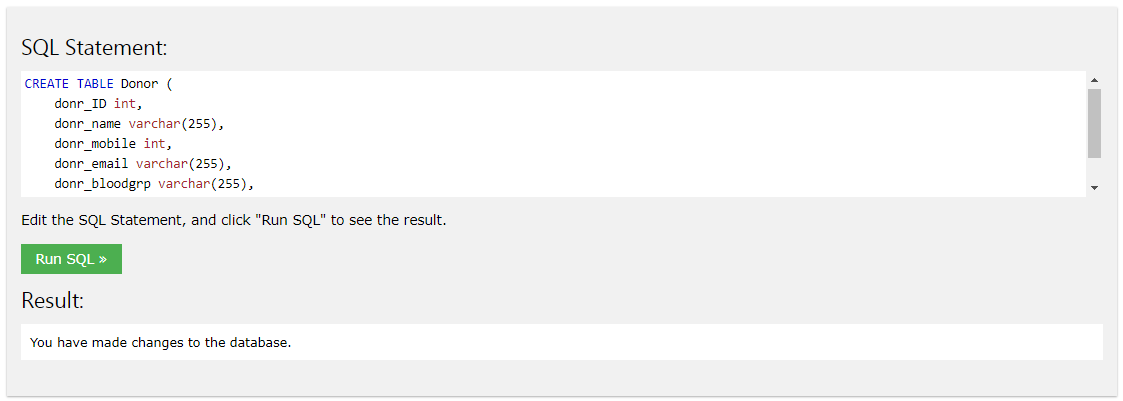
donr\_mobile int,

donr\_email varchar(255),

donr\_bloodgrp varchar(255),

donr\_city varchar(255)

);



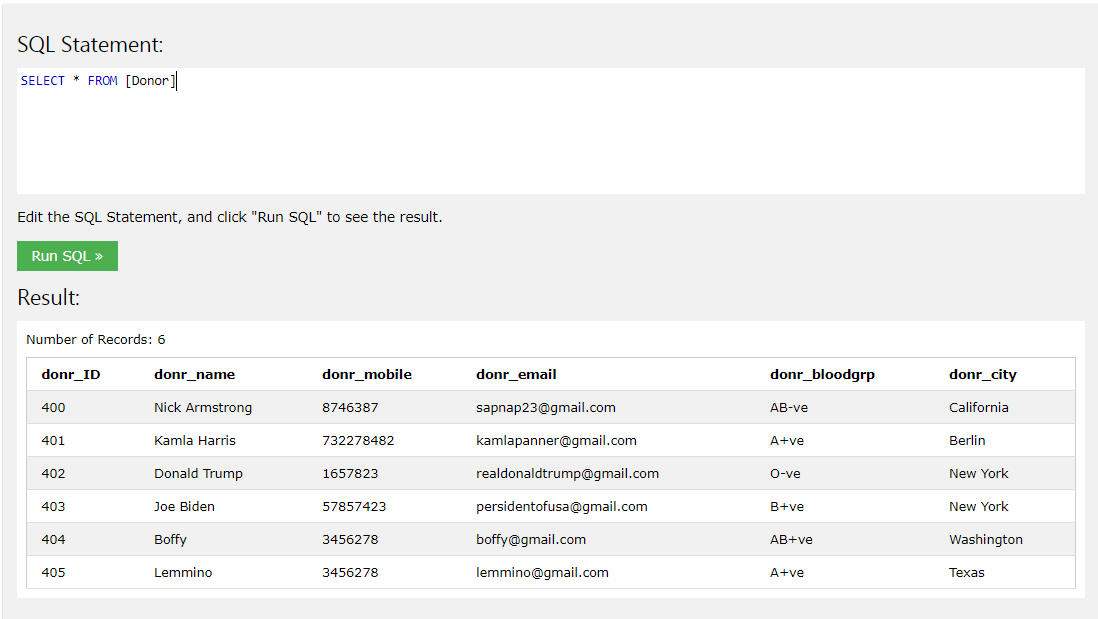
INSERT INTO Donor(donr\_ID,donr\_name,donr\_mobile,donr\_email,donr\_bloodgrp,donr\_city)

VALUES ('400','Nick Armstrong','8746387','sapnap23@gmail.com','AB-ve','California');



/\* After 5 statements \*/

SELECT \* FROM [Donor]



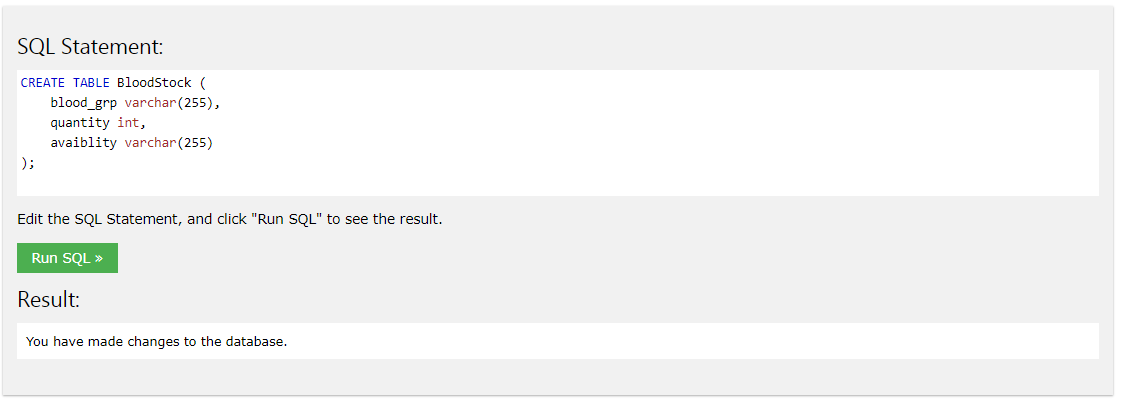
CREATE TABLE BloodStock (

blood\_grp varchar(255),

quantity int,

avaiblity varchar(255)

);



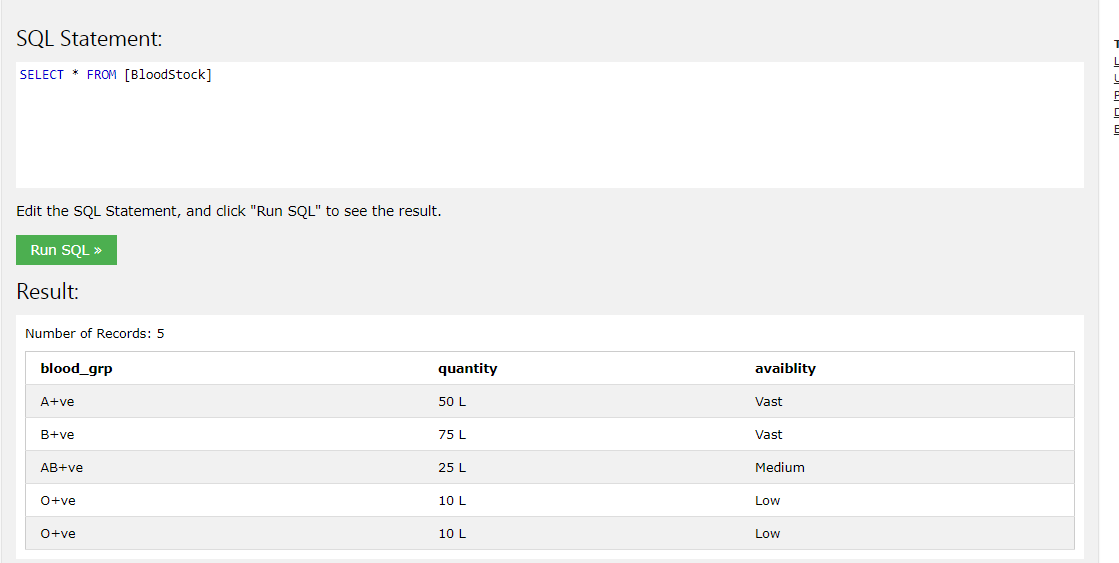
INSERT INTO BloodStock(blood\_grp,quantity,avaiblity)

VALUES ('A+ve','50 L','Vast');



/\* After 5 statements \*/

SELECT \* FROM [BloodStock]

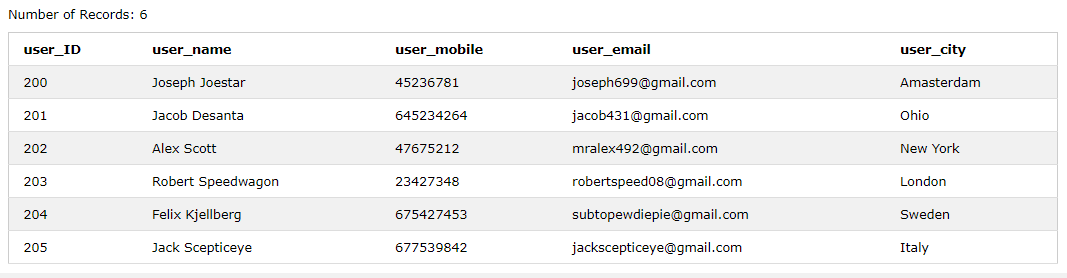


**Final Database**

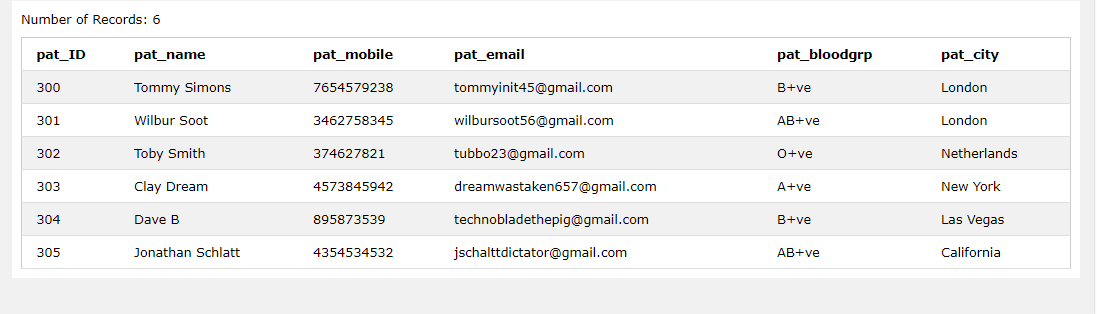
1. **Login Table:**

****

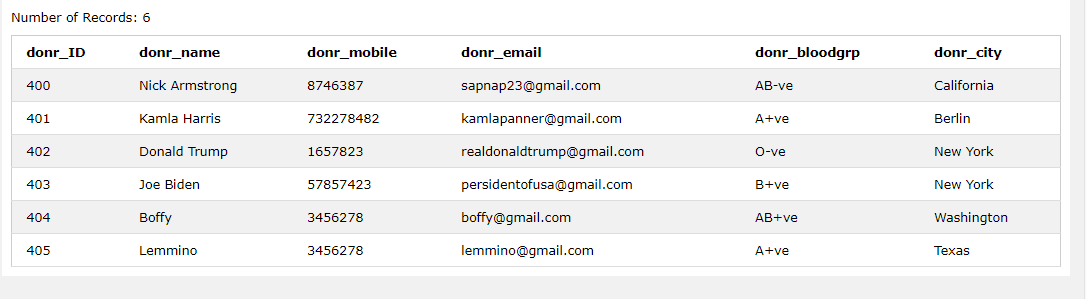
1. **User Table:**

****

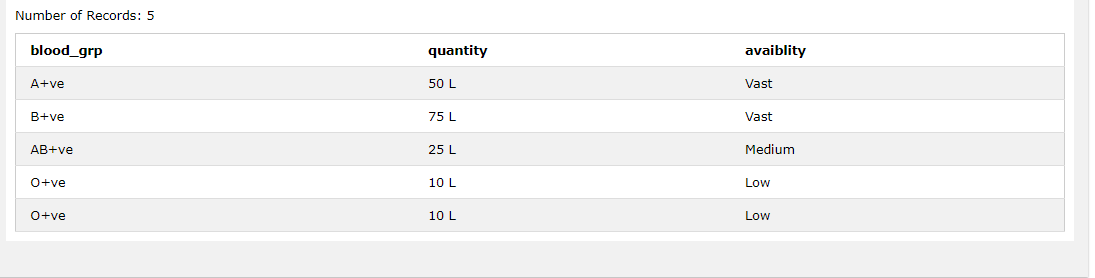
1. **Patient Table:**

****

1. **Donor Table:**

****

1. **Blood Stock**

****

**Applications**

Application of DBMS:

**Sector Use**

|  |  |
| --- | --- |
| Banking | For customer information, account activities, payments, deposits, loans, etc. |
| Airlines | For reservations and schedule information. |
| Universities | For student information, course registrations, colleges and grades. |
| Telecommunication | It helps to keep call records, monthly bills, maintaining balances, etc. |
| Finance | For storing information about stock, sales, and purchases of financial instruments like stocks and bonds. |
| Sales | Use for storing customer, product & sales information. |
| Manufacturing | It is used for the management of supply chain and for tracking production of items. Inventories status in warehouses. |
| HR Management | For information about employees, salaries, payroll, deduction, generation of paychecks, etc. |

**Conclusion**

Database Management System (DBMS) is a software for storing and retrieving users' data while considering appropriate security measures. It consists of a group of programs which manipulate the database. The DBMS accepts the request for data from an application and instructs the operating system to provide the specific data. In large systems, a DBMS helps users and other third-party software to store and retrieve data. DBMS allows users to create their own databases as per their requirement. The term “DBMS” includes the user of the database and other application programs. It provides an interface between the data and the software application. Traditionally, data was organized in file formats. DBMS was a new concept then, and all the research was done to make it overcome the deficiencies in traditional style of data management.

Database is a collection of related data and data is a collection of facts and figures that can be processed to produce information. Mostly data represents recordable facts. Data aids in producing information, which is based on facts. For example, if we have data about marks obtained by all students, we can then conclude about toppers and average marks. A database management system stores data in such a way that it becomes easier to retrieve, manipulate, and produce information.

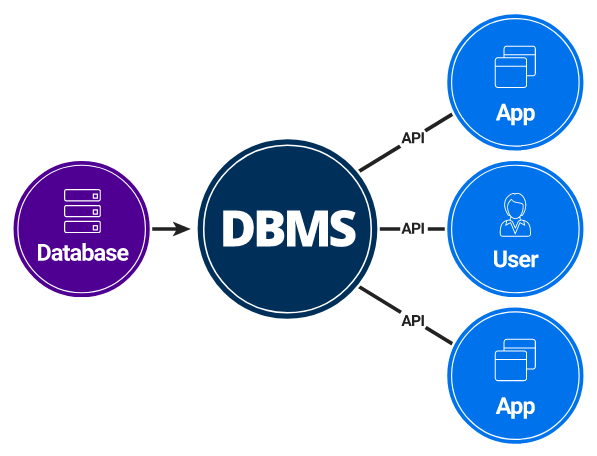


Fig.8. DBMS