Library Management System

In the beginning we hope that our project will be liked by you. We made a complete library management system which provides many advantages, easy to deal with, and has a high performance.

It's not like any traditional library, it also has a website to show our books, answer the clients' questions, take reviews, and provide a reader's community, in general, tracking the users.

In the next few lines, we will show to you our entities and the whole rules description.

The Entities

As known, Any Database must be built on some steps and starting with the ERD, The ERD is abbreviation for Entity Relationship Diagram… Here we will start with the Entities in our ERD…

Book:

It refers to the books in the library. The attributes for each book are Book ID (identifier), Title, ISBN, Publishment Year, Category, and Pages.

Location:

The location of the book.

The attributes for each location are Location ID (identifier), Shelf, and Section.

Author:

The writer of the book. The attributes for each author are Author ID (identifier), Name, Birth Year, and Dead Year.

Employee:

The workers of the library, whether staff members or librarians. The attributes for each employee are ID (identifier), Name, Hourly Rate, Fixed Salary, Gender, and Address.

User:

It refers to the clients who are readers, students, or professionals. Each client has attributes User ID (identifier), Frist Name, Last Name, Phone Number, Location (all the details), Borrowing period allowed, National ID, school / university, and Degree.

Record:

Here is every borrowing record or return record. The attributes for each record are Record ID (identifier), Date, OnTime State, and Return Deadline.

Account:

The data of accounts which belong to the website or the library system for users or staff members. The attributes for each account are Email (identifier), Password, Username, System access Number and Phone.

The Relationships

The second component of the ERD is the Relationships, it must be well determined as it will show the whole processes in the system and the whole range of each entity. It called the Business rules and here are the business rules in our system…

Business rules:

Business rules is determined while discussing with the project manager and the CEO of the Library to have the best result in our system so they will have the best database.

Here are the Rules:

1- One or many Authors Must write one or many Books
2- One Book Must be written by one or more Author

3- One Book Must have One Location
4- One location May have one Book

5- One User May borrow one or more Books
6- One Book May be borrowed by One User

7- One or many Books May be signed in One or many Record
8- One Record Must have One or Many books

9- One Staff Member Must Track One or many Users
10- One User Must be tracked by One Staff Member

11- One Staff member Must manage One or many Records
12- One Record Must be Managed by One Staff member

13- One Staff member Must have One Staff account

14- One User May have One User account

15- One User May assign One or many Records
16- One Record Must be assigned by One User

17- One Employee May supervises One or many Employees
18- One Employee May be Supervised by One Employee.

ERD IS IN THE NEXT PAGE

Enhanced ERD Model

Today the complexity of the data is increasing so it becomes more and more difficult to use the traditional ER model for database modeling. To reduce this complexity of modeling we must make improvements or enhancements were made to the existing ER model to make it able to handle the complex application in a better way.
Enhanced entity-relationship diagrams are advanced database diagrams very similar to regular ER diagrams which represent requirements and complexities of complex databases.
It is a diagrammatic technique for displaying the Sub Class and Super Class; Specialization and Generalization; Union or Category; Aggregation etc.

Generalization and Specialization –
These are very common relationships found in real entities. However, this kind of relationship was added later as an enhanced extension to the classical ER model. **Specialized** classes are often called **subclass** while a **generalized class** is called a superclass, probably inspired by object-oriented programming. A sub-class is best understood by **"IS-A analysis"**. Following statements hopefully makes some sense to your mind "Technician IS-A Employee", "Laptop IS-A Computer".

An entity is a specialized type/class of another entity. For example, a technician is a special Employee in a university system Faculty is a special class of Employee. We call this phenomenon generalization/specialization. In the example here Employee is a generalized entity class while the Technician and Faculty are specialized classes of Employee.

ENHANCED ERD IS IN THE NEXT PAGE

Relational Model

Relational Model (RM) represents the database as a collection of relations. A relation is nothing but a table of values. Every row in the table represents a collection of related data values. These rows in the table denote a real-world entity or relationship.
The table name and column names are helpful to interpret the meaning of values in each row. The data are represented as a set of relations. In the relational model, data are stored as tables. However, the physical storage of the data is independent of the way the data are logically organized.

- The Relational database modelling represents the database as a collection of relations (tables)
- Attribute, Tables, Tuple, Relation Schema, Degree, Cardinality, Column, Relation instance, are some important components of Relational Model
- Relational Integrity constraints are referred to conditions which must be present for a valid Relation approach in DBMS
- Domain constraints can be violated if an attribute value is not appearing in the corresponding domain, or it is not of the appropriate data type
- Insert, Select, Modify and Delete are the operations performed in Relational Model constraints
- The relational database is only concerned with data and not with a structure which can improve the performance of the model
- Advantages of Relational model in DBMS are simplicity, structural independence, ease of use, query capability, data independence, scalability, etc.

RELATIONAL MODEL IS IN THE NEXT PAGE

SQL Code

Create Table Author (Author_ID number Primary key, First_Name number(16) not null,

Last_name number(16) not null, Birth_year number(4) not null, Death_year number(4));

Author_ID	First_Name	Last_name	Birth_year	Death_year
12321	John	Steve	1925	1979
24321	Mary	Alfredo	1890	1945
34345	Chris	Joe	1988	null

Create Table Location(Location_ID number(16) Primary key, shelf number(4) not null,

Section number(4) not null);

Location_ID	Shelf	Section
00235	001	111
02353	045	23
34233	115	534

Create Table Book(Book_ID number(14) Primary key, Title varchar(64) Not null,

ISBN number(13) not null, Year_Published number(4) not null, Pages number(4) not null,

Location_ID number not null, FOREIGN KEY(Location_ID) REFERENCES Location(Location_ID));

Book_ID	Title	ISBN	Year_Published	Pages	Location_ID
0043	Jump	000343	2004	198	00235
3432	Power of habits	000532	1998	200	34233
1242	SQL	002343	2010	564	34445

Create Table Category(Category_ID number, Book_ID number, FOREIGN KEY(Book_ID) REFERENCES book(Book_ID), PRIMARY KEY(Category_ID, Book_ID));

Category_ID	Book_ID	PK
Self Help	3432	Self Help3432
Educational	1242	Educational1242
Coding	1242	Coding1242

As It will take long inserting the next tables will have only column names

Create Table Write(Author_ID number, FOREIGN KEY(Author_ID) REFERENCES Author(Author_ID),

Book_ID number, FOREIGN KEY(Book_ID) REFERENCES Book(Book_ID),

Release_Date date not null, Primary key(Author_ID, Book_ID));

Author_ID	Book_ID	Release_Date	Comp(PK)

Create Table Employee (Employee_ID number Primary key, Name varchar(64) Not Null,

Gender char(1) Not null check (Gender in ('M','F')),Address varchar(128) Not null);

Employee_ID	Name	Gender	Address

Create Table Staff_Member (Employee_ID number Primary key, FOREIGN KEY(Employee_ID) REFERENCES Employee(Employee_ID), Hourly_Rate number(5));

Employee_ID	Hourly_Rate

Create Table Librarian (Employee_ID number Primary key, FOREIGN KEY(Employee_ID) REFERENCES Employee(Employee_ID), Fixed_Salary number(5));

Employee_ID	Fixed_Salary

Create Table Account(Email varchar(320) Primary Key, Password varchar(256) Not null, Phone number(11) Unique Not null).

Email	Password	Phone

Create Table Staff_Account (Email varchar(320) Primary Key, FOREIGN KEY(Email) REFERENCES Account(Email), System_Access_Number varchar(64) Unique Not null, Employee_ID number, FOREIGN KEY(Employee_ID) REFERENCES Employee(Employee_ID));

Email	System_Acess_Number	Employee_ID

Create Table userr (user_id number not NULL PRIMARY KEY, first_name varchar(20) NOT NULL, last_name varchar(20) NOT NULL, city varchar(20) NOT NULL, state varchar(20) NOT NULL, zip_code number(5) NOT NULL, Address varchar(200), borrowing_period number(4) NOT NULL, employee_id number NOT NULL, FOREIGN KEY (employee_id) REFERENCES staff_member(employee_id),);

User_ID	First Name	Last Name	City	State	Zip	Address	Borrowing Period	Employee ID

Create Table phone_number (user_id number, phone_number number(11), FOREIGN KEY(user_id) REFERENCES userr(user_id), PRIMARY KEY(user_id, phone_number));

User_ID	Phone Number	Comp(PK)

Create Table record(record_id number(20) NOT null PRIMARY KEY, rec_date date NOT NULL, on_time_state bool check type in('Return_Record'), return_deadline date check type in('Borrow_Record'), type varchar check type in('Borrow_Record','Return_Record'), user_id number NOT NULL, employee_id number NOT NULL, FOREIGN KEY (user_id) REFERENCES userr(user_id), FOREIGN KEY (employee_id) REFERENCES staff_member(employee_id));

Record ID	Record Date	OnTime State	Return Deadline	Type	User ID	Employee ID

Create Table Signed_in (Book_ID number, FOREIGN KEY(Book_ID) REFERENCES Book(Book_ID), Record_ID number, FOREIGN KEY REFERENCES Record(Record_ID), signature varchar(64) not null, Primary key(Book_ID, Record_ID));

Book ID	Record ID	Signature	Comp(PK)

Create Table reader (national_id number not NULL, user_id number NOT NULL PRIMARY KEY, FOREIGN KEY (user_id) REFERENCES userr(user_id));

User ID	National ID

Create Table student (user_id number not NULL PRIMARY KEY, school varchar(32), university varchar(32), FOREIGN key (user_id) REFERENCES userr(user_id));

User ID	School	University

Create Table professional (user_id number NOT NULL PRIMARY KEY, degree_id varchar(32) NOT NULL, FOREIGN KEY (user_id) REFERENCES userr(user_id));

User ID	Degree ID

Create Table degree (user_id number, degree_id varchar(32),FOREIGN KEY(user_id) REFERENCES userr(user_id), PRIMARY KEY(user_id, degree_id));

Degree ID	User ID	Comp(PK)

Create Table user_account (Email varchar(320) PRIMARY KEY, user_name varchar(64) NOT NULL, user_id number NOT NULL, FOREIGN KEY (user_id) REFERENCES userr(user_id), FOREIGN KEY (Email) REFERENCES Account(Email));

Email	Username	User ID