ADVANCED DATABASE SYSTEMS

Project-UCM LIBRARY

Divya Vuppala

dxv46740@ucmo.edu

ClassID : 37

Table of Contents

Introduction	3
External Schema	4
E-R Diagram	5-7
Unnormalized Diagram	8
Normalization Explanation	9
Normalized Diagram	10
Data Dictionary	11-12
DBGen. Sql	13 - 18
View Example	19
Query Examples	20-24
Update Examples	25-26
Glossary	27
References	28

Introduction

Library is regarded as the brain of any institute, many institutes understand the importance of library to the growth of the institute and users(students). A library is a collection of organized information and resources which is made accessible to everyone. The collection of the resources and information are provided in digital or physical format in either a building or room.

Library system gives all detailed information about the students and books. It will track on the how many books issued to students and also how many all available. Library system offers many convenient features and flexible.

Library management system involves two kinds of users namely student and librarian. Both librarian and student can login in to the system using their credentials. A provider supplies books to the librarian and librarian organizes books in a sequence using book_id in various racks. A librarian has access to update book details. Students can take or return books with particular date of issue and date of return. If a student doesn't return the book within specified time, then the user needs to pay some amount of fine. The fine details are categorized based on the extra days from the date of return.

This database consists of tables login_info, student_details, provider_details , librarian, fine_details, book_info and book_issue_details. Primary keys are assigned to every table to which uniquely identify records. SQL server is used to retrieve the information by executing queries.

External Schema

Login Information: A librarian and student can login in to the system using their credentials the entire login related information is available in this entity.

Student Details: Students are the primary users of the library they request for a book in the library using their credentials in to the system. This entity consists of the details about the student in the university. The details included are name, city, phone number etc. Every student is uniquely identified by an Id which is the primary key. Every student can issue or return a book from the library based upon his membership status i.e., if it is not expired.

Librarian: Librarian initially checks the presence of book by login in to the system using librarian credentials. Librarian handles the overall operation of the library. He has an option to add or remove the books from the library. He keeps track of all the books and also adjusts the books in particular rows and updates the information. This helps a student to access the books easily and quickly.

Provider Details: Provider is the main person who supplies books to the librarian. Every provider is uniquely identified by a provider id.

Book Information: Books are the main resources of the library it contains all the information about the author, category, and version of the book. The primary key is the book id which can be used to uniquely identify a book.

Book Issue Details: The book information details contain the information about the issue date and the return date and many others. It also keeps track of the dates when the book is issued.

Fine Details: If the student doesn’t return the book by due date he/she should pay the penalty. This entity provides the information about various types of fine amounts present in the library.

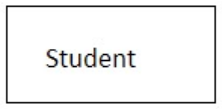
ENTITY RELATIONSHIP DIAGRAM:

An entity-relationship diagram (ERD) is a graphical representation of an information system that shows the relationship people, objects, places, events within that system. An ERD is a data modeling technique that can help define business processes and can be used as the foundation for a relational database.

There are three basic elements in the ER diagram.

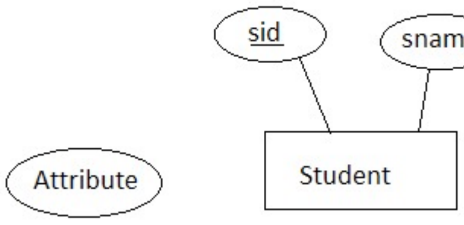
Entity

An entity can be a person, place, event or object.



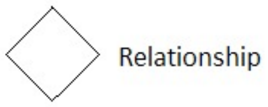
Attribute

An attribute is a property or characteristic of an entity, relationship or another attribute.

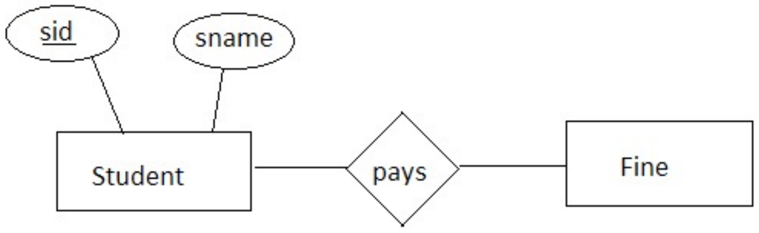


Relationship

A relationship describes how the entities interact.



Example:

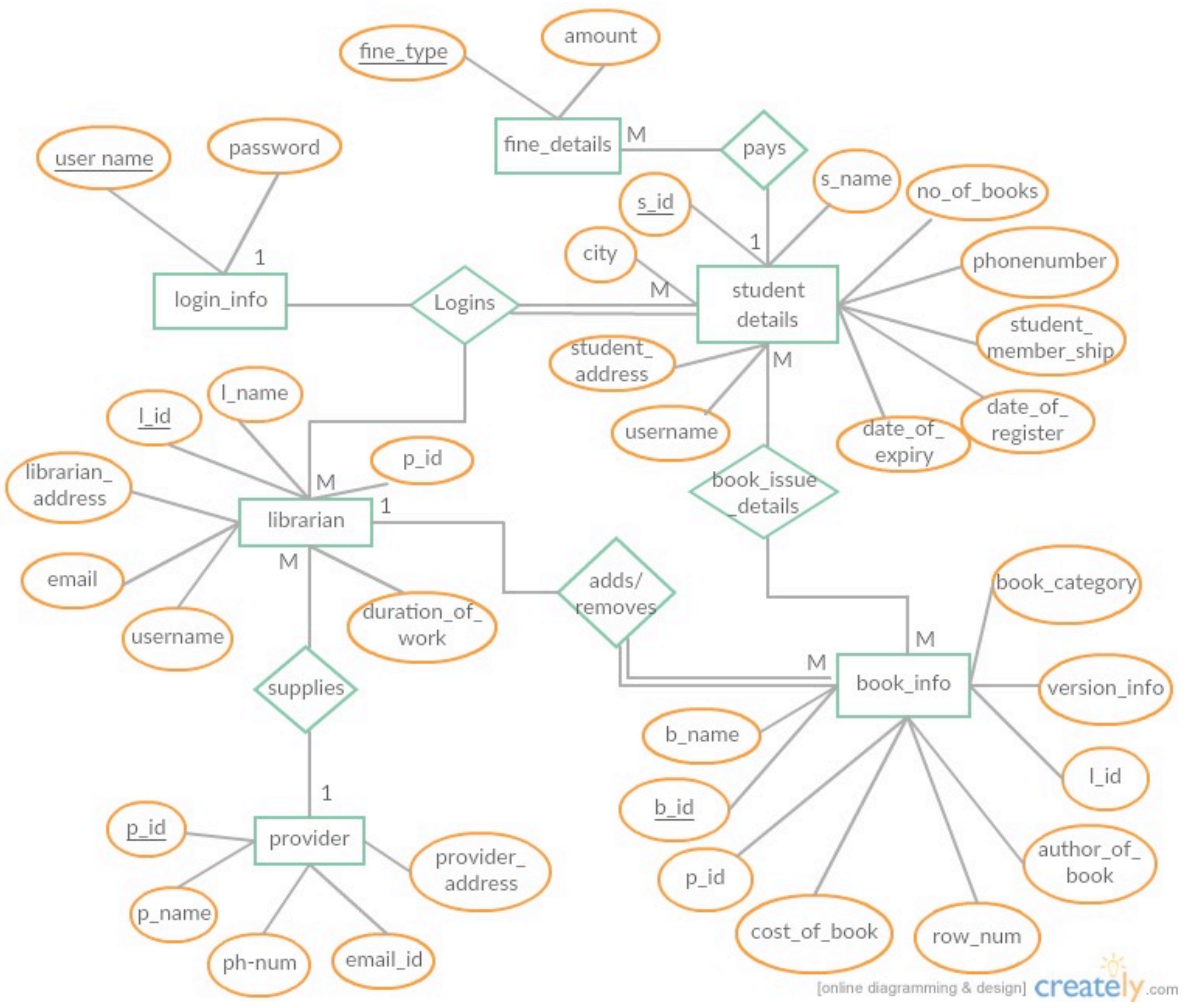


Three main components of an ERD are the entities, which are objects or concepts that can have data stored about them, the relationship between those entities and the cardinality which defines that relationship in terms of numbers.

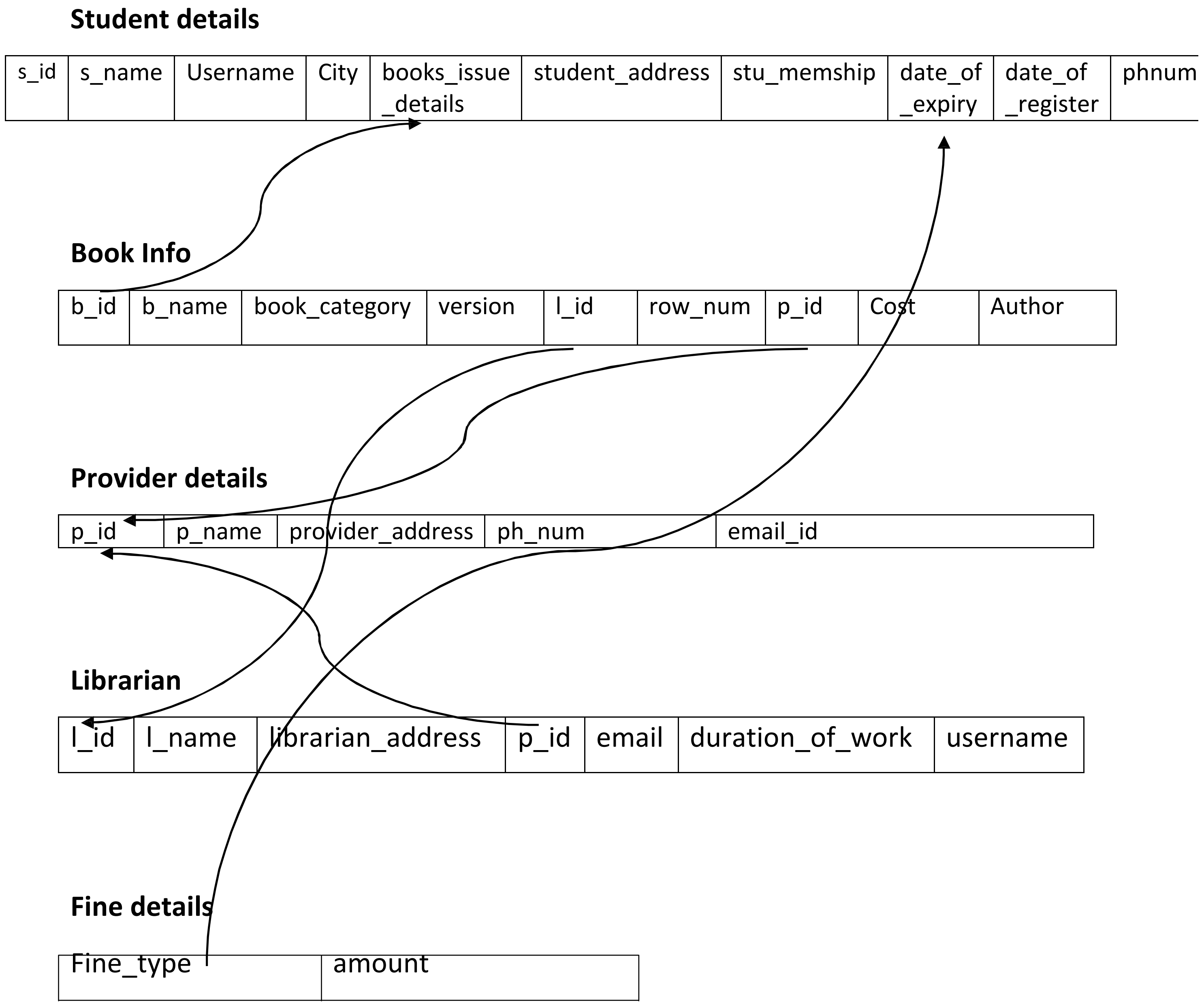
The three main cardinal relationships are:

1) **One-to-one:** One instance of an entity is associated with one other instance of another entity.
2) **One-to-many:** One instance of an entity is associated with zero, one or many instances of another entity but for one instance of entity there is only one instance of entity.
3) **Many-to-Many:** One instance of an entity is associated with one, zero or many instances of another entity and one instance of entity is associated with one , zero or many instances of entity.

E-R Diagram


fine_type
amount
fine_details
M
pays
user name
password
s_name
no_of_books
s_id
1
1
city
phonenumber
login_info
Logins
M
student details
student_ member_ship
student_ address
M
l_name
username
date_of_ register
l_id
date_of_ expiry
p_id
librarian_ address
M
book_issue _details
1
librarian
M
adds/ removes
book_category
email
duration_of_ work
username
M
M
supplies
book_info
version_info
b_name
1
l_id
b_id
p_id
provider
provider_ address
author_of_ book
p_id
p_name
cost_of_book
row_num
ph-num
email_id
[online diagramming & design] creately.com

Unnormalized Diagram



Student details

s_id	s_name	Username	City	books_issue _details	student_address	stu_memship	date_of _expiry	date_of _register	phnum

Book Info

b_id	b_name	book_category	version	l_id	row_num	p_id	Cost	Author

Provider details

p_id	p_name	provider_address	ph_num	email_id

Librarian

l_id	l_name	librarian_address	p_id	email	duration_of_work	username

Fine details

Fine_type	amount

Normalization Explanation:

1NF: A database is in first normal form if it satisfies both conditions:

Contains only atomic values
There are no repeating groups

2NF: A database is in second normal form if it satisfies the conditions:

It should be in first normal form.
All non-key attributes are functional dependent on the primary key.

3NF: A database is in third normal form it is satisfies the conditions:

It should be in second normal form.
There is no transitive functional dependency.

Student details

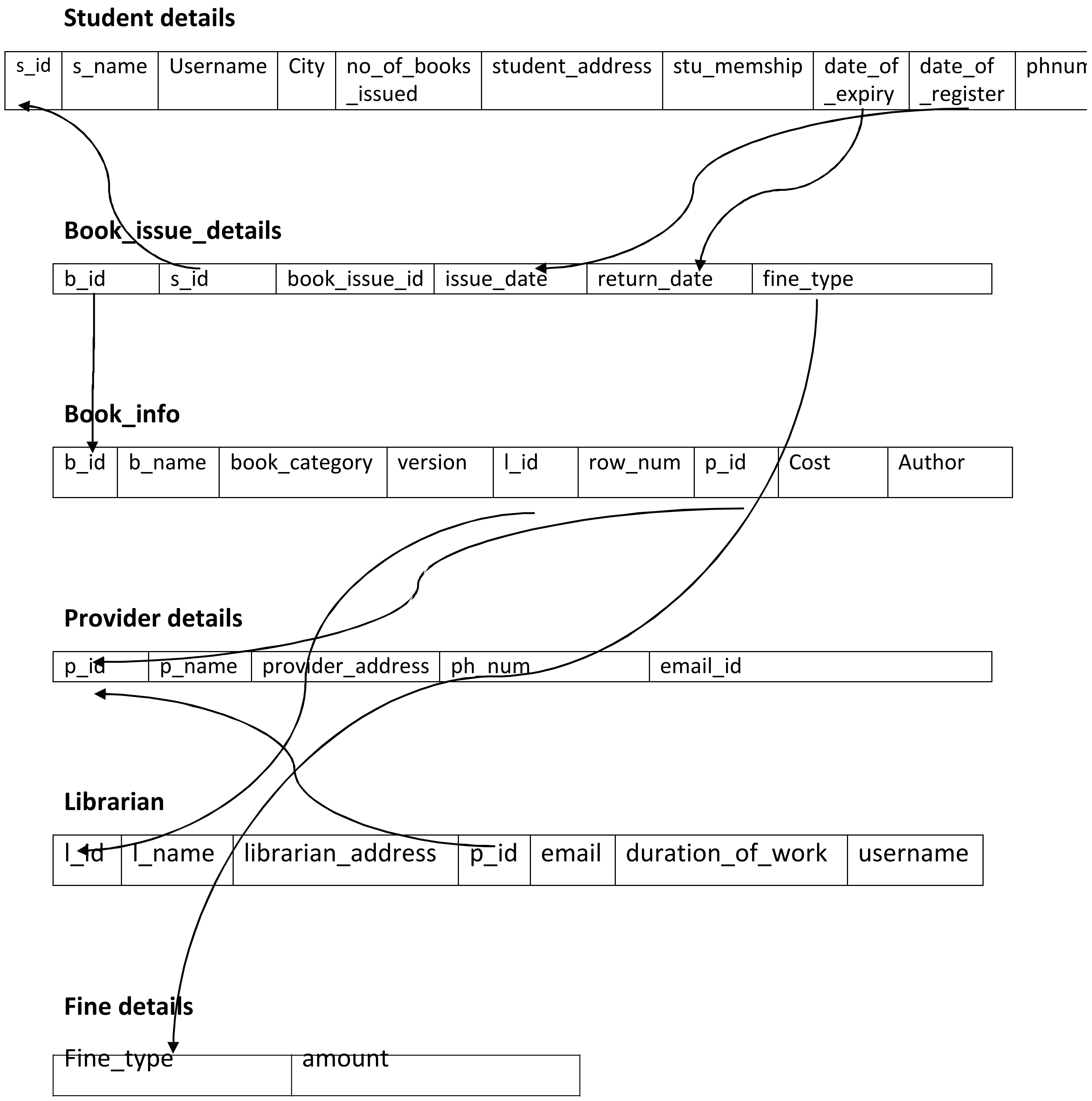
s_id	s_name	Username	City	book_issue _details	student_address	stu_memship	date_of _expiry	date_of _register	phnum

We need to split book_issue details into its own table because it partially depends on primary key.

Book_issue_details

b_id	s_id	book_issue_id	issue_date	return_date	fine_type

Normalized Diagram



Student details

s_id	s_name	Username	City	no_of_books _issued	student_address	stu_memship	date_of _expiry	date_of _register	phnum

Book_issue_details

b_id	s_id	book_issue_id	issue_date	return_date	fine_type

Book_info

b_id	b_name	book_category	version	l_id	row_num	p_id	Cost	Author

Provider details

p_id	p_name	provider_address	ph_num	email_id

Librarian

l_id	l_name	librarian_address	p_id	email	duration_of_work	username

Fine details

Fine_type	amount

Data Dictionary:

Login_info :

Username	Password
Varchar(20)	Varchar(20)
UniqueID for Login	Password for Login

Student_details :

s_id	s_name	no_of_books	City	phonenumber	student_ address	Username
Varchar(20)	Varchar(20)	Int	Varchar(20)	Varchar(20)	Varchar(20)	Varchar(20)
UniquID for student	Student name	No.of books	Student city	Student Phonenumber	Address	Username to login

Date_of_register	Date_of_expiry	Student_member_ship
Varchar(20)	Varchar(20)	Varchar(20)
Register date	Expiry date	Student membership

Librarian :

l_id	l_name	librarian_address	p_id	Email	Duration_of _work	Username
Varchar(20)	Varchar(20)	Varchar(20)	Varchar(20)	Varchar(20)	Int	Varchar(20)
UniqueID for librarian	Librarian name	Address of librarian	Provider id	Email	Duration	Username to login

Book_info :

b_id	b_name	book_category	author_of_book	version_info	row_num	p_id	cost_of_book	l_id
Varchar(20)	Varchar(20)	Varchar(20)	Varchar(20)	Varchar(20)	Varchar(20)	Varchar(20)	Int	Varchar(20)
Book ID	Book name	Book category	Author of book	Version of book	Row num	Provider id	Cost of book	Librarian ID

Provider_details:

p_id	p_name	provider_address	ph_number	email_id
Varchar(20)	Varchar(20)	Varchar(20)	Varchar(20)	Varchar(20)
UniqueID for provider	Provider name	Provider address	Phone number of provider	Email address

Book_issue_details:

b_id	s_id	book_issue_id	Issue_id	return_date	fine_type
Varchar(20)	Varchar(20)	Varchar(20)	Varchar(20)	Varchar(20)	Varchar(20)
Book id	Student id	Book Issue id	Issue id	Return date	Fine category

Fine_details:

fine_type	Amount
Varchar(20)	Int
UniqueID for fine type	Amount

DBGen.sql

```
create table login_info(
        username varchar(20) primary key,
        pass_word varchar(20));

create table student_details(
        s_id varchar(20) primary key,
        s_name varchar(20),
        no_of_books_issued int,
        city varchar(20),
        phonenumber varchar(20),
        student_address varchar(20),
        username varchar(20),
        date_of_register varchar(20),
        date_of_expriry varchar(20),
        student_membership_status varchar(20),
        foreign key(username) references login_info);

create table book_info(
        b_id varchar(20) primary key,
        b_name varchar(20),
        book_category varchar(20),
        author_of_book varchar(20),
        version_info varchar(20),
        row_num varchar(20),
        p_id varchar(20),
        cost_of_book int,
        l_id varchar(20),
        foreign key(p_id) references provider_details,
        foreign key(l_id) references librarian );

create table provider_details(
        p_id varchar(20) primary key,
        p_name varchar(20),
        provider_address varchar(20),
        ph_num varchar(20),
        email_id varchar(20));

create table book_issue_details(
        b_id varchar(20), s_id
        varchar(20), book_issue_id
        varchar(20), issue_date
        varchar(20), return_date
        varchar(20), fine_type
        varchar(20),
        primary key(b_id,s_id,book_issue_id),
        foreign key(b_id) references book_info, foreign
        key(s_id) references student_details, foreign
        key(fine_type) references fine_details);

create table librarian(
        l_id varchar(20) primary key,
        l_name varchar(20),
        librarian_address varchar(20),
        p_id varchar(20),
        email varchar(20),
        duration_of_work int,
        username varchar(20),
        foreign key(p_id) references provider_details,
        foreign key(username) references login_info);

create table fine_details(
        fine_type varchar(20) primary key,
        amount int);

insert into login_info values('one','5676887');
insert into login_info values('two','4545545');
insert into login_info values('three','543525');
insert into login_info values('four','3242314');
insert into login_info values('five','3435454');
insert into login_info values('six','012345');
insert into login_info values('seven','034455');
insert into login_info values('eight','644333');
insert into login_info values('nine','434343');
insert into login_info values('ten','656565');
insert into login_info values('eleven','434343');
insert into login_info values('twelve','434343');
insert into login_info values('thirteen','656565');

insert into student_details
values('S01','Divya',3,'HYD','9434354545','Vanasthalipuram','one','2013-02-12','2014-02-
12','Temporary');
```

```
insert into student_details values('S02','Sravan',4,'PUN','9324354546','Ambedkar','one','2013-
02-12','2014-02-12','Temporary');

insert into student_details values('S03','Madhu',6,'DEL','9234574324','Harrison','two','2013-02-
12','2013-02-12','Expired');

insert into student_details values('S04','Vasu',8,'KOL','9838477473','Charlotte','twelve','2013-
02-12','2014-02-12','Temporary');

insert into student_details values('S05','Akhila',3,'HYD','9536477274','Troost','four','2013-02-
12','2014-02-12','Permanent');

insert into student_details values('S06','Architha',1,'BOM','9847355353','Wornall','seven','2013-
02-12','2014-02-12','Temporary');

insert into student_details values('S07','Sowmya',4,'CHN','9667363233','Gregory','eight','2013-
02-12','2015-02-12','Expired');

insert into student_details values('S08','Sindhu',6,'CHN','9744756434','Meyer','three','2013-02-
12','2020-02-12','Permanent');


insert into book_info values('B01','Advanced
Algo','algorithms','park','V02','R01','P001',564,'L01');

insert into book_info values('B02','Advanced
Computer','networking','shin','V13','R02','P003',646,'L02');

insert into book_info values('B03','Adv Db','database','paul','V5','R03','P004',654,'L04');

insert into book_info values('B04','Compiler
Design','networking','james','V06','R04','P005',600,'L03');

insert into book_info values('B05','Big Data','database','david','V08','R05','P006',750,'L01');

insert into book_info values('B06','Big Data
Analytics','database','bond','V05','R05','P007',800,'L04');

insert into book_info values('B07','Parallel Algo','algorithms','lee','V06','R01','P009',543,'L04');

insert into book_info values('B08','Mobile
Android','mobile','ramesh','V02','R02','P001',900,'L03');

insert into book_info values('B09','Mobile iOS','mobile','suresh','V01','R03','P004',434,'L05');


insert into provider_details values('P001','Ayush','bihar','9435847584','ayush@yahoo.com');
insert into provider_details values('P002','Surabhi','ranchi','9786476366','sur@gmail.com');

insert into provider_details
values('P003','Shambhavi','kolkata','9673456774','shamba@yahoo.com');

insert into provider_details
values('P004','Saumita','kolkata','9241412455','saumi@hotmail.com');

insert into provider_details values('P005','Abhishek','bhopal','9654435435','abhi@gmail.com');

insert into provider_details values('P006','Vinitha','gujarat','9452355552','vini@yahoo.com');

insert into provider_details values('P007','Geeta','kanpur','9465656563','geeta@gmail.com');

insert into provider_details
values('P008','Dhanush','bangalore','9056635353','dhanu@gmail.com');

insert into provider_details
values('P009','Ruchir','chattisgarh','9132344557','ruchi@yahoo.com');

insert into provider_details values('P010','Aastha','delhi','9746464664','aastha@gmail.com');


insert into book_issue_details values('B05','S01','1','2013-05-01','2013-06-01','F8');

insert into book_issue_details values('B06','S04','2','2013-06-02','2013-07-02','F2');

insert into book_issue_details values('B04','S08','3','2013-07-03','2013-08-02','F1');

insert into book_issue_details values('B05','S05','4','2013-08-16','2013-09-017','F4');

insert into book_issue_details values('B09','S04','5','2013-05-14','2013-06-015','F4');

insert into book_issue_details values('B03','S02','6','2013-07-05','2013-08-06','F6');

insert into book_issue_details values('B04','S01','7','2013-03-06','2013-04-07','F7');

insert into book_issue_details values('B02','S07','8','2013-04-07','2013-05-08','F8');


insert into librarian values('L01','Srinivas','chaitanyapuri','P002','srini@gmail.com',6,'four');

insert into librarian values('L02','Radhika','malakpet','P001','radhi@yahoo.com',6,'five');

insert into librarian values('L03','Deeraj','warangal','P004','deeru@hotmail.com',3,'six');

insert into librarian values('L04','Gopi','nalgonda','P005','gopi@gmail.com',4,'four');

insert into librarian values('L05','Sunitha','karimnagar','P001','sunitha@yahoo.com',2,'two');
insert into fine_details values('F1',0);

insert into fine_details values('F2',20);

insert into fine_details values('F3',50);

insert into fine_details values('F4',100);

insert into fine_details values('F5',150);

insert into fine_details values('F6',200);

insert into fine_details values('F7',250);

insert into fine_details values('F8',300);

insert into fine_details values('F9',320);

insert into fine_details values('F10',340);

insert into fine_details values('F11',360);

insert into fine_details values('F12',400);

insert into fine_details values('F13',450);
```

View Example

A view is needed to see all the student membership status in the library. This view is a tool which provides access to see the users whose status is temporary or permanent. In addition to this, this created view allows librarians to view the details of the students who are taking/returning the books to the library from the various racks.

This view returns the names of the students and their membership status.

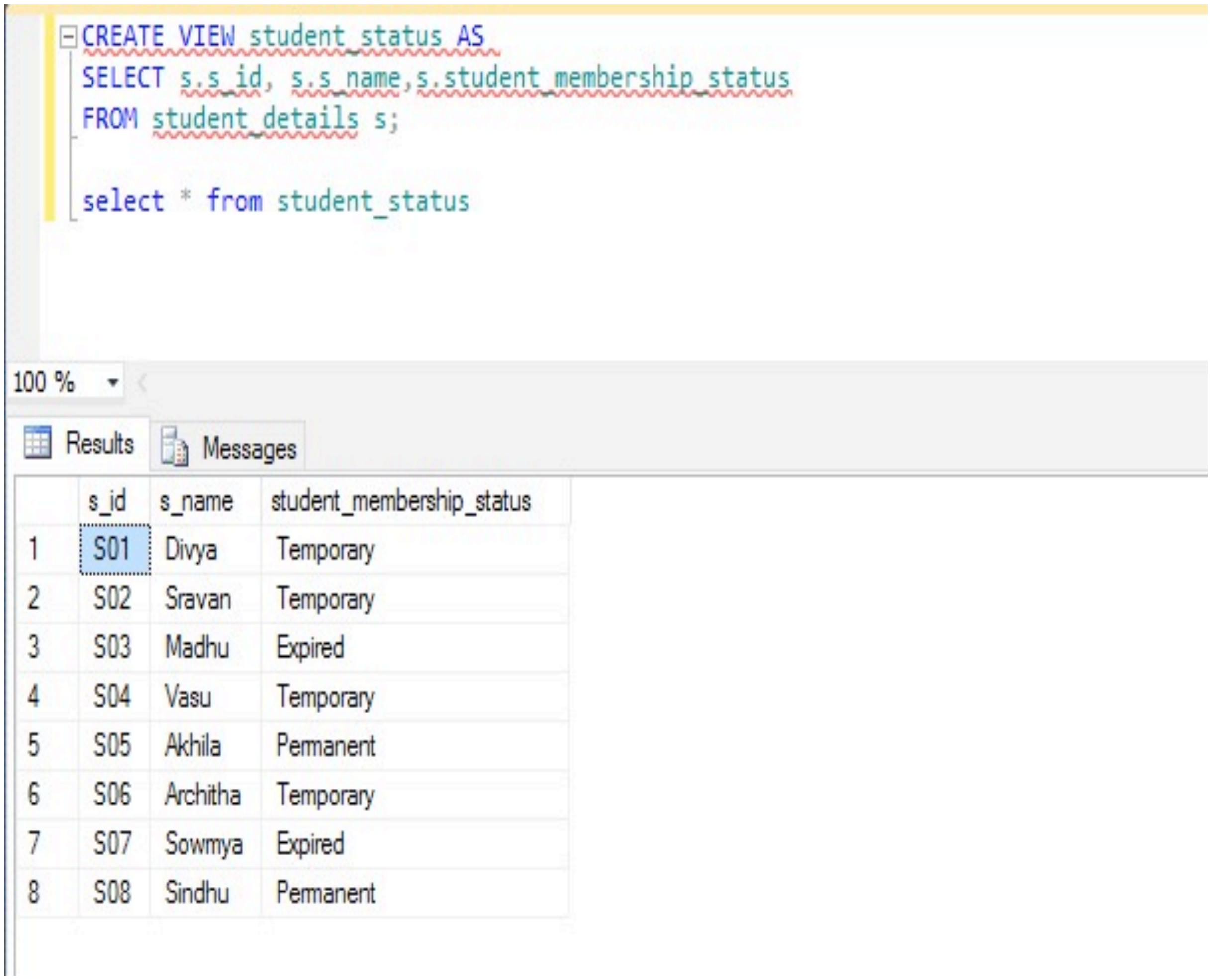
Note: This query assumed that the user has given the table name from which we can find the status of the students.

CREATE VIEW student_status AS

SELECT s.s_id, s.s_name,s.student_membership_status

FROM student_details s;

SELECT * FROM student_status;



```
CREATE VIEW student_status AS
SELECT s.s_id, s.s_name,s.student_membership_status
FROM student_details s;

select * from student_status
```

	s_id	s_name	student_membership_status
1	S01	Divya	Temporary
2	S02	Sravan	Temporary
3	S03	Madhu	Expired
4	S04	Vasu	Temporary
5	S05	Akhila	Permanent
6	S06	Architha	Temporary
7	S07	Sowmya	Expired
8	S08	Sindhu	Permanent

Query Examples: Query 1

Few keywords can be used to search the students with a certain keyword. This helps the users to find something by searching based upon this keyword. Additionally, this may be used to find more of the student details.

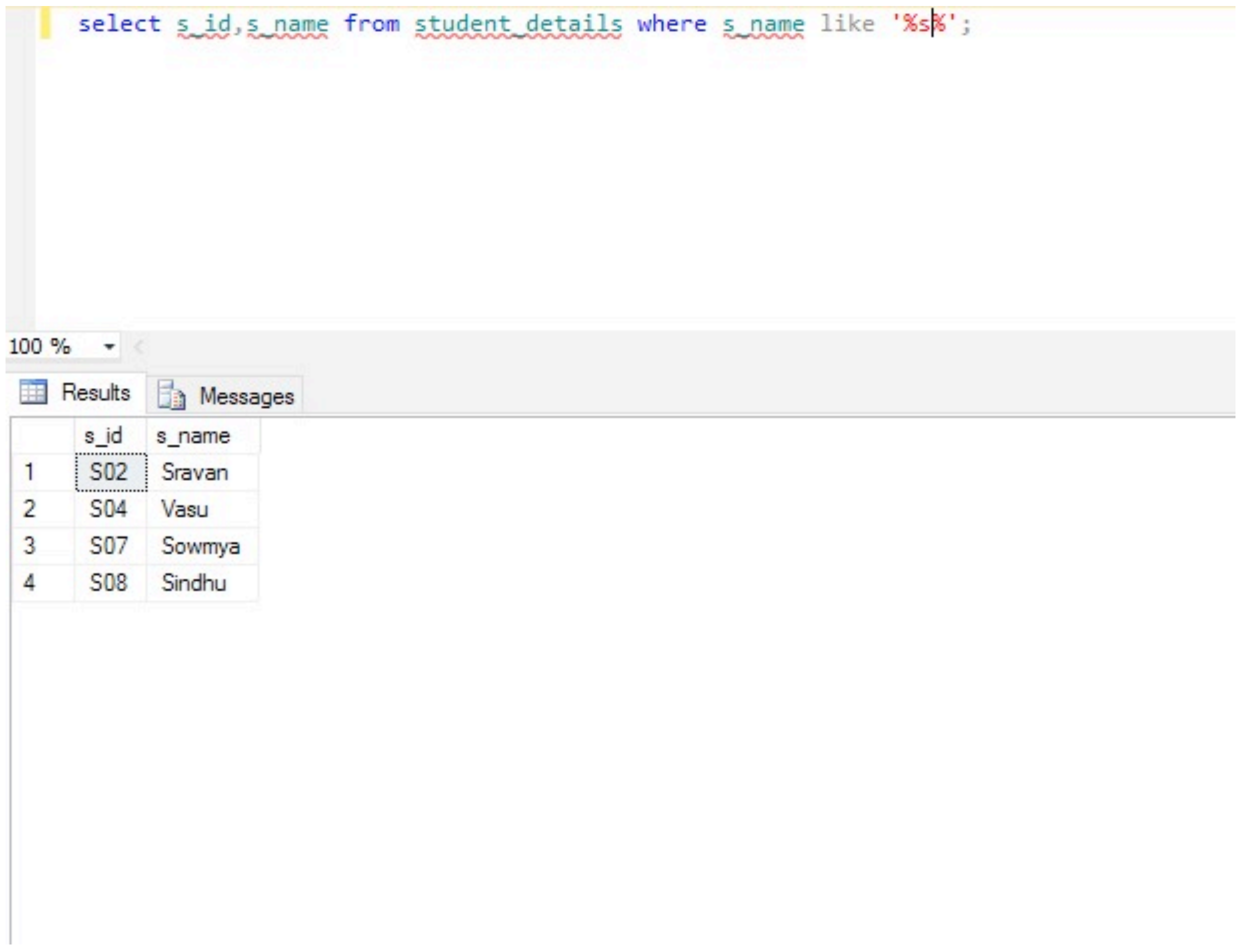
Note: This query assumes the user has provided a keyword and based upon this 'keyword' we find the results. SELECT

s_id,s_name FROM

student_details s

WHERE s,name LIKE '%keyword%';

To execute and find the results, the keyword used was 'S'.



```
select s_id,s_name from student_details where s_name like '%s%';
```

	s_id	s_name
1	S02	Sravan
2	S04	Vasu
3	S07	Sowmya
4	S08	Sindhu

Query Examples: Query 2

The next common query will be to find out the total sum of the amount of the various types of the find types and also find the total number of types of the fine types present in this library management system. It helps the user by displaying the total amount of fine ranges. so the result displays the sum and number of values for which this sum is calculated.

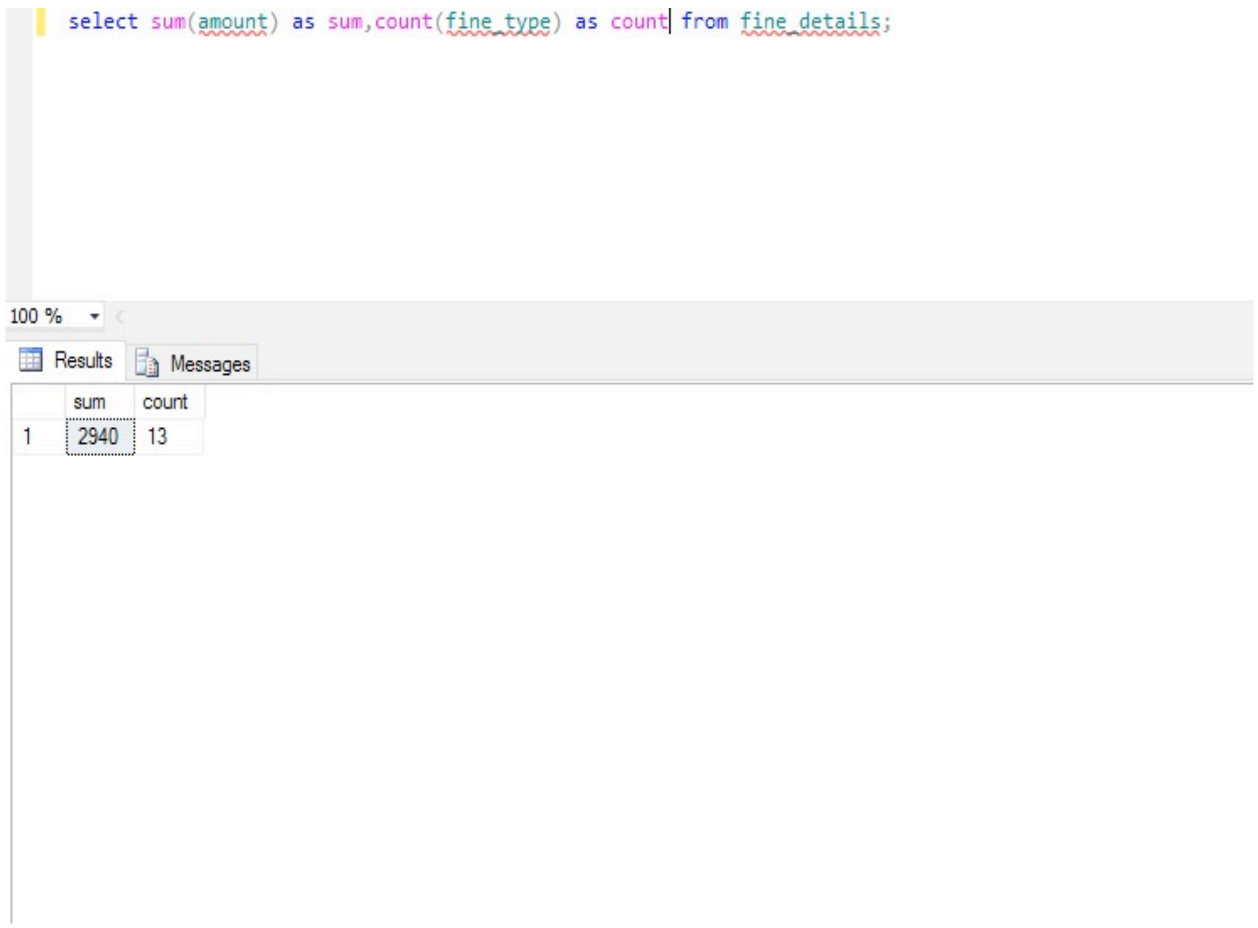
This query might look easy but it helps the users to find the various types of fine ranges. It helps in avoiding users to calculate each and every record manually.

Note: The query runs on the table fine_details. sum and count are the main keywords used to retrieve the results.

SELECT sum(AMOUNT ATTRIBUTE) as sum, count(FINE_TYPE) as count

FROM fine_details;

To display the resulting attribute name 'as' keyword is used and the column names are given as sum and count in the above query.



```
select sum(amount) as sum,count(fine_type) as count from fine_details;
```

```
   sum   count
1  2940  13
```

Query Examples: Query 3

The next common query will be to find the number of books whose values lie between given fixed values. This helps to find out books in some particular range. If a user wants to have a look at the books which are of low cost then he can enter less values and check if any books are available. 'between' is the word used to find the results between some particular range.

This might appear very simple, but it is very useful. It helps to avoid the random guessing and helps to find the accurate results. Here in this example we find the details of the books whose cost of book lies between 400 and 600.

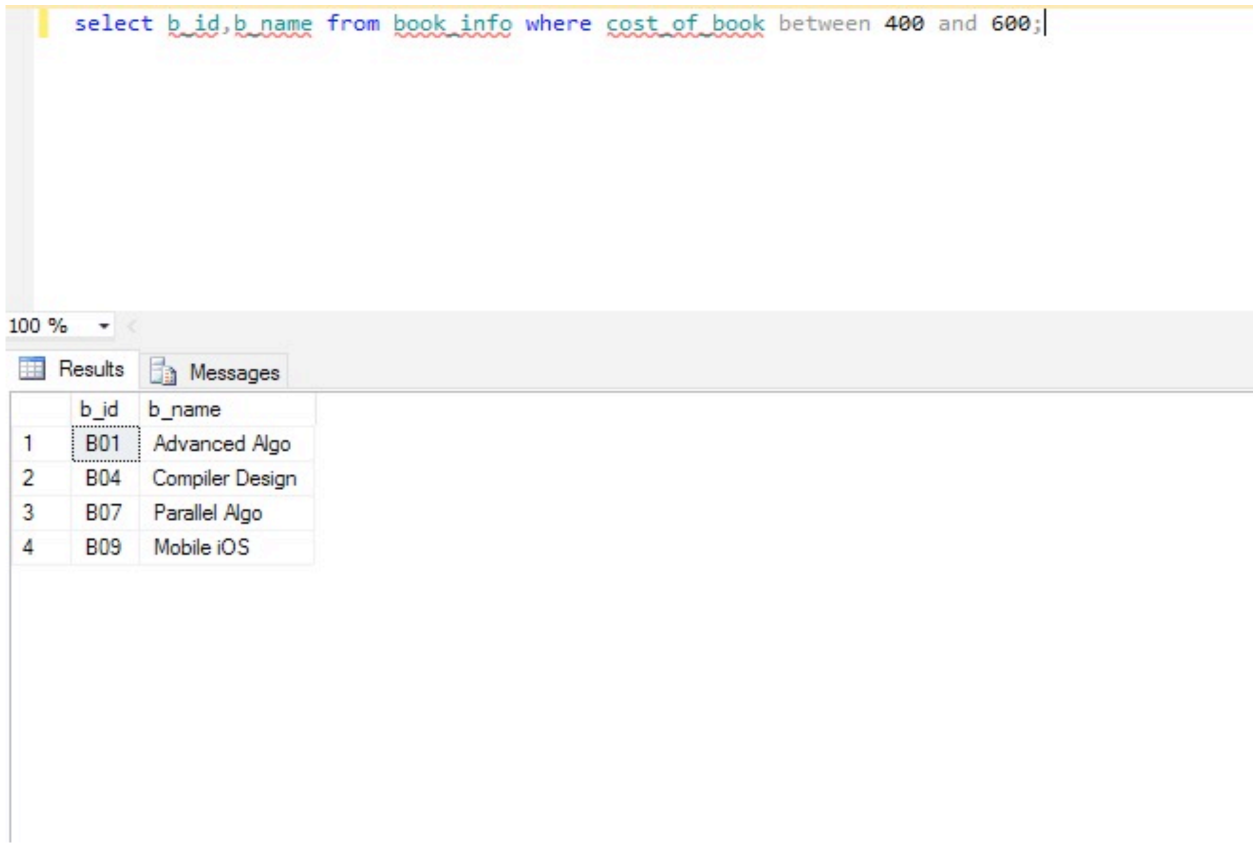
Note: This Query is applied to all the books available in the library.

SELECT b_id,b_name

FROM book_info

WHERE cost_of_book between value1 and value2;

In this Query the value1 and value2 are 400 and 600 respectively.



```
select b_id,b_name from book_info where cost_of_book between 400 and 600;
```

	b_id	b_name
1	B01	Advanced Algo
2	B04	Compiler Design
3	B07	Parallel Algo
4	B09	Mobile iOS

Query Examples: Query 4

The fourth Query involves finding out the details of the student who has not issued any book till date and also from the those results we are eliminating the student with some specific student id.

It gives the details of the student whether they have issued or not and to cross check the student details in the table.

Note: The tables used are student_details and book_issue details.

SELECT s_id from student_details

WHERE NOT EXISTS

(SELECT *

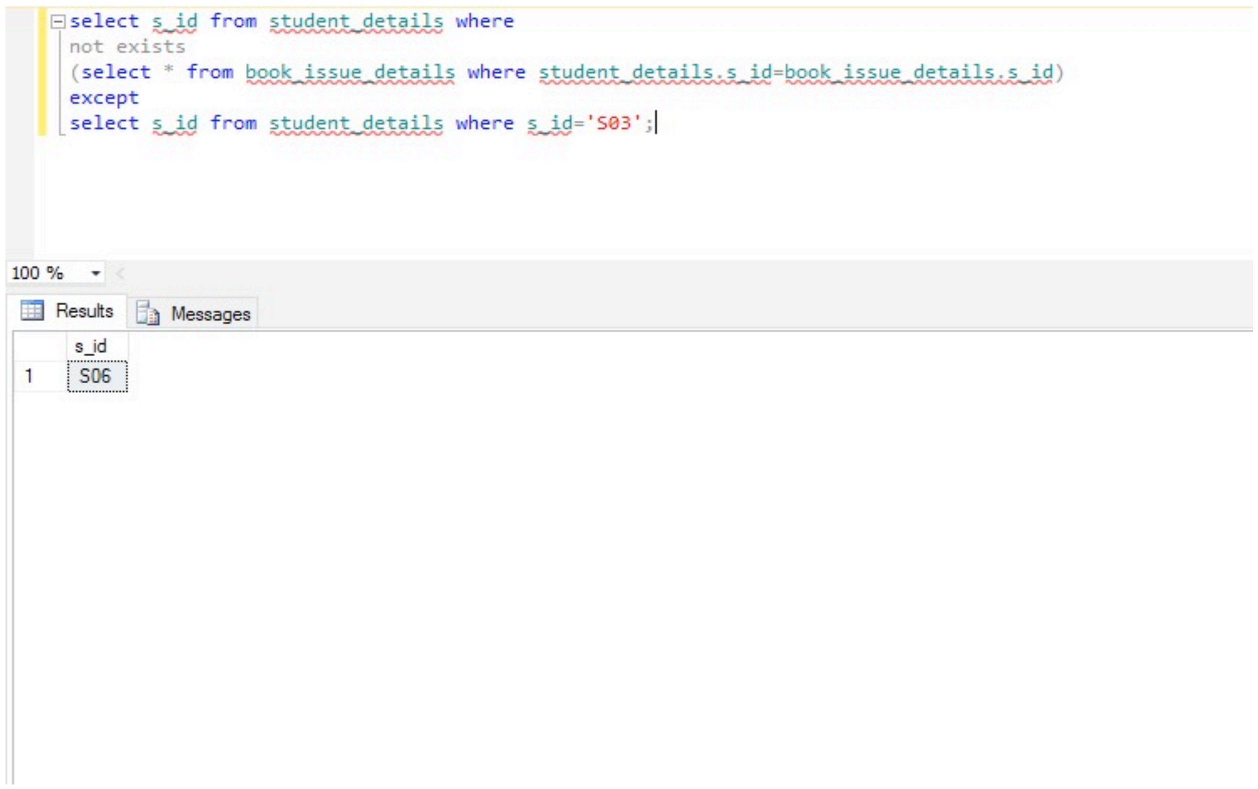
FROM book_issue_details

WHERE student_details.s_id=book_issue_details.s_id)

EXCEPT

(SELECT s_id FROM student_details WHERE s_id='S03');

In order to get the correct result the user with the exception used was 'S03'.



```
select s_id from student_details where
not exists
(select * from book_issue_details where student_details.s_id=book_issue_details.s_id)
except
select s_id from student_details where s_id='S03';
```

100 %

Results | Messages

```
   s_id
1  S06
```

Query Examples: Query 5

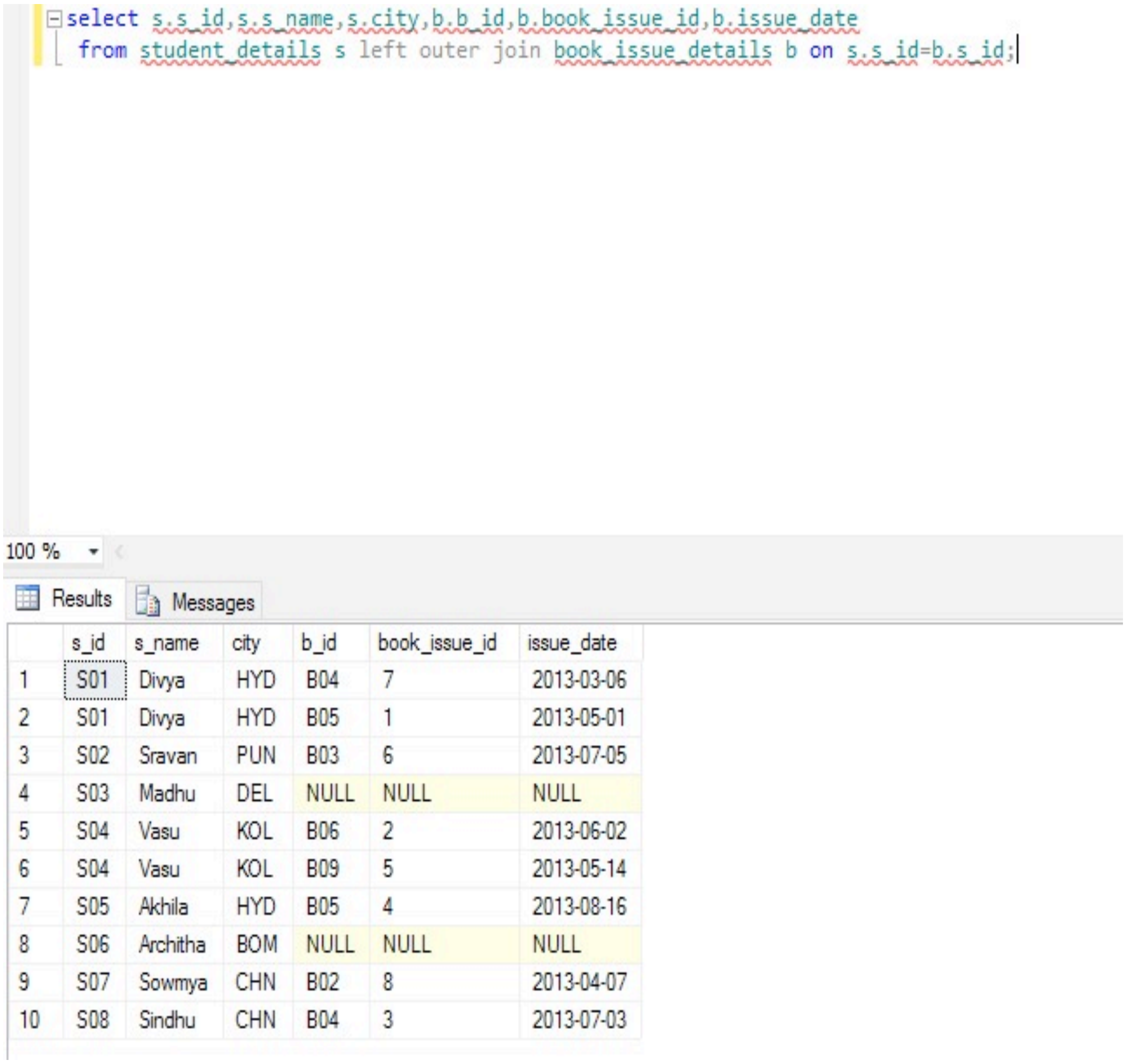
The last query example uses the Left Join operation to who have issued the books from the library. It also provides with the details of when the user has issued the book.

This query returns the records common in both the relations and also retains all the remaining records in the left table with the corresponding record values set to NULL

SELECT s.s_id, s.s_name, s.city, b.b_id, b.book_issue_id, b.issue_date

FROM student_details s

LEFT JOIN book_issue details ON s.s_id = b.s_id;



```
select s.s_id,s.s_name,s.city,b.b_id,b.book_issue_id,b.issue_date
 from student_details s left outer join book_issue_details b on s.s_id=b.s_id;
```

	s_id	s_name	city	b_id	book_issue_id	issue_date
1	S01	Divya	HYD	B04	7	2013-03-06
2	S01	Divya	HYD	B05	1	2013-05-01
3	S02	Sravan	PUN	B03	6	2013-07-05
4	S03	Madhu	DEL	NULL	NULL	NULL
5	S04	Vasu	KOL	B06	2	2013-06-02
6	S04	Vasu	KOL	B09	5	2013-05-14
7	S05	Akhila	HYD	B05	4	2013-08-16
8	S06	Architha	BOM	NULL	NULL	NULL
9	S07	Sowmya	CHN	B02	8	2013-04-07
10	S08	Sindhu	CHN	B04	3	2013-07-03

Update Examples: Update 1

Users are allowed to update or edit theoir details corresponding to some particular details. This is very useful for the user if they have inserted wrong details in the tables the very first time or else if they get any updated information and so they want to change it. update is the keyword used to perform this operation.

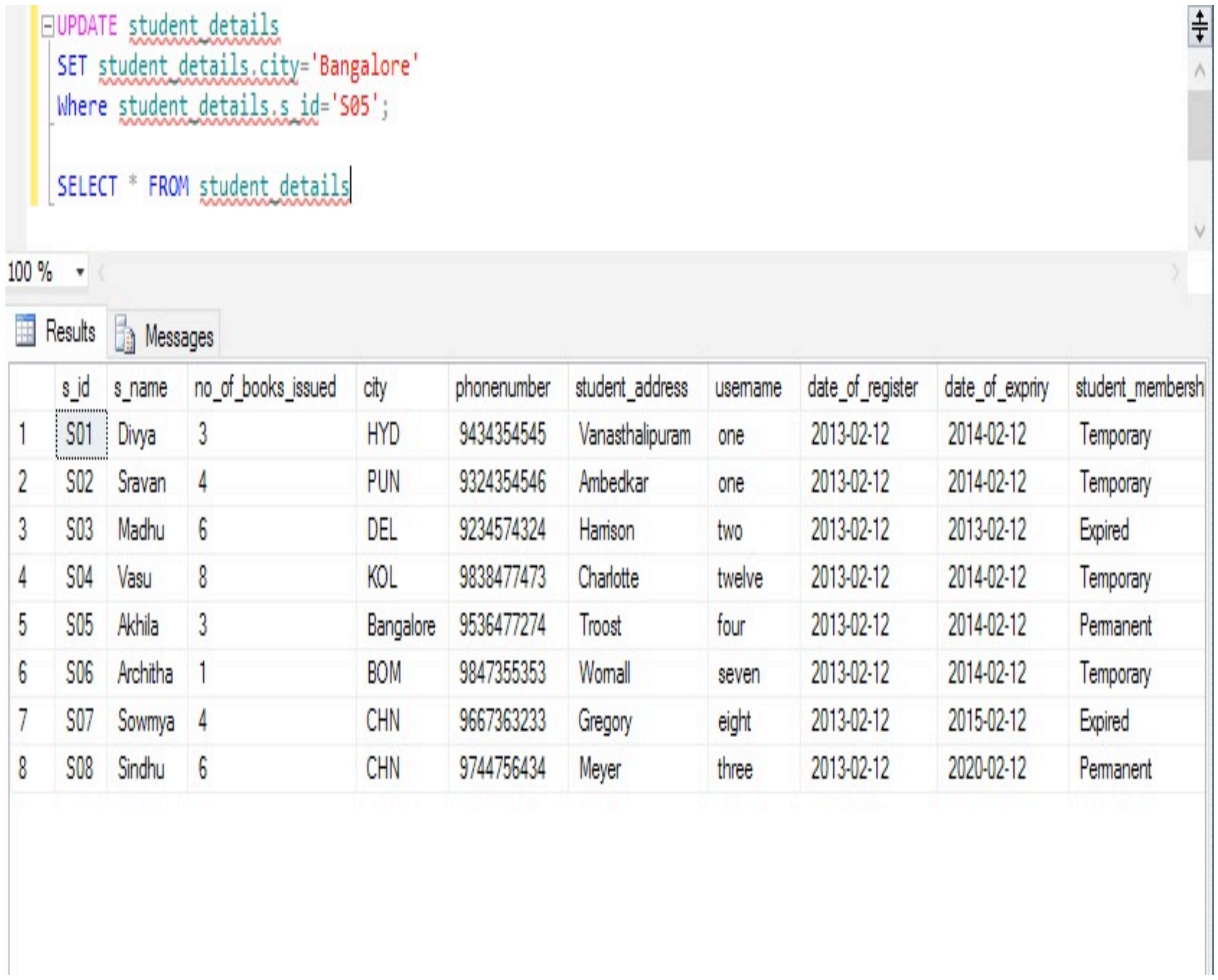
In this example, we wish to update the city of a student with some particular id.

UPDATE student_details

SET student_details.city='Bangalore'

Where student_details.s_id='S05';

This is used because the student was initially at one place and now his address was updated. so to change that this query is very useful.



```
UPDATE student_details
SET student_details.city='Bangalore'
Where student_details.s_id='S05';

SELECT * FROM student_details
```

	s_id	s_name	no_of_books_issued	city	phonenumber	student_address	usemame	date_of_register	date_of_expriy	student_membersh
1	S01	Divya	3	HYD	9434354545	Vanasthalipuram	one	2013-02-12	2014-02-12	Temporary
2	S02	Sravan	4	PUN	9324354546	Ambedkar	one	2013-02-12	2014-02-12	Temporary
3	S03	Madhu	6	DEL	9234574324	Harrison	two	2013-02-12	2013-02-12	Expired
4	S04	Vasu	8	KOL	9838477473	Charlotte	twelve	2013-02-12	2014-02-12	Temporary
5	S05	Akhila	3	Bangalore	9536477274	Troost	four	2013-02-12	2014-02-12	Permanent
6	S06	Architha	1	BOM	9847355353	Wornall	seven	2013-02-12	2014-02-12	Temporary
7	S07	Sowmya	4	CHN	9667363233	Gregory	eight	2013-02-12	2015-02-12	Expired
8	S08	Sindhu	6	CHN	9744756434	Meyer	three	2013-02-12	2020-02-12	Permanent

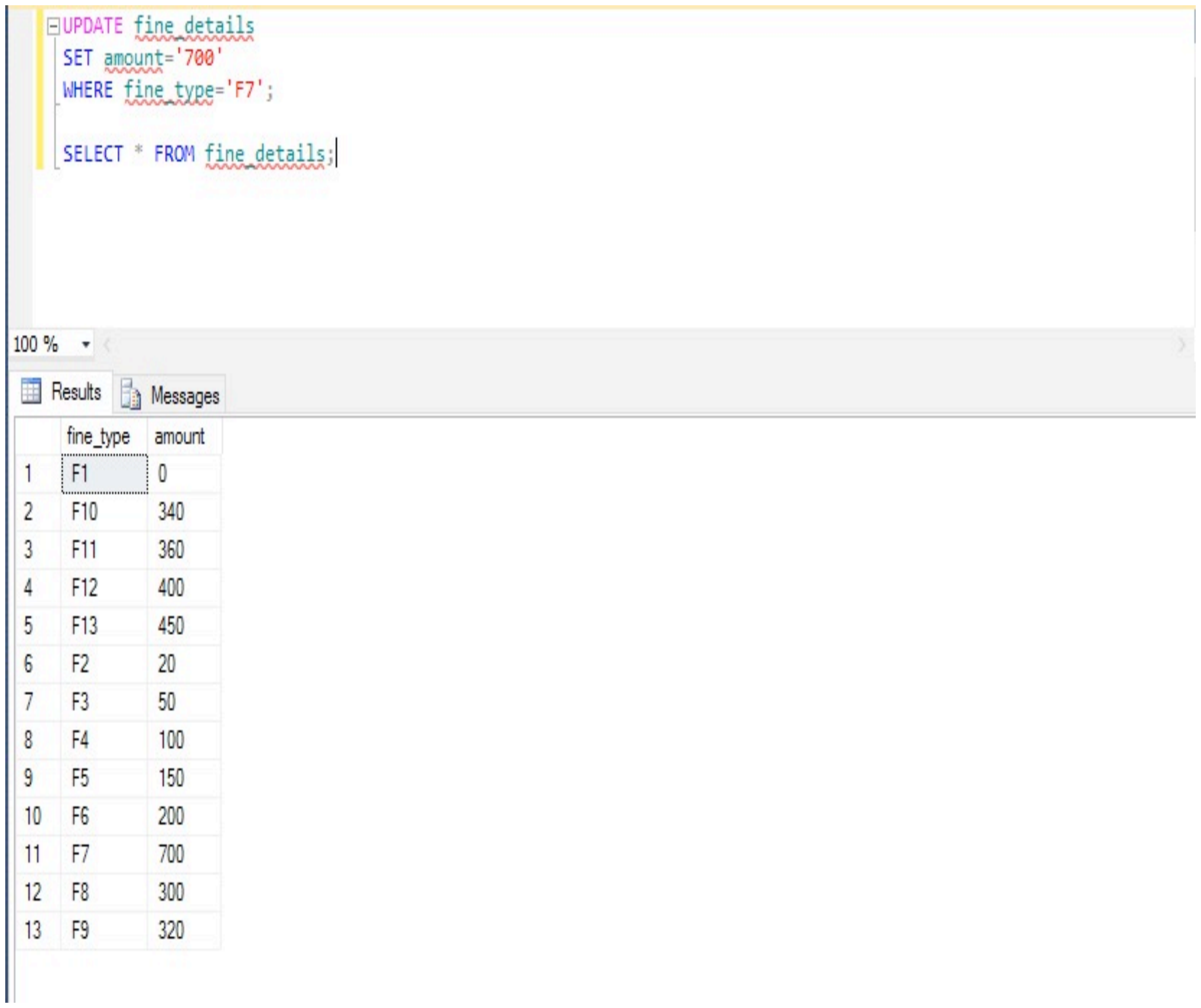
Update Examples: Update 2

There are many attributed which a user can update other such update is changing the fine amount for a particular range. If an amount for any range is very huge and students are facing issues to pay the large amount as they lost the book this is very useful.

UPDATE fine_details

SET amount='700'

WHERE fine_type='F7';



```
UPDATE fine_details
SET amount='700'
WHERE fine_type='F7';

SELECT * FROM fine_details;
```

	fine_type	amount
1	F1	0
2	F10	340
3	F11	360
4	F12	400
5	F13	450
6	F2	20
7	F3	50
8	F4	100
9	F5	150
10	F6	200
11	F7	700
12	F8	300
13	F9	320

Glossary

Library : This provides the ability to access the book, also provides a functionality to issue or return the books.

Temporary: Status denotes that the student's membership in the library is not permanent.

Book Details: Contains all the information related to all the books present in the library. This also provides minute details like rack number in which the corresponding books are placed.

Fine type: Based upon the days the book needs to be returned and if a book is not returned by a student there are various types of fines defined in this library. Depending upon the number of days Fine types are given and the amount will have to be paid by the student.

Date of Register: Returns the information related to the student's enrollment date in the library. From then only a user will have access to the books and facilities in the library.

Version information: Gives the details about when the book is published and what is its current version in the market for that particular book. Looking up for this information a student can wish to take the version in which he needs.

Duration of work: Every librarian has a shift in which he can work in the library. This duration of work defines the number of hours a librarian can work. By achieving this data a student can have an estimate about when the library can be accessed.

Login information: Contains all entire list of students and librarians in a particular university or an educational system. Count function can be applied to the attribute username and find the total number of people. This also has a useful credentials required for a student or a librarian to access the library based upon the type of the user. The secure information is to be carefully handled.

Cost of Book: The amount at which a student can entirely purchase his own book from any store is the cost provided in the database.

Row number: Information about the books and the racks where the books are placed in a library. This helps the user to find the required book easily without any difficulty.

References :

http://searchcrm.techtarget.com/definition/entity-relationship-diagram

http://creately.com/ER-diagram-software

http://creately.com/blog/diagrams/er-diagrams-tutorial/

http://www.studytonight.com/dbms/database-normalization.php