University's library

Project Proposal

Brief Description :

Our project talks about the university library, as the university library includes a large number of different and diverse books in different languages and ideas, it makes it easier for students to search and learn better and faster, as the main objective of our project is to facilitate the process of borrowing books from the library.

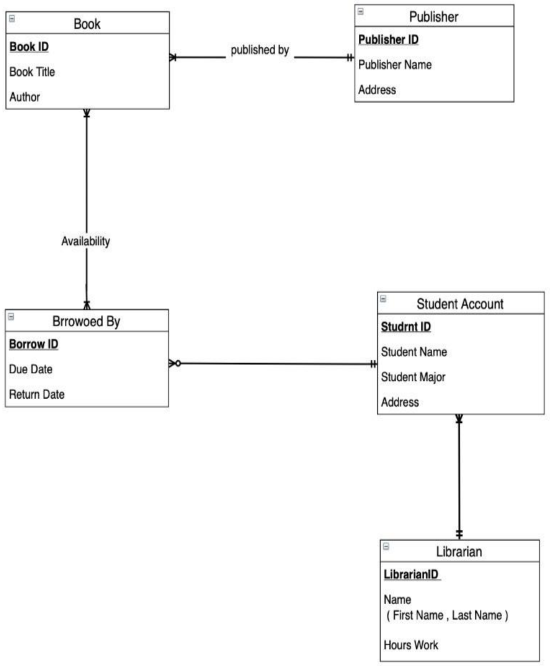
Problems:

Many students face a problem in buying books due to their high prices, and students may need some books, but they are sometimes not available in libraries, and libraries are outside working hours which may cause a problem for many students because they need to borrow or buy books at outside working hours.

Solutions:

And due to the many problems that students may face, the university library has provided a wonderful feature, which is the digital library, where the digital library helps students to book books via the Internet without the need to wait for the library's working hours, as the digital library provides the books that the student needs, and when he By booking books through the digital university library, he can go and receive the book without having to wait for a long time, as the goal of the digital library is to save effort, money and time for students.

ER Diagram :



Entities :

- Publisher**:** Represent the person that publishes the book, the Primary Key is **Publisher_ID.**

- Book**:** Represent the type the Student will borrow, the Primary Key is **Book_ID.**

- Borrow**:** Represent the type of the book and the date of Due Date and Return Date, the Primary key is **Borrow_ID.**

- Student Account: Represent the information about the students who will borrow the books the Primary Key is **Student_ID**.

- Librarian: Represent the person who gives the students the books that they will borrow, the primary Key is **Librarian_ID.**

Attribute:

Librarian

- Librarin_ID PRIMARY KEY
- Name
- Composite
- (First_Name,Last_Name)
- Hours Work

StudentAccount

- Student_ID PRIMARY KEY
- Student_Name
- Student Major
- Address

Borrowed

- BrrowID PRIMARY KEY
- DueDate
- Return

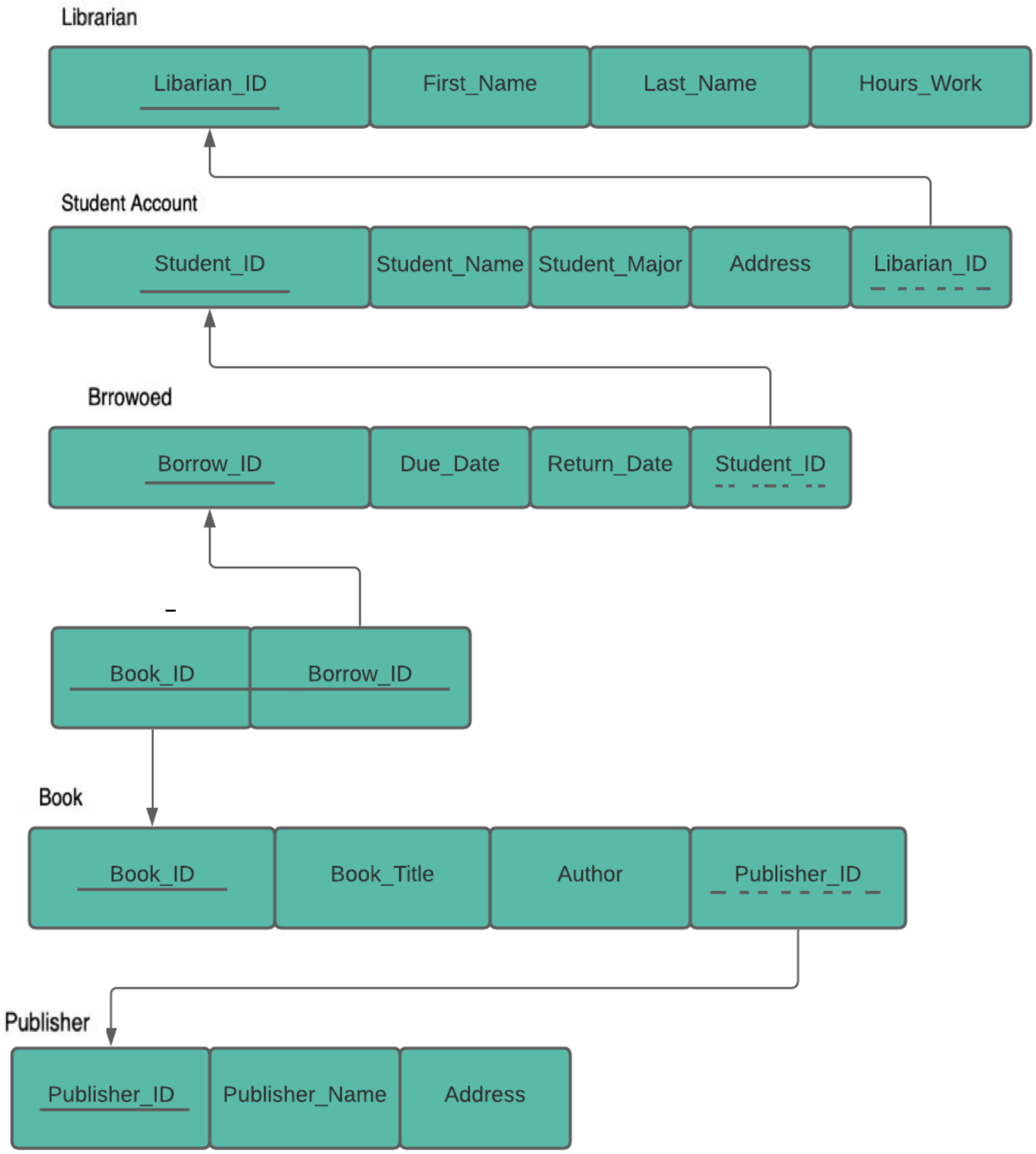
Book

- Book_ID PRIMARY KEY
- Book_Title
- Author

Publisher

- Publisher_ID PRIMARY KEY
- Publisher_Name
- Address

Convert ER Diagram to a relational schema :



Functional Dependencies :

Librarin_ID < First_Name, Last_Name, Hours_work

Student_ID ‹ Student_Name , Student_Major , Address, Librarin_ID

Borrow_ID < Due_Date, Rreturn_Date, Student_ID

Book_ID ‹ Book_Title, Author, Publisher_ID Publisher_ID <
PublisherName, Address

Normalization:

1NF

All attribute value is atomic and there no repeating groups So,

all of them is in the 1NF.

2NF

The relation is in 1NF, all non-key attribute is fully functionally dependent on entire primary key So, all of them in 2NF.

3NF

The relation is in the 2NF, all transitive dependencies have been removed so all of them in 3NF

Librarian(Librarian_ID,First_Name,Last_Name,Hours_Work)

StudentAccount(Student_ID,StudentName,StudentMajor,Address,LibrarianID*)

Brrowed(Brrow_ID,DueDate,Return,_Student_ID*)

BookBrrow(Book_ID*, Brrow_ID *)

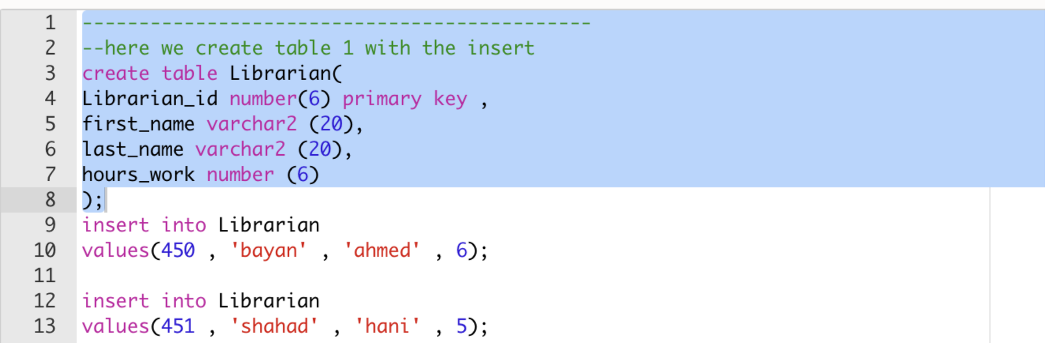
Book(Book_ID,Book_Title,Author,publisher_ID*)

Publisher(Publisher_ID,publisher_Name,Address)

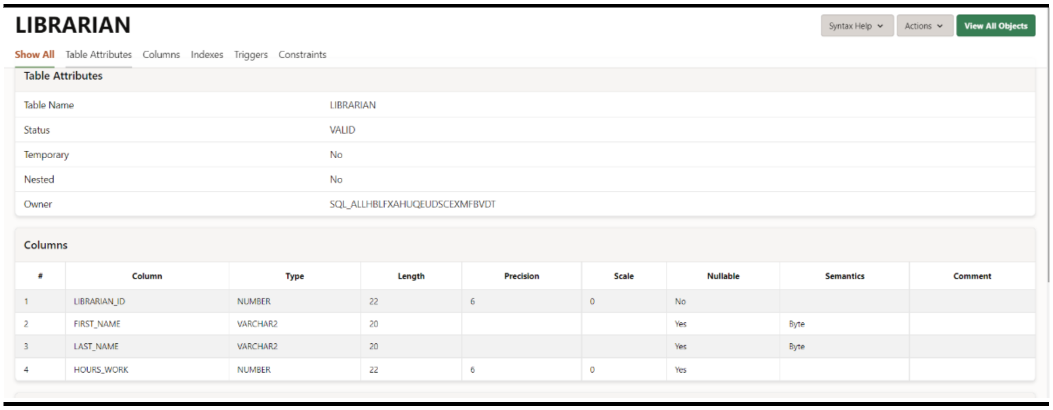
Phase2: Physical Database Implementation

Part1: Create the normalized tables :
Part2: Populate your tables with 5 rows at least :

Table 1 Librarian :



```
---------------------------------------------
--here we create table 1 with the insert
create table Librarian(
Librarian_id number(6) primary key ,
first_name varchar2 (20),
last_name varchar2 (20),
hours_work number (6)
);
insert into Librarian
values(450 , 'bayan' , 'ahmed' , 6);

insert into Librarian
values(451 , 'shahad' , 'hani' , 5);
```



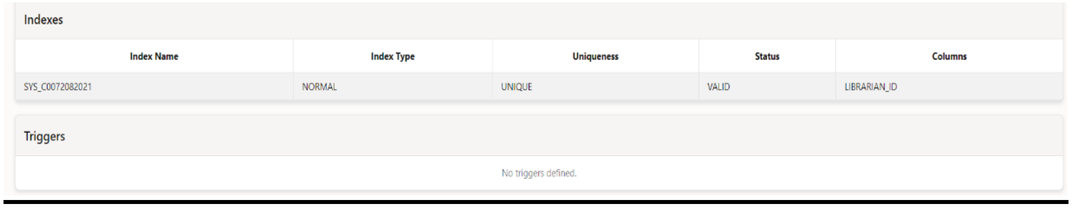
LIBRARIAN

Table Attributes

Table Name	LIBRARIAN
Status	VALID
Temporary	No
Nested	No
Owner	SQL_ALLHBLFXAHUQEUDSCEXMFBVDT

Columns

#	Column	Type	Length	Precision	Scale	Nullable	Semantics	Comment
1	LIBRARIAN_ID	NUMBER	22	6	0	No		
2	FIRST_NAME	VARCHAR2	20			Yes	Byte	
3	LAST_NAME	VARCHAR2	20			Yes	Byte	
4	HOURS_WORK	NUMBER	22	6	0	Yes		



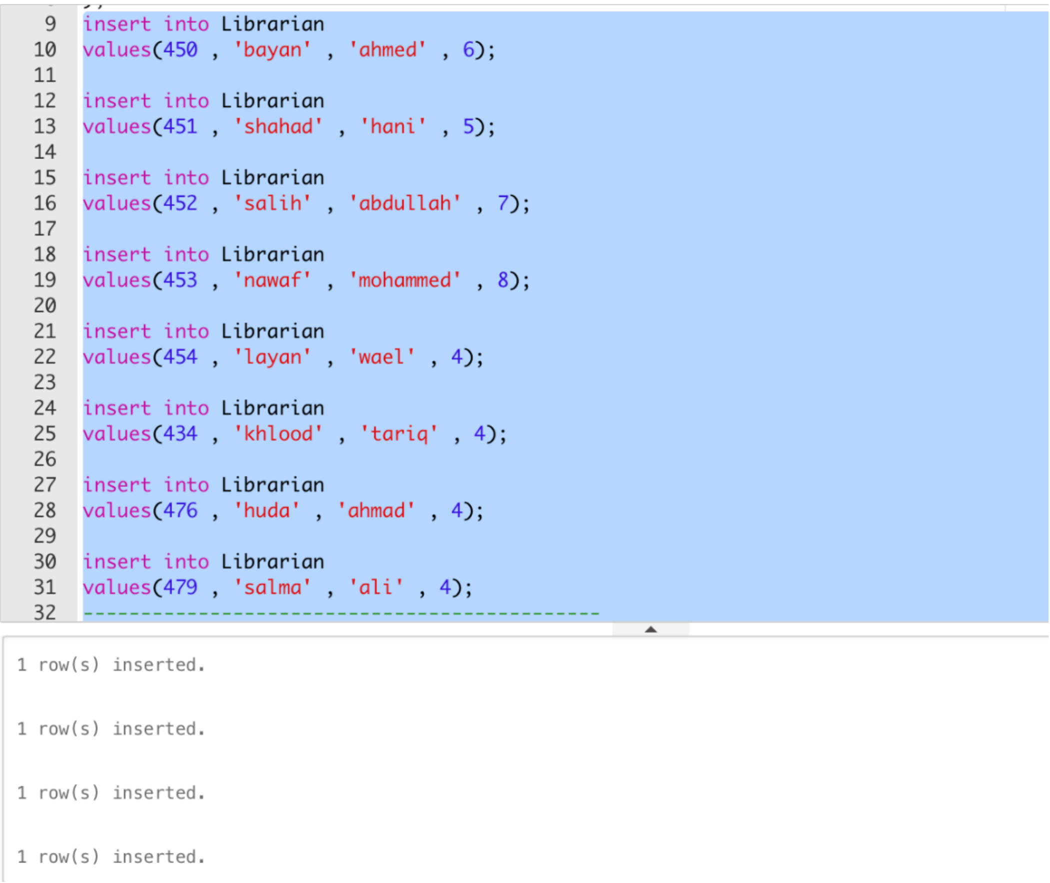
Indexes

Index Name	Index Type	Uniqueness	Status	Columns
SYS_C0072082021	NORMAL	UNIQUE	VALID	LIBRARIAN_ID

Triggers

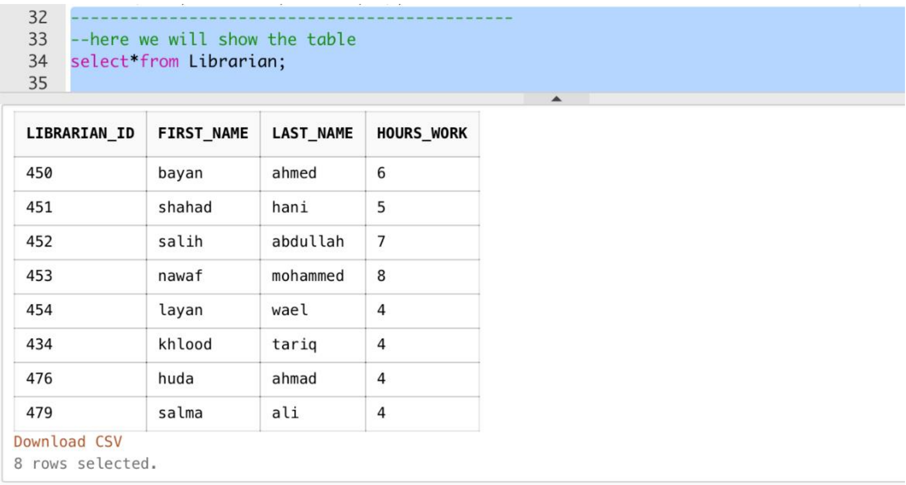
No triggers defined.

The insert of the table 1 :



```
insert into Librarian
values(450 , 'bayan' , 'ahmed' , 6);

insert into Librarian
values(451 , 'shahad' , 'hani' , 5);

insert into Librarian
values(452 , 'salih' , 'abdullah' , 7);

insert into Librarian
values(453 , 'nawaf' , 'mohammed' , 8);

insert into Librarian
values(454 , 'layan' , 'wael' , 4);

insert into Librarian
values(434 , 'khlood' , 'tariq' , 4);

insert into Librarian
values(476 , 'huda' , 'ahmad' , 4);

insert into Librarian
values(479 , 'salma' , 'ali' , 4);
---------------------------------------------
```

```
1 row(s) inserted.


1 row(s) inserted.


1 row(s) inserted.


1 row(s) inserted.
```



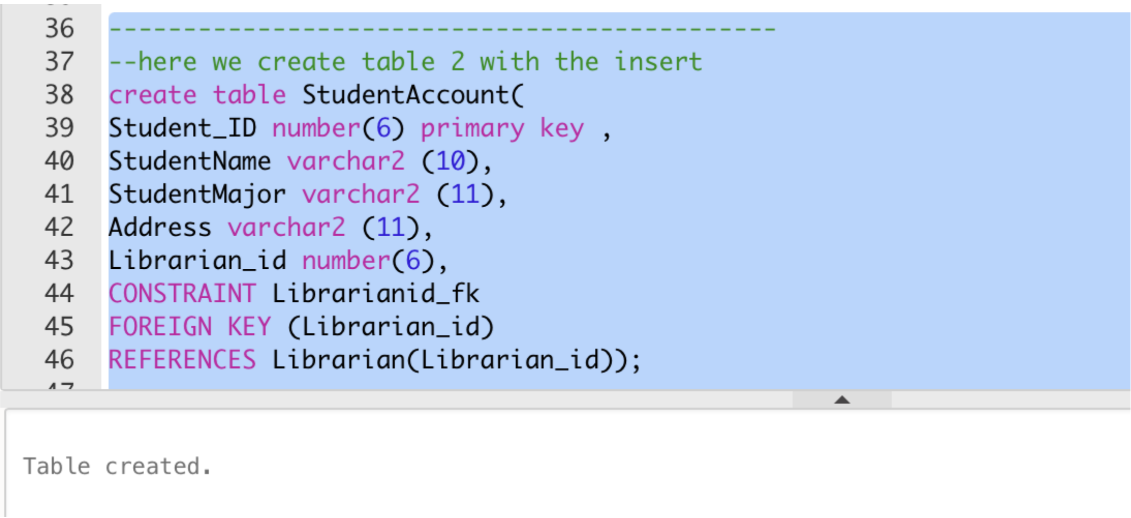
```
---------------------------------------------
--here we will show the table
select*from Librarian;
```

LIBRARIAN_ID	FIRST_NAME	LAST_NAME	HOURS_WORK
450	bayan	ahmed	6
451	shahad	hani	5
452	salih	abdullah	7
453	nawaf	mohammed	8
454	layan	wael	4
434	khlood	tariq	4
476	huda	ahmad	4
479	salma	ali	4

Download CSV
8 rows selected.

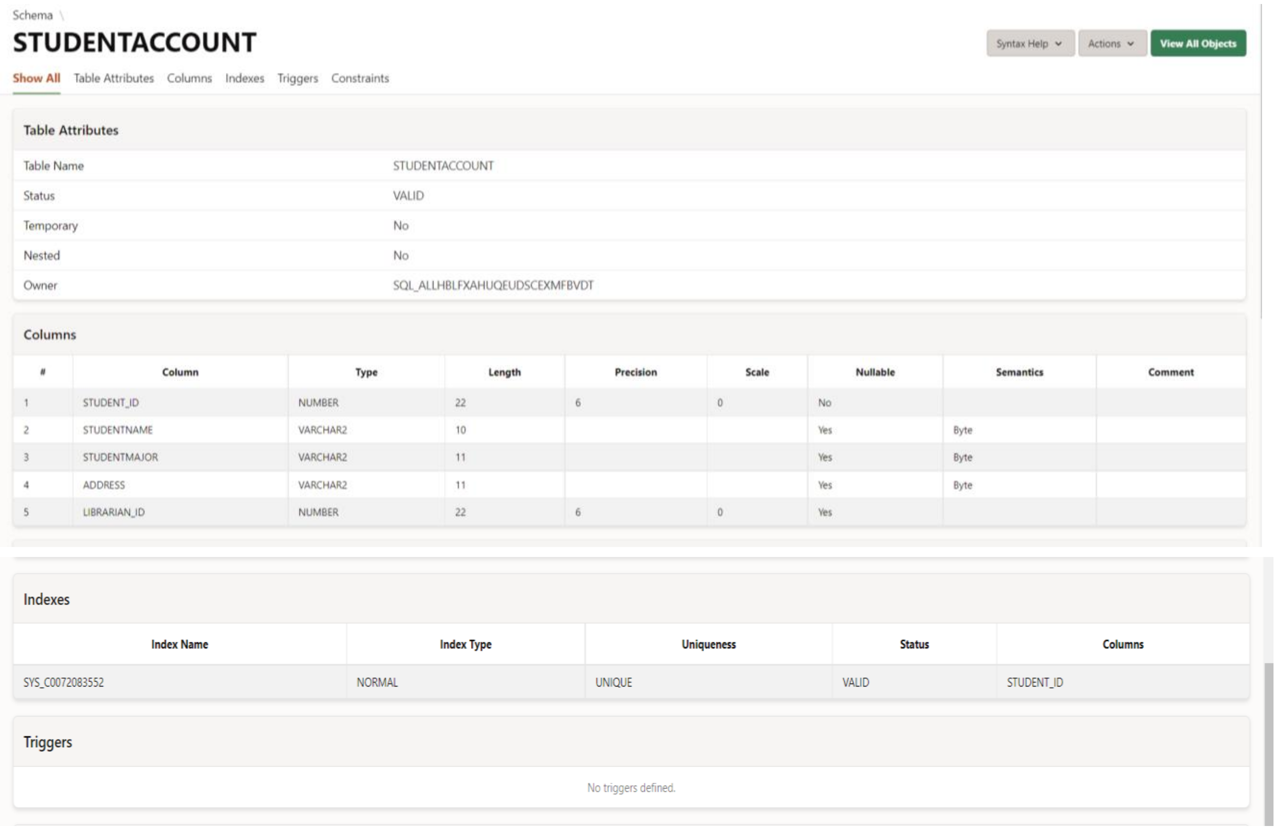
We create a table for the Librarian.And we insert all the values in this table, where, also Librarian_id is the primary key, Then we write the **SELECT*FROM Librarian**; to show the table

Table 2 StudentAccount :



```
---------------------------------------------
--here we create table 2 with the insert
create table StudentAccount(
Student_ID number(6) primary key ,
StudentName varchar2 (10),
StudentMajor varchar2 (11),
Address varchar2 (11),
Librarian_id number(6),
CONSTRAINT Librarianid_fk
FOREIGN KEY (Librarian_id)
REFERENCES Librarian(Librarian_id));
```

```
Table created.
```



STUDENTACCOUNT

Table Attributes

Table Name	STUDENTACCOUNT
Status	VALID
Temporary	No
Nested	No
Owner	SQL_ALLHBLFXAHUQEUDSCEXMFBVDT

Columns

#	Column	Type	Length	Precision	Scale	Nullable	Semantics	Comment
1	STUDENT_ID	NUMBER	22	6	0	No		
2	STUDENTNAME	VARCHAR2	10			Yes	Byte	
3	STUDENTMAJOR	VARCHAR2	11			Yes	Byte	
4	ADDRESS	VARCHAR2	11			Yes	Byte	
5	LIBRARIAN_ID	NUMBER	22	6	0	Yes		

Indexes

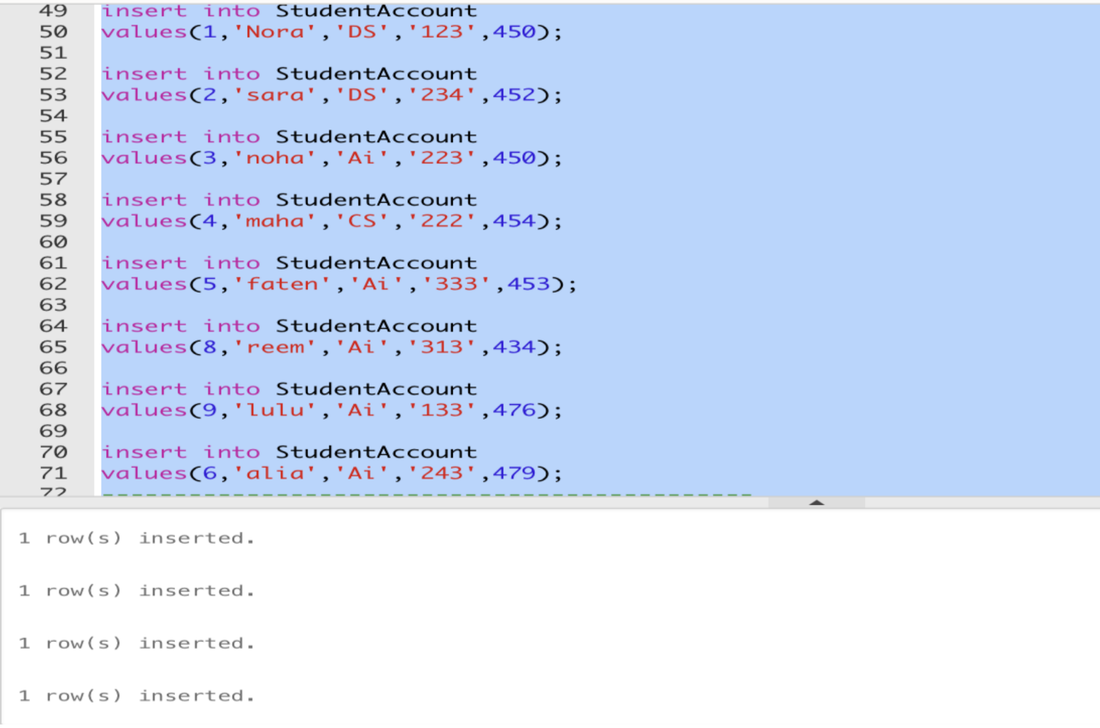
Index Name	Index Type	Uniqueness	Status	Columns
SYS_C0072083552	NORMAL	UNIQUE	VALID	STUDENT_ID

Triggers

No triggers defined.

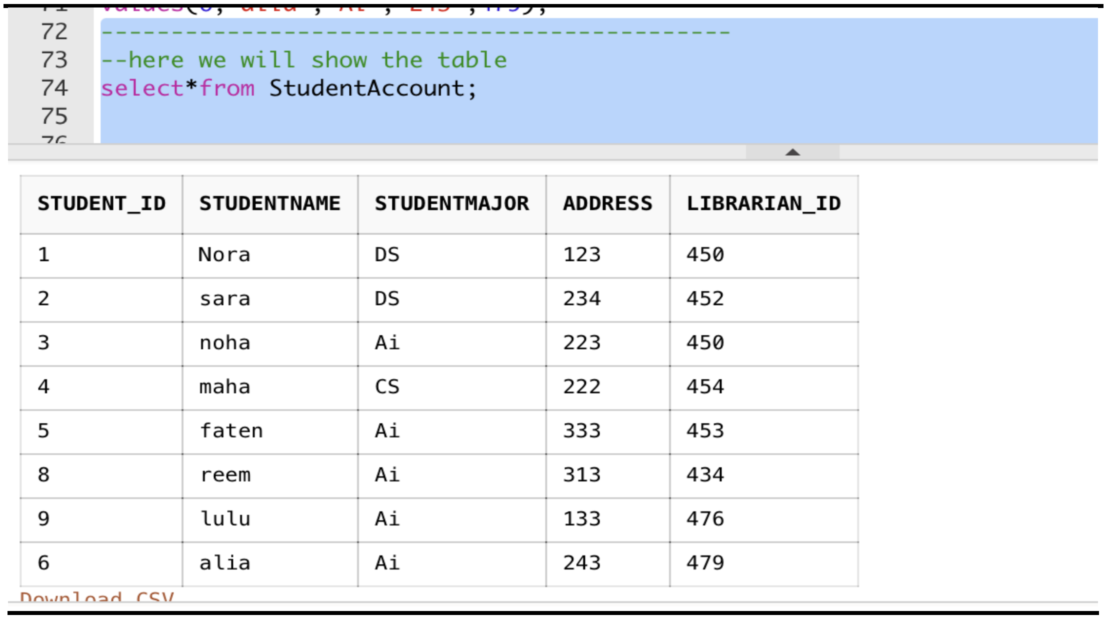
We create table 2 **StudentAccount**, and the **student_id** is primary key.

The insert of the table 2 :



```
insert into StudentAccount
values(1,'Nora','DS','123',450);

insert into StudentAccount
values(2,'sara','DS','234',452);

insert into StudentAccount
values(3,'noha','Ai','223',450);

insert into StudentAccount
values(4,'maha','CS','222',454);

insert into StudentAccount
values(5,'faten','Ai','333',453);

insert into StudentAccount
values(8,'reem','Ai','313',434);

insert into StudentAccount
values(9,'lulu','Ai','133',476);

insert into StudentAccount
values(6,'alia','Ai','243',479);
----------------------------------------------
```

```
1 row(s) inserted.

1 row(s) inserted.

1 row(s) inserted.

1 row(s) inserted.
```



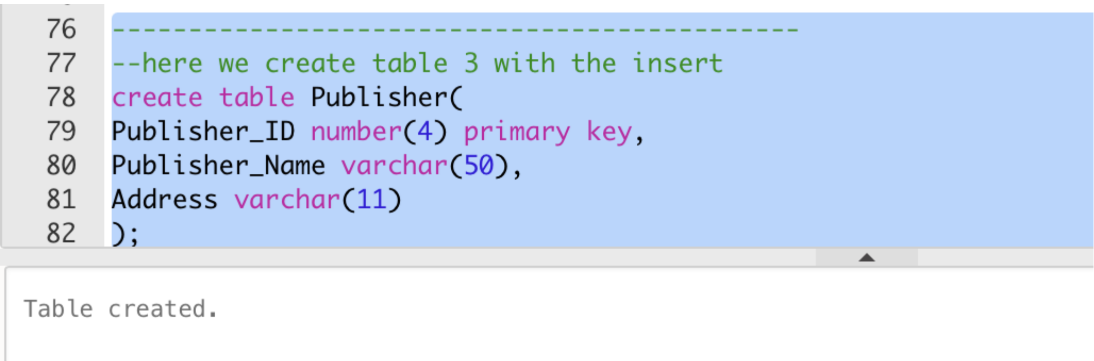
```
----------------------------------------------
--here we will show the table
select*from StudentAccount;
```

STUDENT_ID	STUDENTNAME	STUDENTMAJOR	ADDRESS	LIBRARIAN_ID
1	Nora	DS	123	450
2	sara	DS	234	452
3	noha	Ai	223	450
4	maha	CS	222	454
5	faten	Ai	333	453
8	reem	Ai	313	434
9	lulu	Ai	133	476
6	alia	Ai	243	479

Download CSV

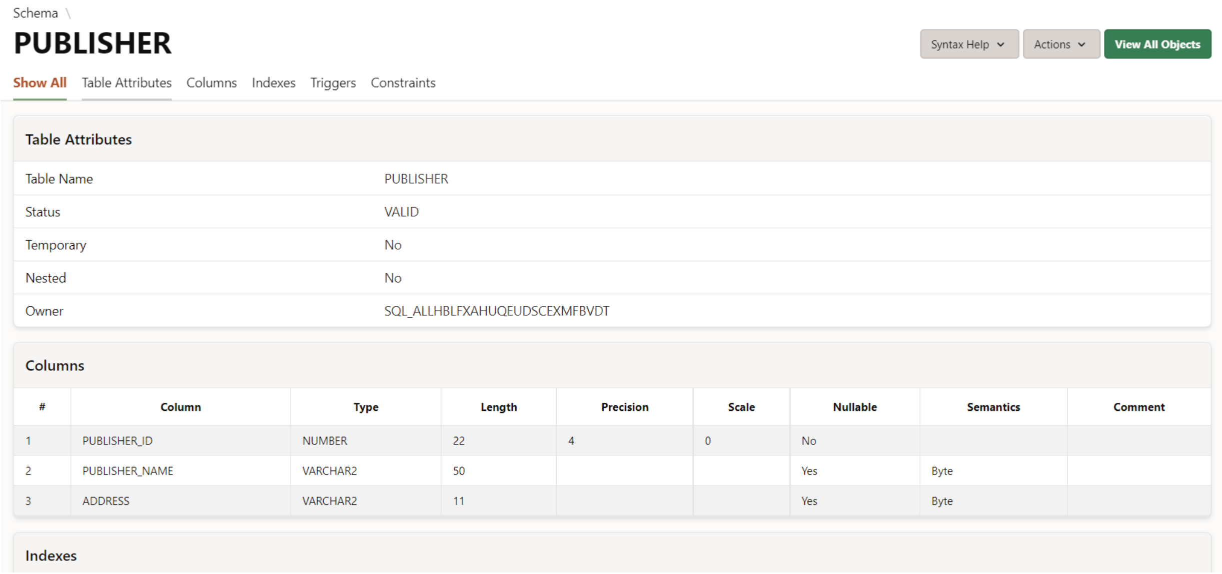
After we create table 2 we insert the value to the table , then we write the **SELECT*FROM StudentAccount** ;
to show the table .

Table 3 Publisher:



```
---------------------------------------------
--here we create table 3 with the insert
create table Publisher(
Publisher_ID number(4) primary key,
Publisher_Name varchar(50),
Address varchar(11)
);
```

```
Table created.
```



Schema \

PUBLISHER

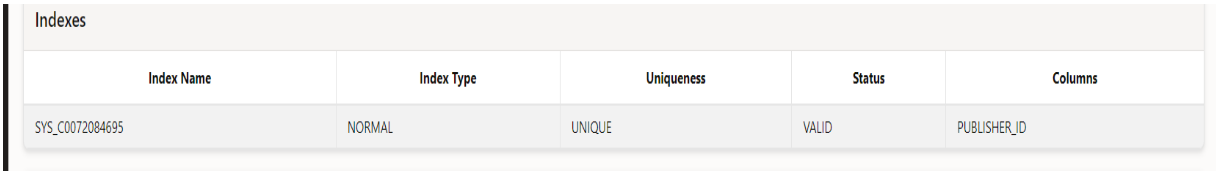
Table Attributes

Table Name	PUBLISHER
Status	VALID
Temporary	No
Nested	No
Owner	SQL_ALLHBLFXAHUQEUDSCEXMFBVDT

Columns

#	Column	Type	Length	Precision	Scale	Nullable	Semantics	Comment
1	PUBLISHER_ID	NUMBER	22	4	0	No		
2	PUBLISHER_NAME	VARCHAR2	50			Yes	Byte	
3	ADDRESS	VARCHAR2	11			Yes	Byte	

Indexes

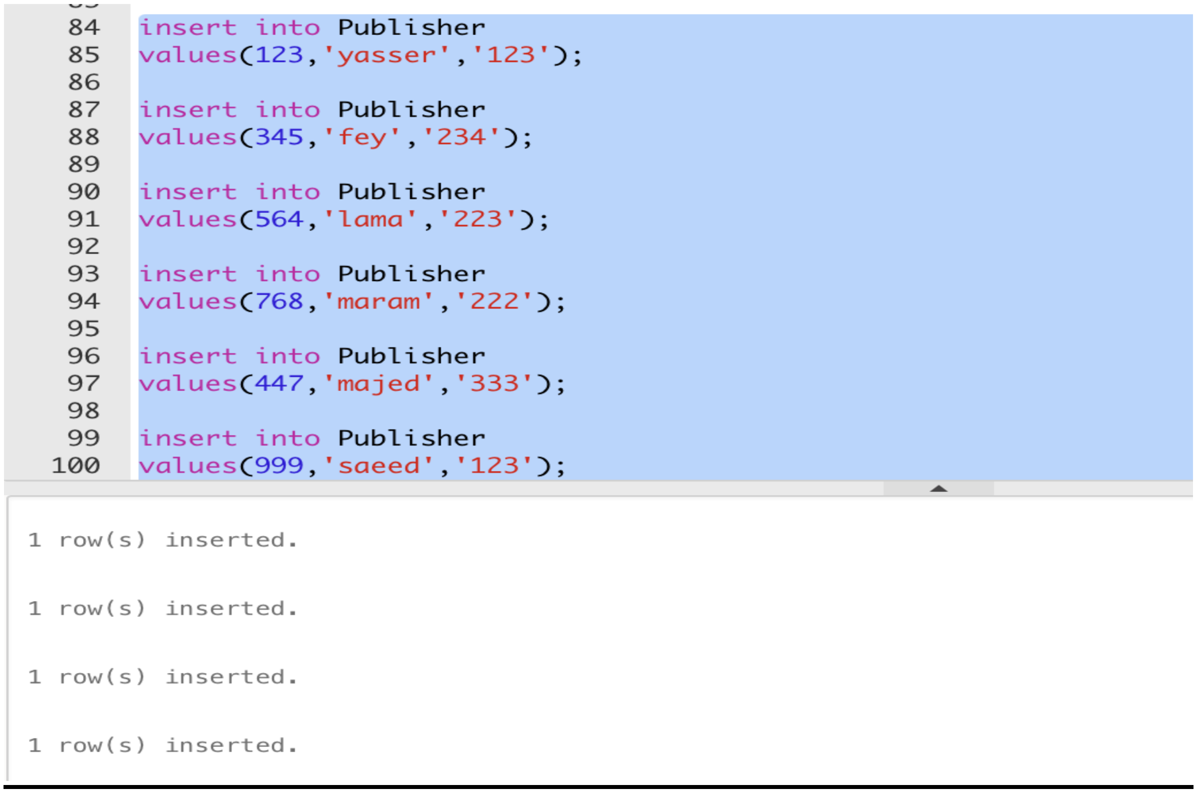


Indexes

Index Name	Index Type	Uniqueness	Status	Columns
SYS_C0072084695	NORMAL	UNIQUE	VALID	PUBLISHER_ID

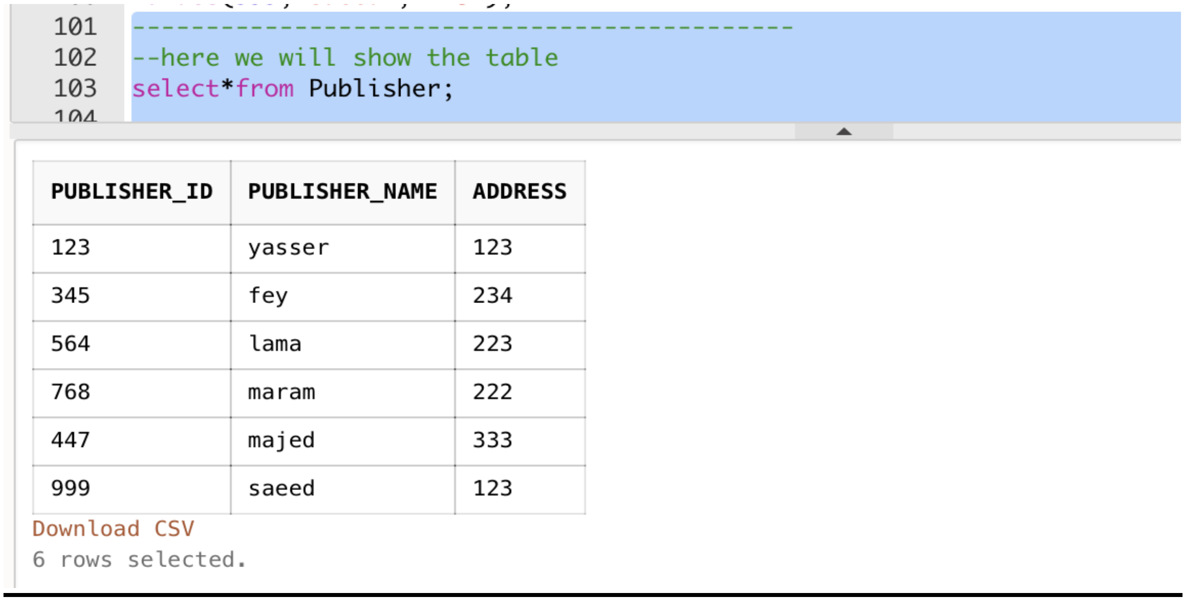
Here we create table 3 for publisher and we make the **publisher_ID** is **primary key .**

Insert to the Table 3 :



```
insert into Publisher
values(123,'yasser','123');

insert into Publisher
values(345,'fey','234');

insert into Publisher
values(564,'lama','223');

insert into Publisher
values(768,'maram','222');

insert into Publisher
values(447,'majed','333');

insert into Publisher
values(999,'saeed','123');
```

```
1 row(s) inserted.

1 row(s) inserted.

1 row(s) inserted.

1 row(s) inserted.
```



```
---------------------------------------------
--here we will show the table
select*from Publisher;
```

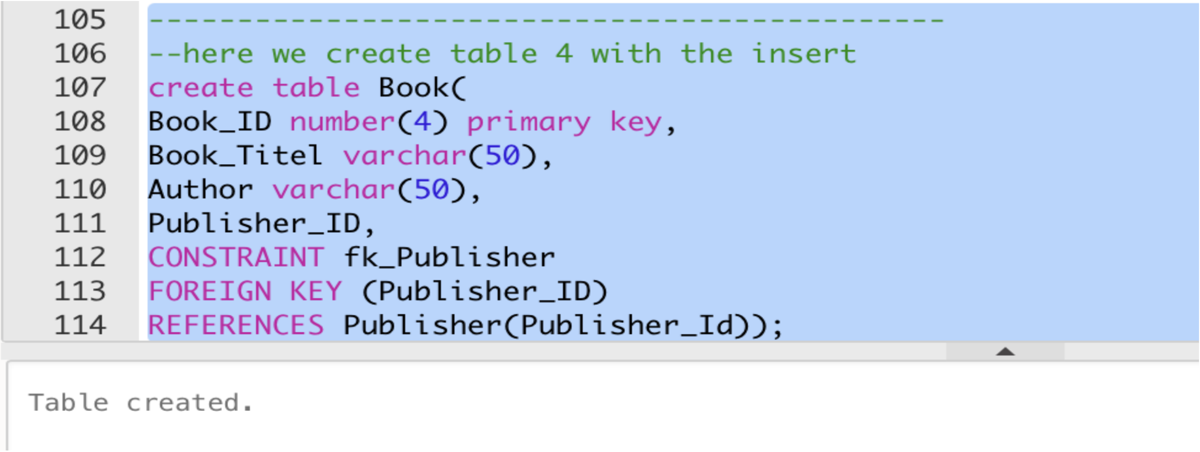
PUBLISHER_ID	PUBLISHER_NAME	ADDRESS
123	yasser	123
345	fey	234
564	lama	223
768	maram	222
447	majed	333
999	saeed	123

```
Download CSV
6 rows selected.
```

After we create the table 3 we insert all value in the table and we show the table with **SELECT* FROM Publisher ;**

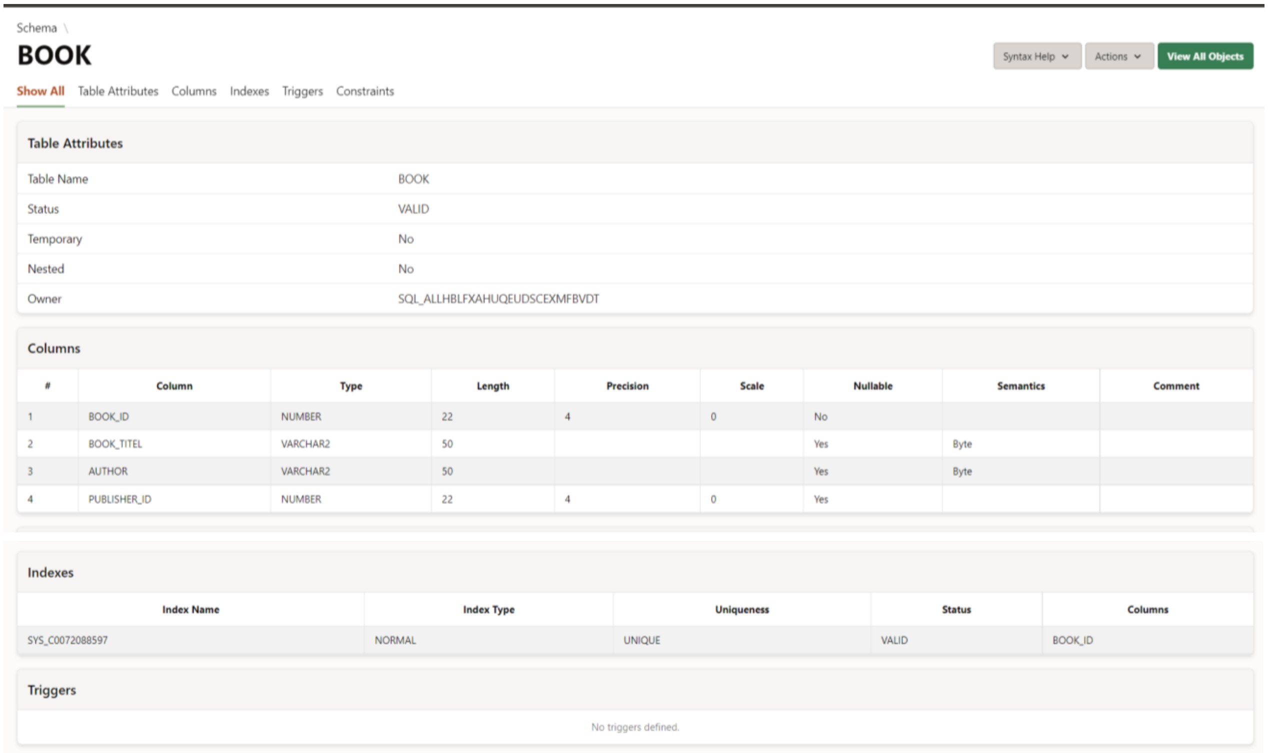
To show the table

Table 4 Book :



```
--------------------------------------------
--here we create table 4 with the insert
create table Book(
Book_ID number(4) primary key,
Book_Titel varchar(50),
Author varchar(50),
Publisher_ID,
CONSTRAINT fk_Publisher
FOREIGN KEY (Publisher_ID)
REFERENCES Publisher(Publisher_Id));
```

```
Table created.
```



BOOK

Table Attributes

Table Name	BOOK
Status	VALID
Temporary	No
Nested	No
Owner	SQL_ALLHBLFXAHUQEUDSCEXMFBVDT

Columns

#	Column	Type	Length	Precision	Scale	Nullable	Semantics	Comment
1	BOOK_ID	NUMBER	22	4	0	No		
2	BOOK_TITEL	VARCHAR2	50			Yes	Byte	
3	AUTHOR	VARCHAR2	50			Yes	Byte	
4	PUBLISHER_ID	NUMBER	22	4	0	Yes		

Indexes

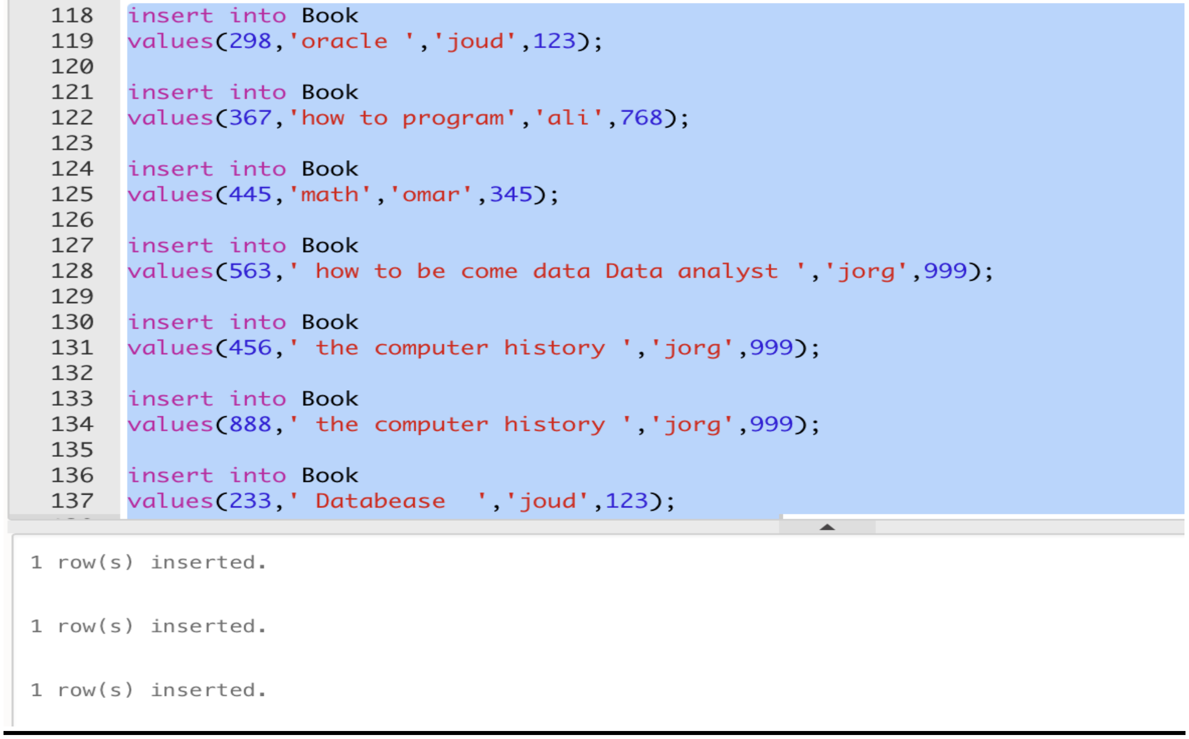
Index Name	Index Type	Uniqueness	Status	Columns
SYS_C0072088597	NORMAL	UNIQUE	VALID	BOOK_ID

Triggers

No triggers defined.

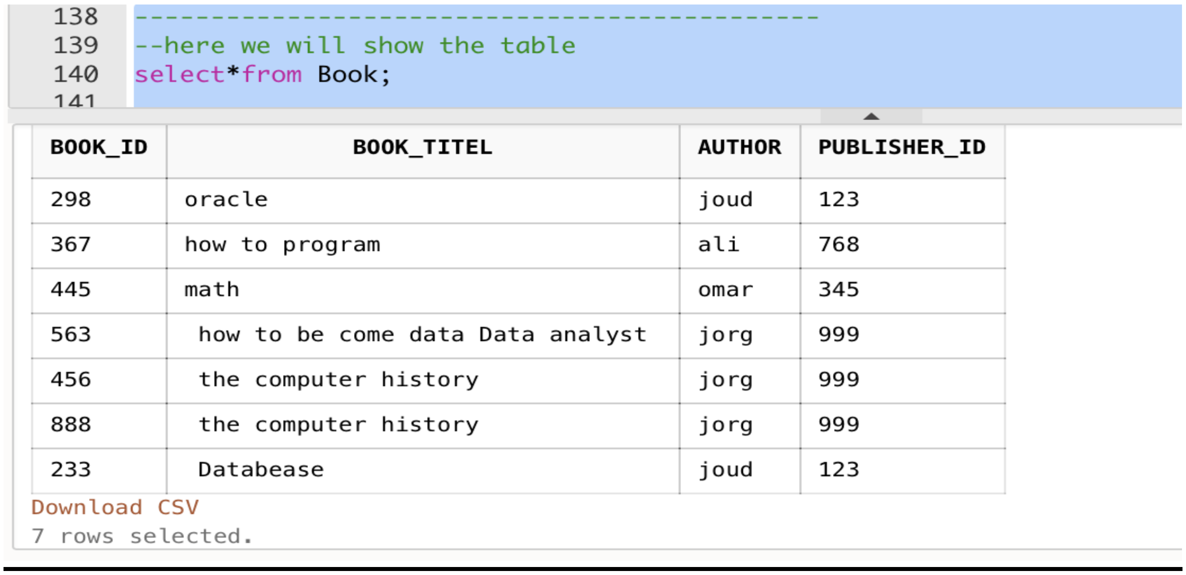
Here we create table 4 and it’s having primary key and forging key The primary key is **Book_ID** and the foring key is (**Publisher_ID**).

Insert to table 4 :



```
insert into Book
values(298,'oracle ','joud',123);

insert into Book
values(367,'how to program','ali',768);

insert into Book
values(445,'math','omar',345);

insert into Book
values(563,' how to be come data Data analyst ','jorg',999);

insert into Book
values(456,' the computer history ','jorg',999);

insert into Book
values(888,' the computer history ','jorg',999);

insert into Book
values(233,' Databease  ','joud',123);
```

```
1 row(s) inserted.

1 row(s) inserted.

1 row(s) inserted.
```



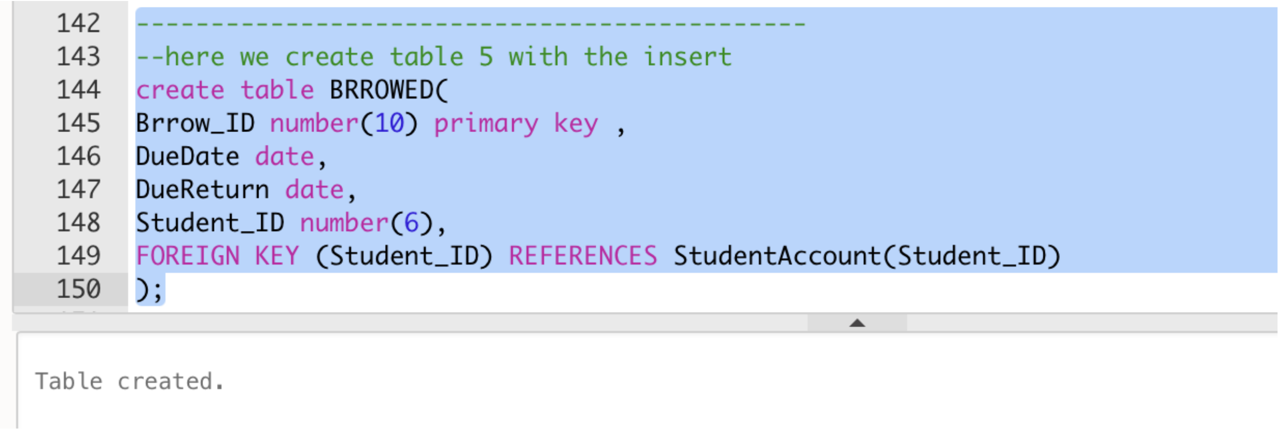
```
---------------------------------------------
--here we will show the table
select*from Book;
```

BOOK_ID	BOOK_TITEL	AUTHOR	PUBLISHER_ID
298	oracle	joud	123
367	how to program	ali	768
445	math	omar	345
563	how to be come data Data analyst	jorg	999
456	the computer history	jorg	999
888	the computer history	jorg	999
233	Databease	joud	123

```
Download CSV
7 rows selected.
```

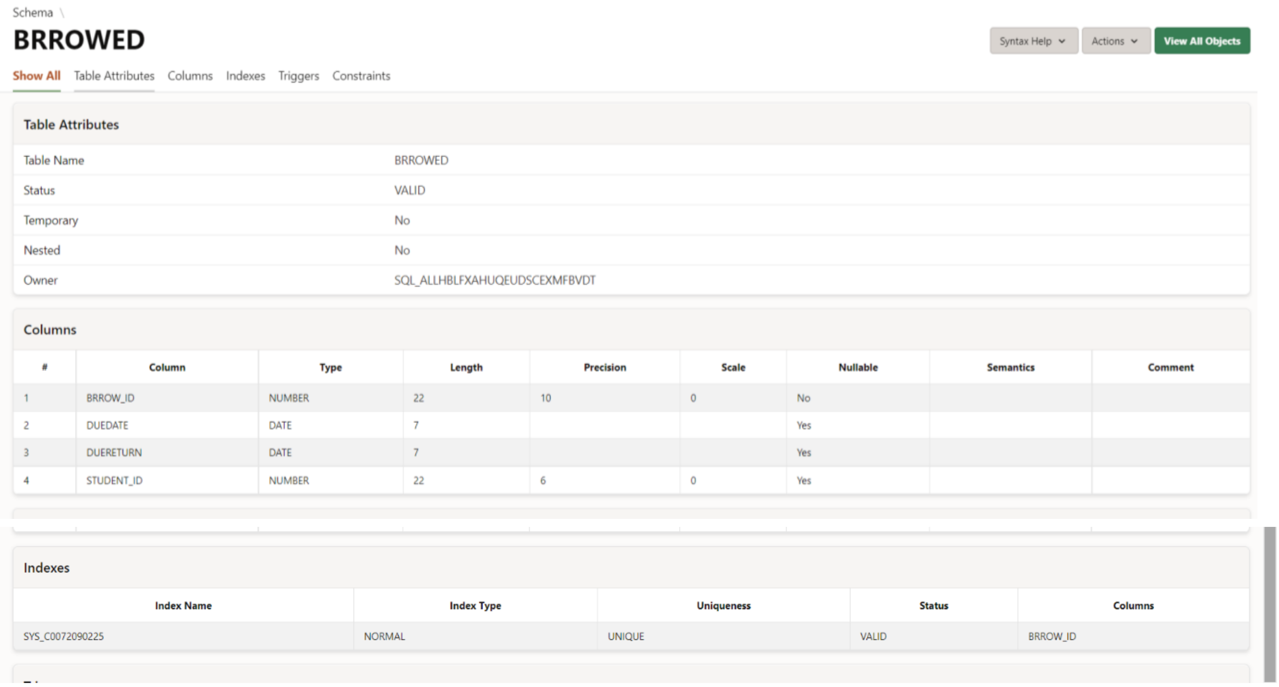
Here we insert all the value to the table then we show the table with **SELECT*FROM Book** to show the table.

<u>Table 5 BORROWED:</u>



```sql
-----------------------------------------------
--here we create table 5 with the insert
create table BRROWED(
Brrow_ID number(10) primary key ,
DueDate date,
DueReturn date,
Student_ID number(6),
FOREIGN KEY (Student_ID) REFERENCES StudentAccount(Student_ID)
);
```

```
Table created.
```



BRROWED

Table Attributes

Table Name	BRROWED
Status	VALID
Temporary	No
Nested	No
Owner	SQL_ALLHBLFXAHUQEUDSCEXMFBVDT

Columns

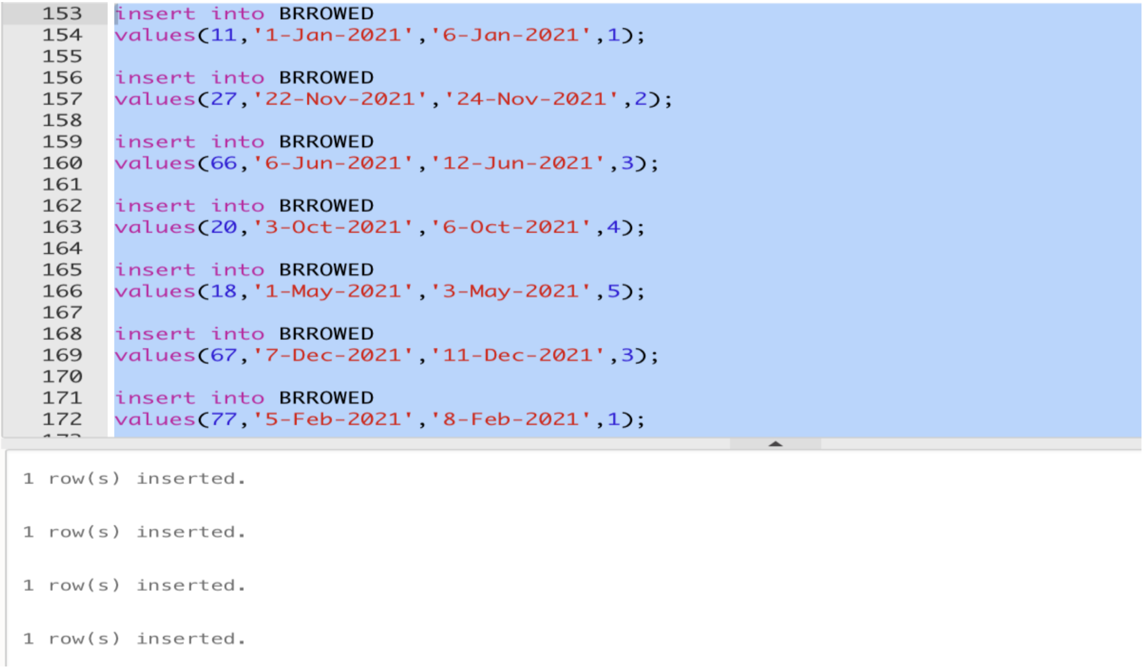
#	Column	Type	Length	Precision	Scale	Nullable	Semantics	Comment
1	BRROW_ID	NUMBER	22	10	0	No		
2	DUEDATE	DATE	7			Yes		
3	DUERETURN	DATE	7			Yes		
4	STUDENT_ID	NUMBER	22	6	0	Yes		

Indexes

Index Name	Index Type	Uniqueness	Status	Columns
SYS_C0072090225	NORMAL	UNIQUE	VALID	BRROW_ID

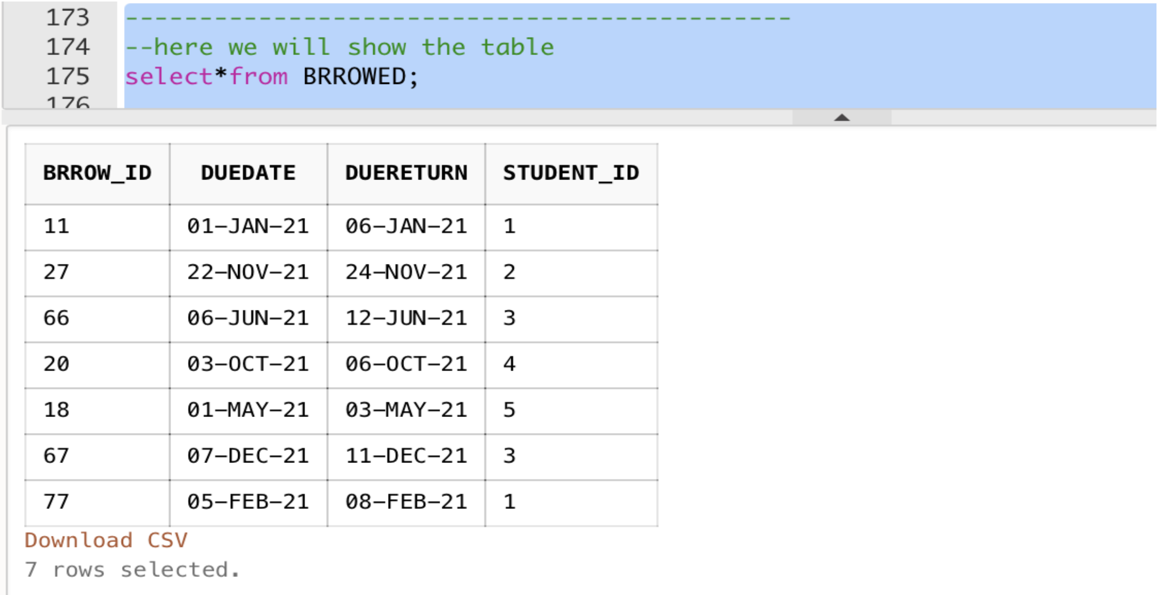
Here we create table 5 and it’s having primary key and forging key and the PK is **<u>BRROWED_id</u>** and forging key is **<u>(Student_ID).</u>**

Insert to table 5:



```
insert into BRROWED
values(11,'1-Jan-2021','6-Jan-2021',1);

insert into BRROWED
values(27,'22-Nov-2021','24-Nov-2021',2);

insert into BRROWED
values(66,'6-Jun-2021','12-Jun-2021',3);

insert into BRROWED
values(20,'3-Oct-2021','6-Oct-2021',4);

insert into BRROWED
values(18,'1-May-2021','3-May-2021',5);

insert into BRROWED
values(67,'7-Dec-2021','11-Dec-2021',3);

insert into BRROWED
values(77,'5-Feb-2021','8-Feb-2021',1);
```

```
1 row(s) inserted.

1 row(s) inserted.

1 row(s) inserted.

1 row(s) inserted.
```



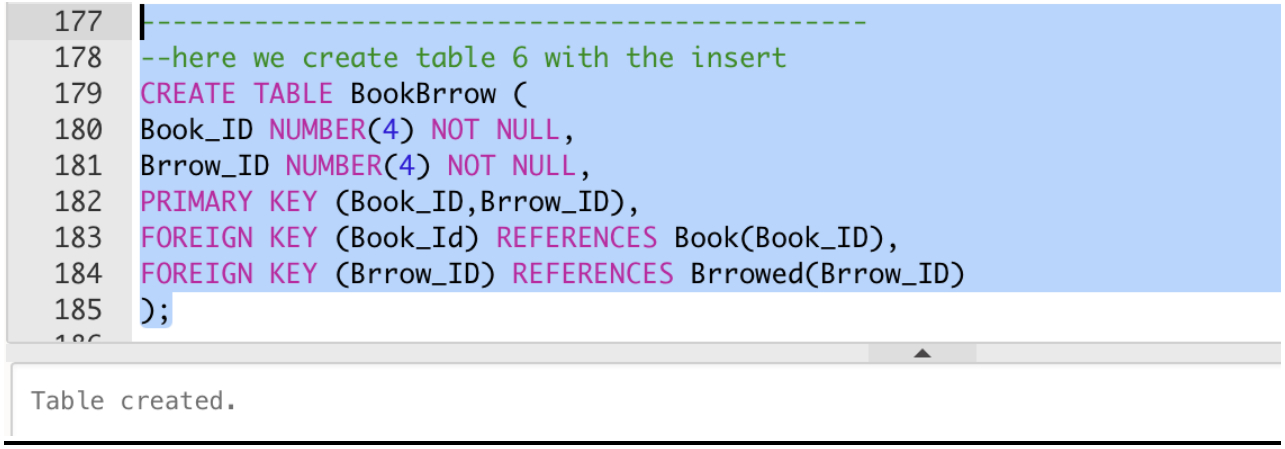
```
---------------------------------------------
--here we will show the table
select*from BRROWED;
```

BRROW_ID	DUEDATE	DUERETURN	STUDENT_ID
11	01-JAN-21	06-JAN-21	1
27	22-NOV-21	24-NOV-21	2
66	06-JUN-21	12-JUN-21	3
20	03-OCT-21	06-OCT-21	4
18	01-MAY-21	03-MAY-21	5
67	07-DEC-21	11-DEC-21	3
77	05-FEB-21	08-FEB-21	1

Download CSV
7 rows selected.

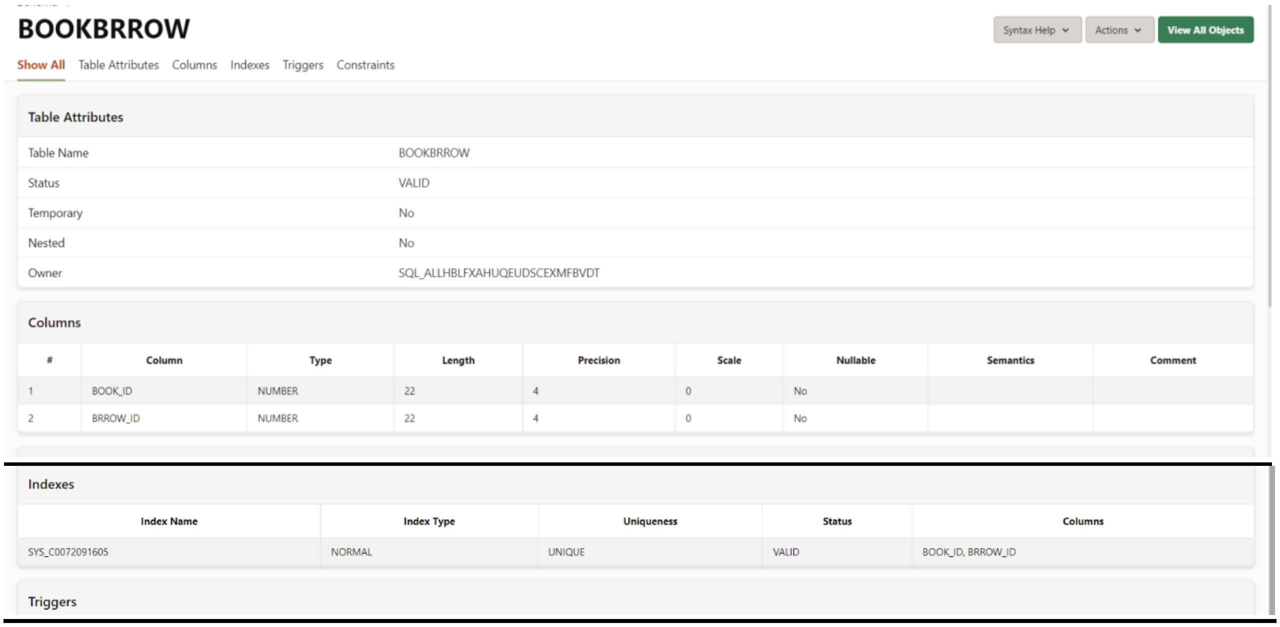
Here we insert the value to the table and we show the table with **SELECT*FROM BRROWED;** To show the table.

Table 6 BookBrrow:



```
------------------------------------------------
--here we create table 6 with the insert
CREATE TABLE BookBrrow (
Book_ID NUMBER(4) NOT NULL,
Brrow_ID NUMBER(4) NOT NULL,
PRIMARY KEY (Book_ID,Brrow_ID),
FOREIGN KEY (Book_Id) REFERENCES Book(Book_ID),
FOREIGN KEY (Brrow_ID) REFERENCES Brrowed(Brrow_ID)
);
```

```
Table created.
```



BOOKBRROW

Table Attributes

Table Name	BOOKBRROW
Status	VALID
Temporary	No
Nested	No
Owner	SQL_ALLHBLFXAHUQEUDSCEXMFBVDT

Columns

#	Column	Type	Length	Precision	Scale	Nullable	Semantics	Comment
1	BOOK_ID	NUMBER	22	4	0	No		
2	BRROW_ID	NUMBER	22	4	0	No		

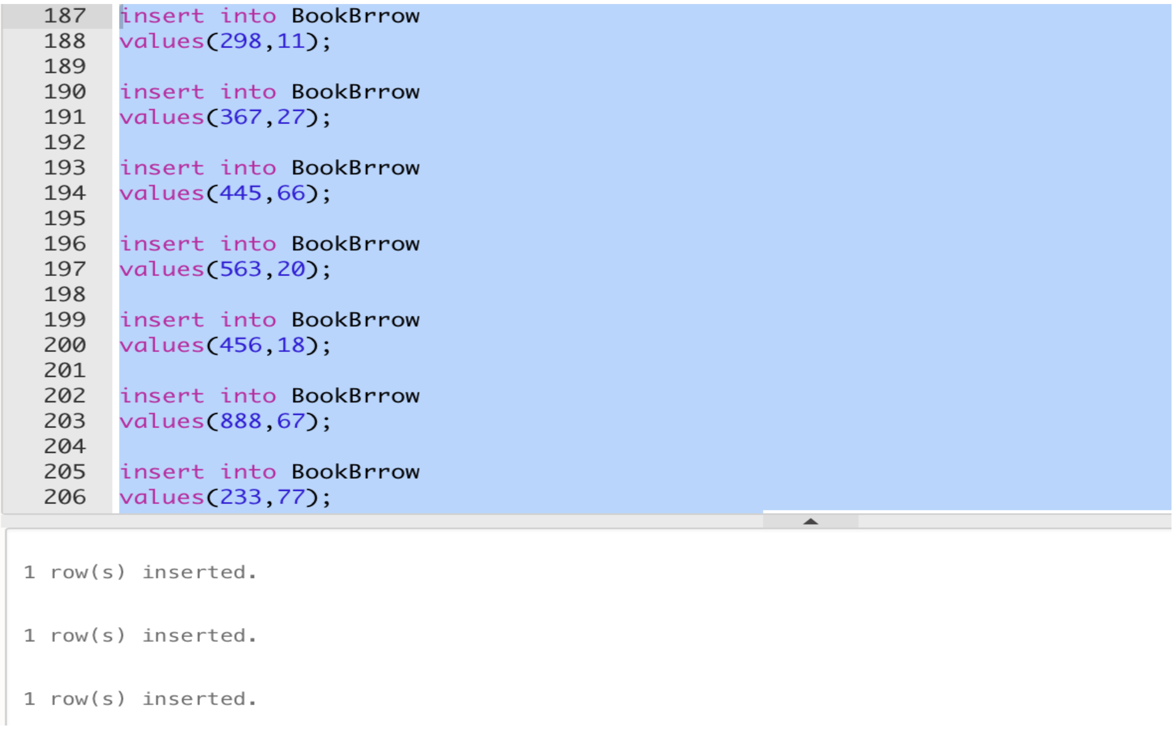
Indexes

Index Name	Index Type	Uniqueness	Status	Columns
SYS_C0072091605	NORMAL	UNIQUE	VALID	BOOK_ID, BRROW_ID

Triggers

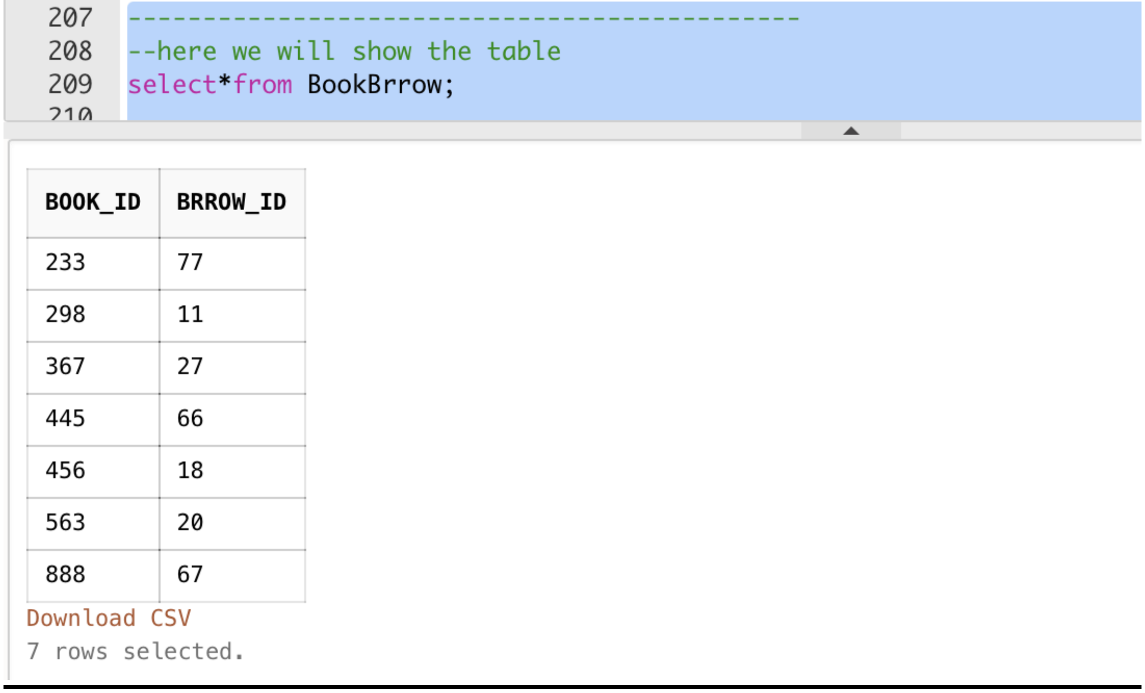
Here we create table 5 that have to forging key from Book and BRROWED And the **forging key is (Book_Id) and (Brrow_ID)** and the table also have to primary key . And the primary key is : **Book_ID, Brrow_ID.**

Insert to table 6 :



```
insert into BookBrrow
values(298,11);

insert into BookBrrow
values(367,27);

insert into BookBrrow
values(445,66);

insert into BookBrrow
values(563,20);

insert into BookBrrow
values(456,18);

insert into BookBrrow
values(888,67);

insert into BookBrrow
values(233,77);
```

```
1 row(s) inserted.

1 row(s) inserted.

1 row(s) inserted.
```



```
---------------------------------------------
--here we will show the table
select*from BookBrrow;
```

BOOK_ID	BRROW_ID
233	77
298	11
367	27
445	66
456	18
563	20
888	67

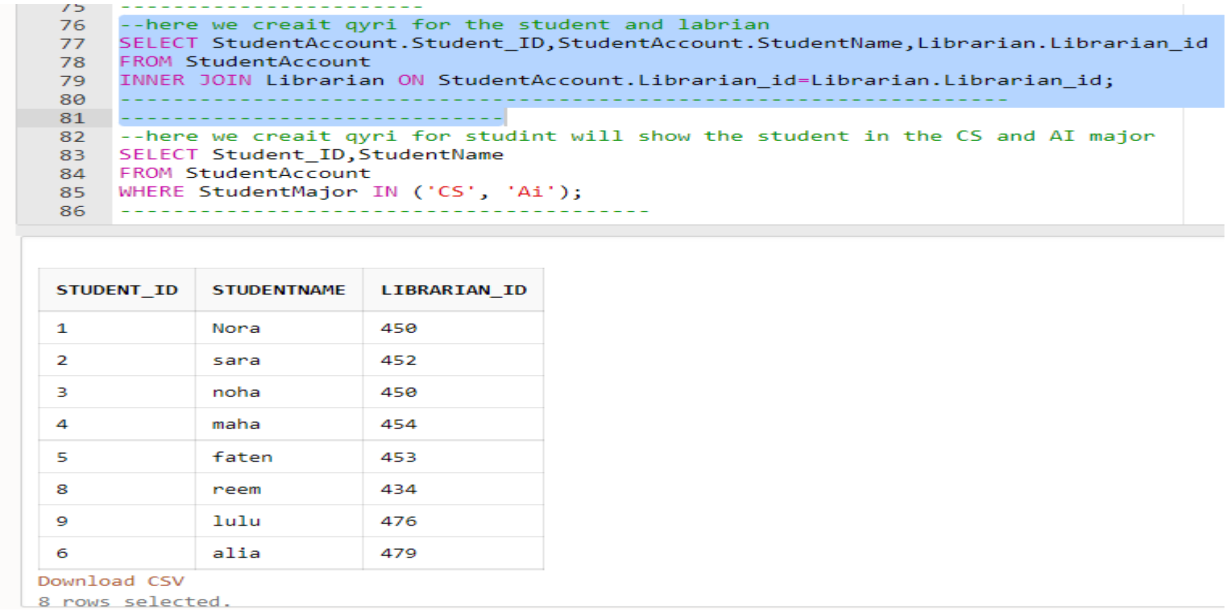
Download CSV

7 rows selected.

Here we insert all the value to the table and we **write SELECT * FROM BookBrrow** to show the table .

Part 3: Design and implement at least 4 queries.

The first **query:**



```
------------------------
--here we creait qyri for the student and labrian
SELECT StudentAccount.Student_ID,StudentAccount.StudentName,Librarian.Librarian_id
FROM StudentAccount
INNER JOIN Librarian ON StudentAccount.Librarian_id=Librarian.Librarian_id;
----------------------------------------------------------------------
------------------------------
--here we creait qyri for studint will show the student in the CS and AI major
SELECT Student_ID,StudentName
FROM StudentAccount
WHERE StudentMajor IN ('CS', 'Ai');
-----------------------------------------
```

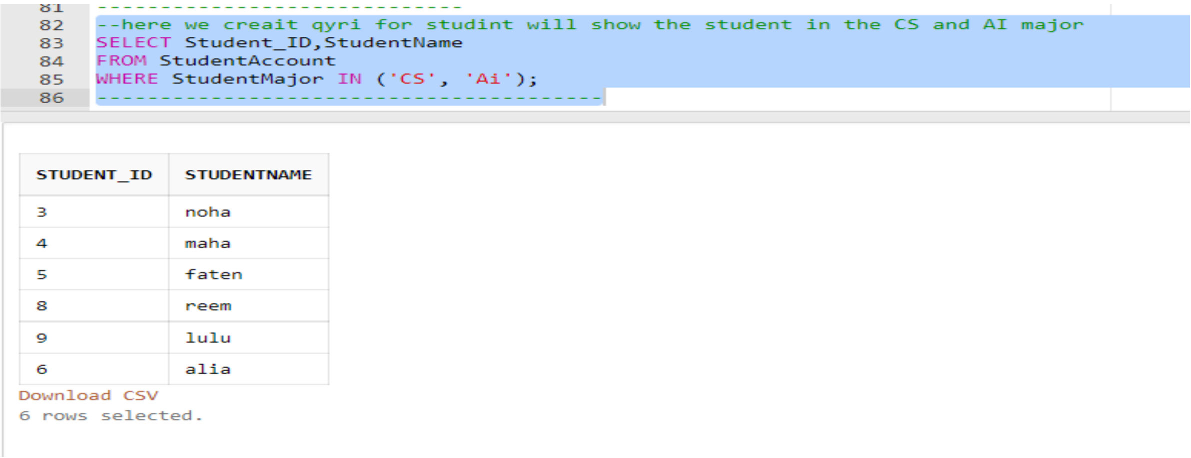
STUDENT_ID	STUDENTNAME	LIBRARIAN_ID
1	Nora	450
2	sara	452
3	noha	450
4	maha	454
5	faten	453
8	reem	434
9	lulu	476
6	alia	479

Download CSV

8 rows selected.

In the first queries we use the join to create table has the **student ID and student name** with **librarian ID**, the point of this queries is that every student will borrow book it's will has the id of librarian who give them the book so the table will show the student name and id and librarian id**.**

Second query



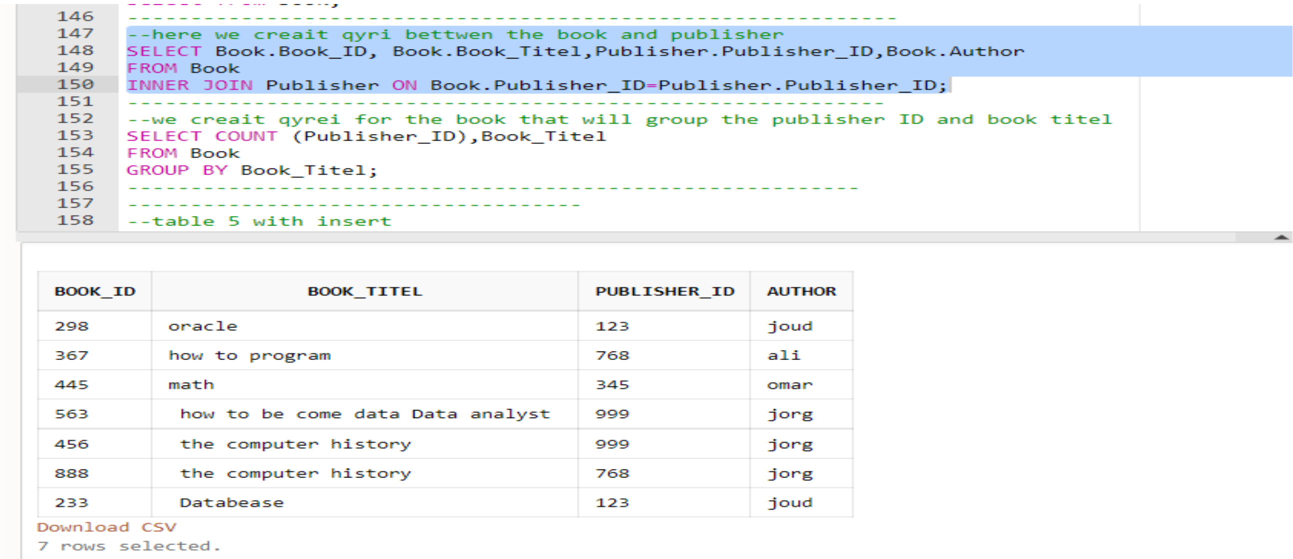
```
-----------------------------
--here we creait qyri for studint will show the student in the CS and AI major
SELECT Student_ID,StudentName
FROM StudentAccount
WHERE StudentMajor IN ('CS', 'Ai');
-----------------------------------------
```

STUDENT_ID	STUDENTNAME
3	noha
4	maha
5	faten
8	reem
9	lulu
6	alia

Download CSV
6 rows selected.

In this queries we use where to show us the student's in major 'CS' and student's in major 'AI', so the table will show the student's NAME and student's ID .

Third query:



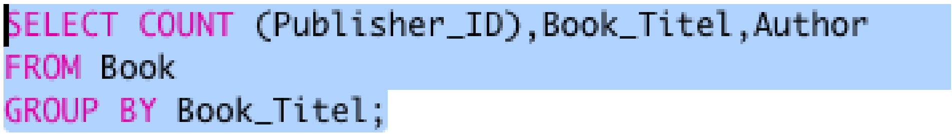
```
-------------------------------------------------------------
--here we creait qyri bettwen the book and publisher
SELECT Book.Book_ID, Book.Book_Titel,Publisher.Publisher_ID,Book.Author
FROM Book
INNER JOIN Publisher ON Book.Publisher_ID=Publisher.Publisher_ID;
-------------------------------------------------------------
--we creait qyrei for the book that will group the publisher ID and book titel
SELECT COUNT (Publisher_ID),Book_Titel
FROM Book
GROUP BY Book_Titel;
----------------------------------------------------------
-----------------------------------
--table 5 with insert
```

BOOK_ID	BOOK_TITEL	PUBLISHER_ID	AUTHOR
298	oracle	123	joud
367	how to program	768	ali
445	math	345	omar
563	how to be come data Data analyst	999	jorg
456	the computer history	999	jorg
888	the computer history	768	jorg
233	Databease	123	joud

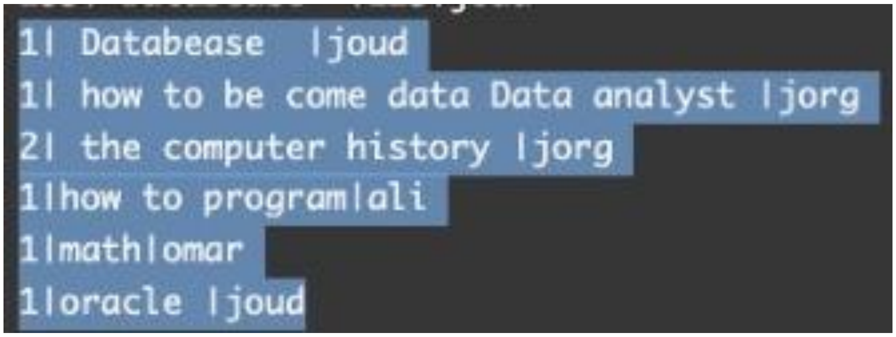
Download CSV
7 rows selected.

In this query we use the joint between the publisher and the book tables, this queries we help us to see every **book name** who is the **author of the book, and the publisher who publish the book**..So, the table shows the **book_id** and **book_title** and the **publisher id,** and the **author of the book**

Forth query :



```
SELECT COUNT (Publisher_ID),Book_Titel,Author
FROM Book
GROUP BY Book_Titel;
```

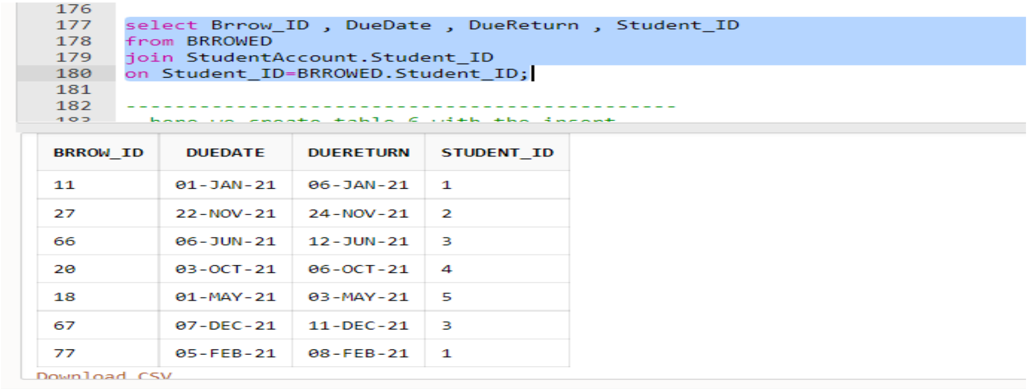


```
1| Databease  |joud
1| how to be come data Data analyst |jorg
2| the computer history |jorg
1|how to program|ali
1|math|omar
1|oracle |joud
```

For this query we use GROUP BY and we count the Publisher_ID and we collected them in group by Book_Titel , we also needed the Author to show the name of book author.

Since when the SQL was constantly giving us an error even though the code was correct , we had to use another compiler:
https://www.mycompiler.io/new/sql

Fifth query:



```
select Brrow_ID , DueDate , DueReturn , Student_ID
from BRROWED
join StudentAccount.Student_ID
on Student_ID=BRROWED.Student_ID;

-----------------------------------------------
  here we create table 6 with the insert
```

BRROW_ID	DUEDATE	DUERETURN	STUDENT_ID
11	01-JAN-21	06-JAN-21	1
27	22-NOV-21	24-NOV-21	2
66	06-JUN-21	12-JUN-21	3
20	03-OCT-21	06-OCT-21	4
18	01-MAY-21	03-MAY-21	5
67	07-DEC-21	11-DEC-21	3
77	05-FEB-21	08-FEB-21	1

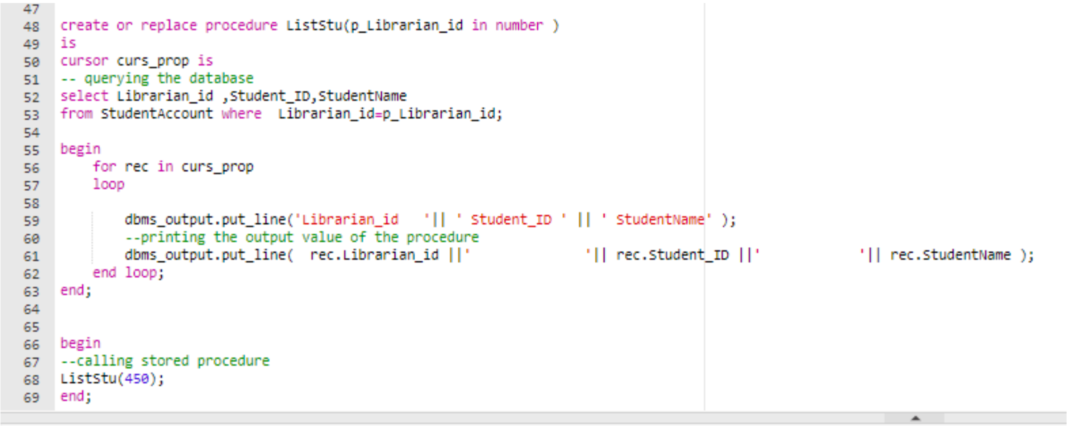
Download CSV

In this query, we use JOIN between BRROWED and StudentAccount and show the BRROW_ID and DEUDATE AND DEURETRUN and STUDENT_ID of the student who take the book.

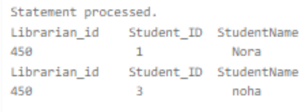
Part4: Design 2 Stored Procedures

The first Procedure:

Input the ID of the librarian, get on the list of the ID, and the names of the students who follow a particular librarian



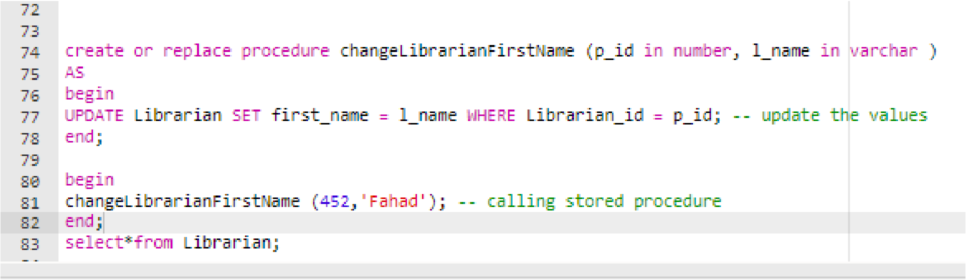
```
create or replace procedure ListStu(p_Librarian_id in number )
is
cursor curs_prop is
-- querying the database
select Librarian_id ,Student_ID,StudentName
from StudentAccount where  Librarian_id=p_Librarian_id;

begin
    for rec in curs_prop
    loop

        dbms_output.put_line('Librarian_id   '|| ' Student_ID ' || ' StudentName' );
        --printing the output value of the procedure
        dbms_output.put_line(  rec.Librarian_id ||'              '|| rec.Student_ID ||'            '|| rec.StudentName );
    end loop;
end;


begin
--calling stored procedure
ListStu(450);
end;
```



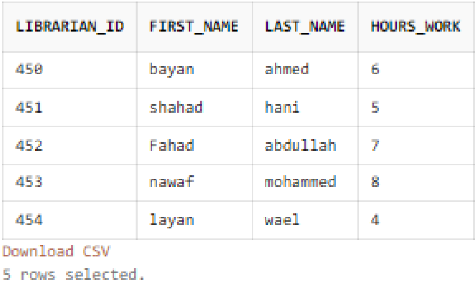
```
Statement processed.
Librarian_id    Student_ID  StudentName
450              1            Nora
Librarian_id    Student_ID  StudentName
450              3            noha
```

The Second Procedure:

Update the name of the librarian by inputting the ID librarian and the new name



```


create or replace procedure changeLibrarianFirstName (p_id in number, l_name in varchar )
AS
begin
UPDATE Librarian SET first_name = l_name WHERE Librarian_id = p_id; -- update the values
end;

begin
changeLibrarianFirstName (452,'Fahad'); -- calling stored procedure
end;
select*from Librarian;
```



LIBRARIAN_ID	FIRST_NAME	LAST_NAME	HOURS_WORK
450	bayan	ahmed	6
451	shahad	hani	5
452	Fahad	abdullah	7
453	nawaf	mohammed	8
454	layan	wael	4

Download CSV
5 rows selected.