

AD 688
 Tiange Chang
 Assignment 2
 2/22/2022

Section 1

3. SELECT * FROM Book; SELECT * FROM Patron; SELECT * FROM Loan;

	call_no	title	subject		user_id	name	age		call_no	user_id	fine	paid
1	100	Physics Handbook	Physics	1	100	Wong	22	1	100	100	NULL	yes
2	200	Database Systems	Computing	2	150	Colin	31	2	300	100	NULL	NULL
3	300	Modula-2	Computing	3	200	King	21	3	900	200	1.9	yes
4	400	Database Design	Computing	4	250	Das	67	4	400	200	16.3	yes
5	500	Software Testing	Computing	5	300	Niall	17	5	600	200	16.3	yes
6	600	Business Society	Business	6	350	Smith	72	6	500	250	NULL	NULL
7	700	Graphs	Mathematics	7	400	Jones	41	7	600	250	36.5	yes
8	800	Cell Biology	Biology					8	700	300	NULL	NULL
9	900	Set Theory	Mathematics					9	800	350	2.9	yes
								10	900	400	NULL	NULL

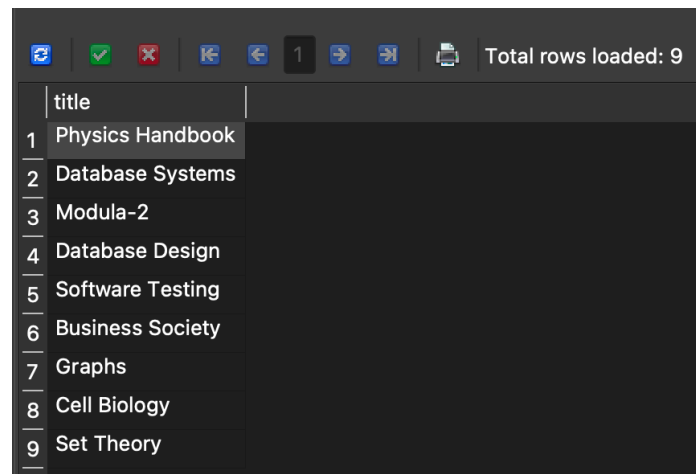
4. ALTER table Patron ADD address varchar(30);

	user_id	name	age	address
1	100	Wong	22	NULL
2	150	Colin	31	NULL
3	200	King	21	NULL
4	250	Das	67	NULL
5	300	Niall	17	NULL
6	350	Smith	72	NULL
7	400	Jones	41	NULL

5. CREATE TABLE Seniors (user_id INT NOT NULL, name varchar(50) NOT NULL, age INT NOT NULL, address varchar(30)); INSERT INTO Seniors SELECT * From Patron where age > 65; SELECT * From Seniors;

	user_id	name	age	address
1	250	Das	67	NULL
2	350	Smith	72	NULL

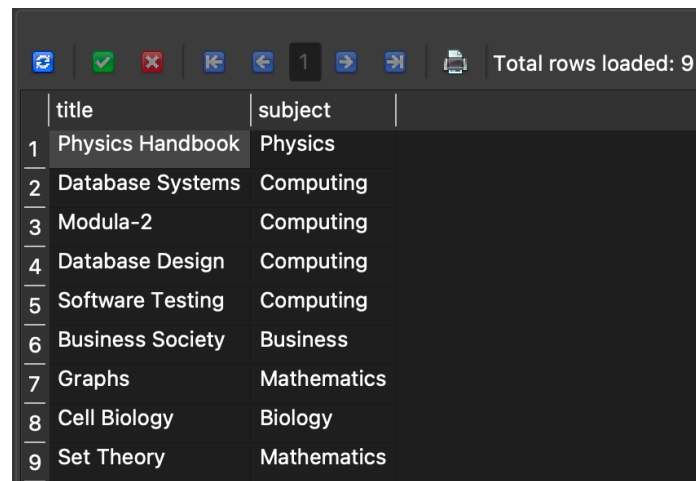
6. `SELECT DISTINCT title FROM Book;`



The screenshot shows a database query result with 9 rows. The header row is 'title'. The rows are numbered 1 to 9. The titles are: Physics Handbook, Database Systems, Modula-2, Database Design, Software Testing, Business Society, Graphs, Cell Biology, and Set Theory. The status bar at the top indicates 'Total rows loaded: 9'.

	title
1	Physics Handbook
2	Database Systems
3	Modula-2
4	Database Design
5	Software Testing
6	Business Society
7	Graphs
8	Cell Biology
9	Set Theory

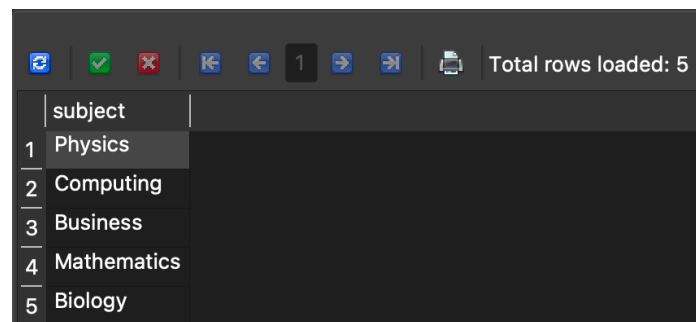
7. `SELECT title, subject FROM Book;`



The screenshot shows a database query result with 9 rows. The header row has two columns: 'title' and 'subject'. The rows are numbered 1 to 9. The titles and subjects are: Physics Handbook (Physics), Database Systems (Computing), Modula-2 (Computing), Database Design (Computing), Software Testing (Computing), Business Society (Business), Graphs (Mathematics), Cell Biology (Biology), and Set Theory (Mathematics). The status bar at the top indicates 'Total rows loaded: 9'.

	title	subject
1	Physics Handbook	Physics
2	Database Systems	Computing
3	Modula-2	Computing
4	Database Design	Computing
5	Software Testing	Computing
6	Business Society	Business
7	Graphs	Mathematics
8	Cell Biology	Biology
9	Set Theory	Mathematics

8. `SELECT DISTINCT subject FROM Book;`



The screenshot shows a database query result with 5 rows. The header row is 'subject'. The rows are numbered 1 to 5. The subjects are: Physics, Computing, Business, Mathematics, and Biology. The status bar at the top indicates 'Total rows loaded: 5'.

	subject
1	Physics
2	Computing
3	Business
4	Mathematics
5	Biology

9. `SELECT title FROM Book WHERE subject = 'Mathematics';`






		Total rows loaded: 2
	title	
1	Graphs	
2	Set Theory	

10. `SELECT * FROM Book WHERE call_no = '300';`




		Total rows loaded: 1
call_no	title	subject
300	Modula-2	Computing

Section 2

1. `SELECT user_id, call_no, fine/2 as British_Pounds FROM Loan;`

1

Total rows loaded: 10

	user_id	call_no	British_Pounds
1	100	100	NULL
2	100	300	NULL
3	200	900	0.95
4	200	400	8.15
5	200	600	8.15
6	250	500	NULL
7	250	600	18.25
8	300	700	NULL
9	350	800	1.45
10	400	900	NULL

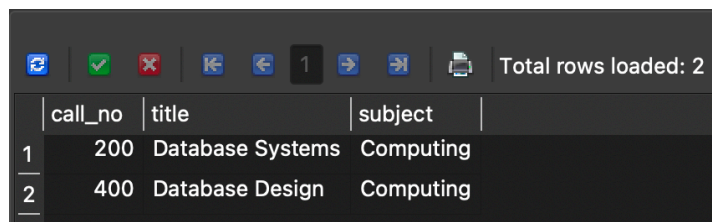
2. `SELECT user_id, call_no, fine/2 as British_Pounds FROM Loan WHERE British_Pounds > 10;`

		Total rows loaded: 1
user_id	call_no	British_Pounds
250	600	18.25

3. `SELECT call_no FROM Loan WHERE (user_id= 200 OR user_id= 250) and fine > 2;`

		Total rows loaded: 3
call_no		
400		
600		
600		

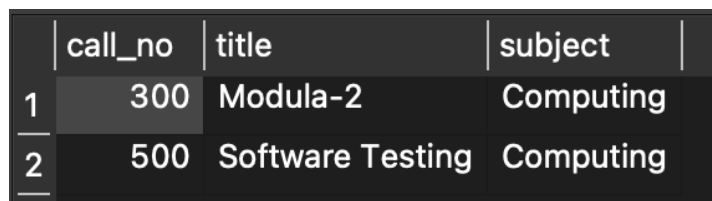
4. `SELECT * FROM Book WHERE title LIKE 'Database%';`



A screenshot of a database query result interface. At the top, there is a toolbar with icons for refresh, success, error, back, forward, and a page indicator showing '1'. To the right, it says 'Total rows loaded: 2'. Below the toolbar is a table with the following data:

	call_no	title	subject
1	200	Database Systems	Computing
2	400	Database Design	Computing

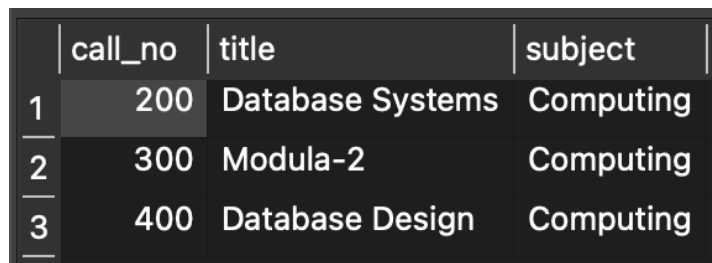
5. `SELECT * FROM Book WHERE title LIKE '_O%';`



A screenshot of a database query result interface. At the top, there is a toolbar with icons for refresh, success, error, back, forward, and a page indicator showing '1'. To the right, it says 'Total rows loaded: 2'. Below the toolbar is a table with the following data:

	call_no	title	subject
1	300	Modula-2	Computing
2	500	Software Testing	Computing

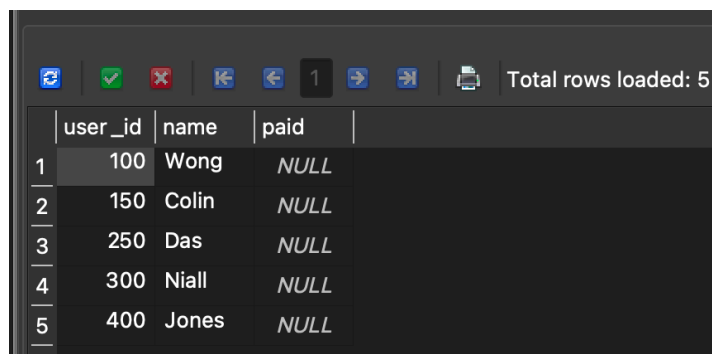
6. `SELECT * FROM Book WHERE call_no BETWEEN 200 AND 400;`



A screenshot of a database query result interface. At the top, there is a toolbar with icons for refresh, success, error, back, forward, and a page indicator showing '1'. To the right, it says 'Total rows loaded: 3'. Below the toolbar is a table with the following data:

	call_no	title	subject
1	200	Database Systems	Computing
2	300	Modula-2	Computing
3	400	Database Design	Computing

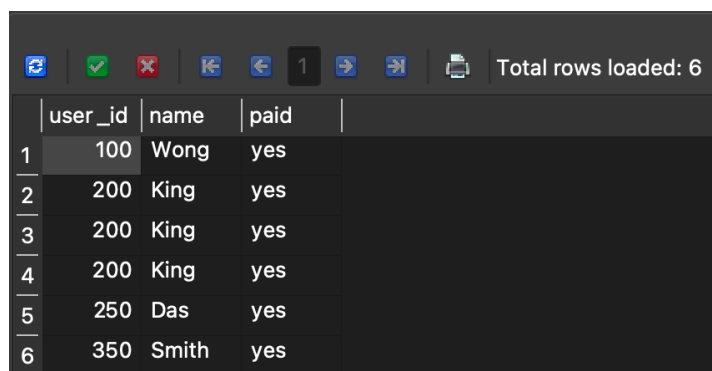
7. `SELECT user_id, name, paid FROM Patron LEFT JOIN Loan ON Patron.user_id = Loan.user_id WHERE paid IS NULL;`



A screenshot of a database query result interface. At the top, there is a toolbar with icons for refresh, success, error, back, forward, and a page indicator showing '1'. To the right, it says 'Total rows loaded: 5'. Below the toolbar is a table with the following data:

	user_id	name	paid
1	100	Wong	NULL
2	150	Colin	NULL
3	250	Das	NULL
4	300	Niall	NULL
5	400	Jones	NULL

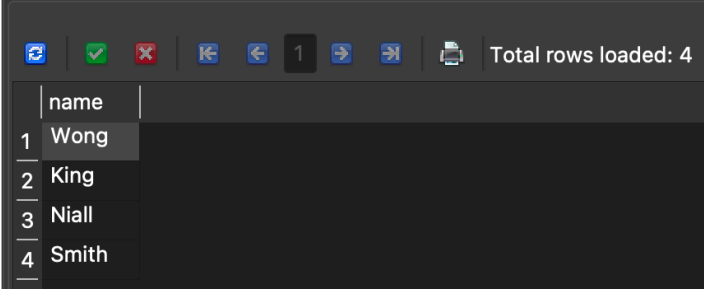
8. `SELECT user_id, name, paid FROM Patron LEFT JOIN Loan ON Patron.user_id = Loan.user_id WHERE paid IS NOT NULL;`



A screenshot of a database query result interface. At the top, there is a toolbar with icons for refresh, success, error, back, forward, and a page indicator showing '1'. To the right, it says 'Total rows loaded: 6'. Below the toolbar is a table with the following data:

	user_id	name	paid
1	100	Wong	yes
2	200	King	yes
3	200	King	yes
4	200	King	yes
5	250	Das	yes
6	350	Smith	yes

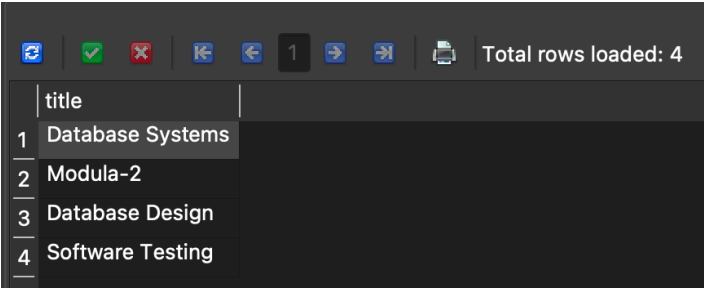
9. `SELECT name FROM Patron WHERE user_id IN (100, 200, 300, 350);`



A screenshot of a database query result interface. At the top, there is a toolbar with icons for refresh, check, error, back, forward, and a page indicator showing '1'. To the right of the toolbar, it says 'Total rows loaded: 4'. Below the toolbar is a table with a single column labeled 'name'. The table contains four rows of data: 'Wong', 'King', 'Niall', and 'Smith'.

	name
1	Wong
2	King
3	Niall
4	Smith

10. `SELECT title FROM Book WHERE subject IN ('Computing', 'History');`



A screenshot of a database query result interface. At the top, there is a toolbar with icons for refresh, check, error, back, forward, and a page indicator showing '1'. To the right of the toolbar, it says 'Total rows loaded: 4'. Below the toolbar is a table with a single column labeled 'title'. The table contains four rows of data: 'Database Systems', 'Modula-2', 'Database Design', and 'Software Testing'.

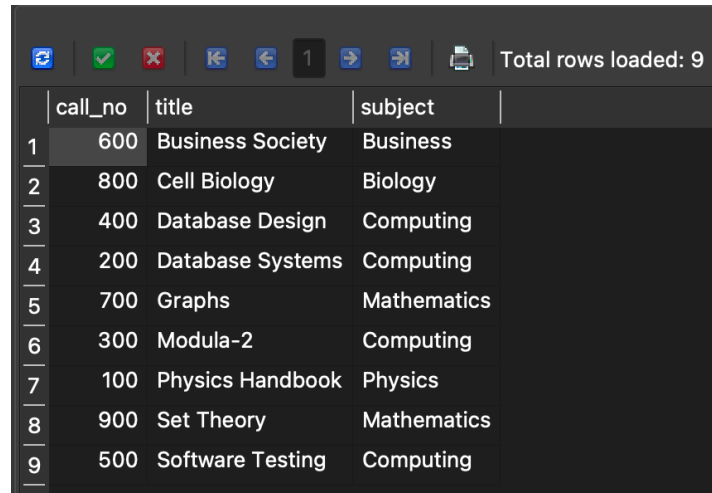
	title
1	Database Systems
2	Modula-2
3	Database Design
4	Software Testing

Section 3

1. `SELECT MAX(fine) FROM Loan;`
The largest fine paid for an overdue book was **36.5**.
2. `SELECT MIN(fine) FROM Loan;`
The least fine paid for an overdue book was **1.9**.
3. `SELECT SUM(fine) FROM Loan;`
The library collected **73.9** in fines.
4. `SELECT AVG(fine) FROM Loan;`
The average fine was **14.78**.
5. `SELECT COUNT(title) FROM Book;`
There were **9** books in the library.
6. `SELECT COUNT(paid) FROM Loan;`
6 times has a fine been collected.
7. `SELECT COUNT(subject) FROM Book WHERE subject = 'Computing';`
4 books were computing books.
8. `SELECT COUNT(DISTINCT subject) FROM Book;`
5 subject areas were in the library.

Section 4

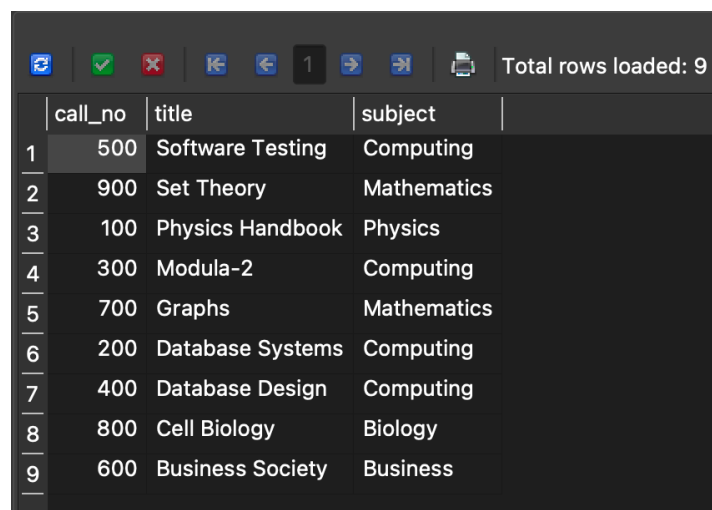
1. SELECT * FROM Book ORDER BY title ASC;



The screenshot shows a database query result with 9 rows. The columns are call_no, title, and subject. The results are ordered by title in ascending order. The interface includes a toolbar with icons for refresh, check, error, back, forward, and a page indicator showing '1'. The status bar indicates 'Total rows loaded: 9'.

	call_no	title	subject
1	600	Business Society	Business
2	800	Cell Biology	Biology
3	400	Database Design	Computing
4	200	Database Systems	Computing
5	700	Graphs	Mathematics
6	300	Modula-2	Computing
7	100	Physics Handbook	Physics
8	900	Set Theory	Mathematics
9	500	Software Testing	Computing

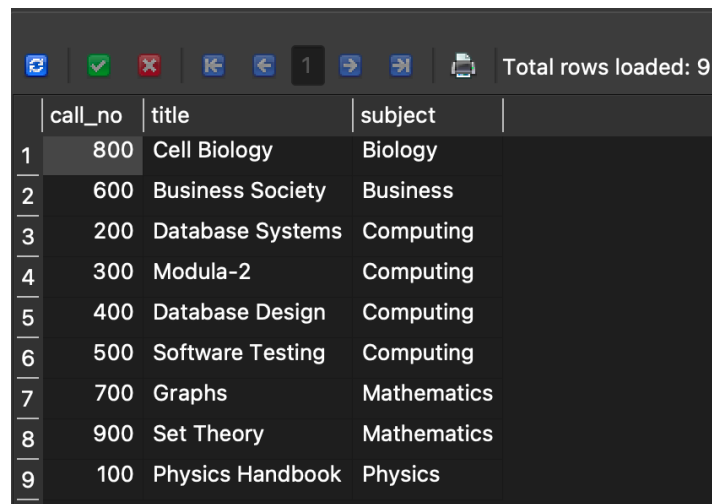
SELECT * FROM Book ORDER BY title DESC;



The screenshot shows a database query result with 9 rows. The columns are call_no, title, and subject. The results are ordered by title in descending order. The interface includes a toolbar with icons for refresh, check, error, back, forward, and a page indicator showing '1'. The status bar indicates 'Total rows loaded: 9'.

	call_no	title	subject
1	500	Software Testing	Computing
2	900	Set Theory	Mathematics
3	100	Physics Handbook	Physics
4	300	Modula-2	Computing
5	700	Graphs	Mathematics
6	200	Database Systems	Computing
7	400	Database Design	Computing
8	800	Cell Biology	Biology
9	600	Business Society	Business

2. SELECT * FROM Book ORDER BY subject, call_no;



The screenshot shows a database query result with 9 rows. The columns are call_no, title, and subject. The results are ordered by subject, and then by call_no within each subject group. The interface includes a toolbar with icons for refresh, check, error, back, forward, and a page indicator showing '1'. The status bar indicates 'Total rows loaded: 9'.

	call_no	title	subject
1	800	Cell Biology	Biology
2	600	Business Society	Business
3	200	Database Systems	Computing
4	300	Modula-2	Computing
5	400	Database Design	Computing
6	500	Software Testing	Computing
7	700	Graphs	Mathematics
8	900	Set Theory	Mathematics
9	100	Physics Handbook	Physics

3. `SELECT user_id, SUM(fine) FROM Loan GROUP BY user_id;`

	user_id	SUM(fine)
1	100	NULL
2	200	34.5
3	250	36.5
4	300	NULL
5	350	2.9
6	400	NULL

4. `SELECT user_id, SUM(fine), call_no FROM Loan WHERE call_no > 400 GROUP BY user_id HAVING SUM(fine) > 30;`

	user_id	SUM(fine)	call_no
1	250	36.5	500

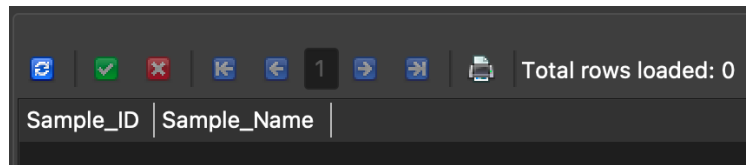
Section 5

1. `SELECT name, user_id, call_no FROM Patron LEFT JOIN Loan ON Patron.user_id = Loan.user_id LEFT JOIN Book On Book.call_no = Loan.call_no;`

	name	user_id	call_no
1	Wong	100	100
2	Wong	100	300
3	Colin	150	NULL
4	King	200	400
5	King	200	600
6	King	200	900
7	Das	250	500
8	Das	250	600
9	Niall	300	700
10	Smith	350	800
11	Jones	400	900

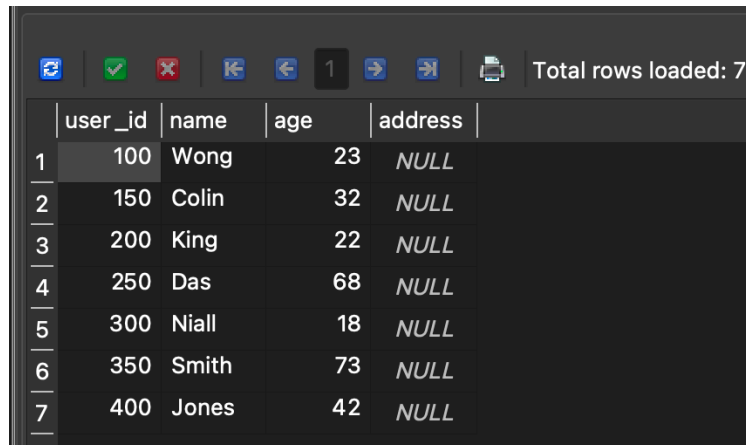
Section 6

1. CREATE TABLE Sample_Table(Sample_ID INT NOT NULL, Sample_Name varchar(60)); SELECT * FROM Sample_Table;



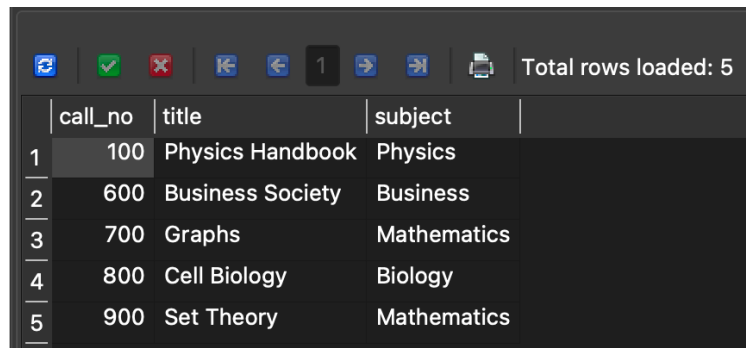
Sample_ID	Sample_Name
-----------	-------------

2. DROP TABLE Sample_Table;
After running this code, the table was deleted.
3. UPDATE Patron SET age = age + 1; SELECT * FROM Patron;



	user_id	name	age	address
1	100	Wong	23	NULL
2	150	Colin	32	NULL
3	200	King	22	NULL
4	250	Das	68	NULL
5	300	Niall	18	NULL
6	350	Smith	73	NULL
7	400	Jones	42	NULL

4. DELETE FROM Book WHERE subject = 'Computing'; SELECT * FROM Book;



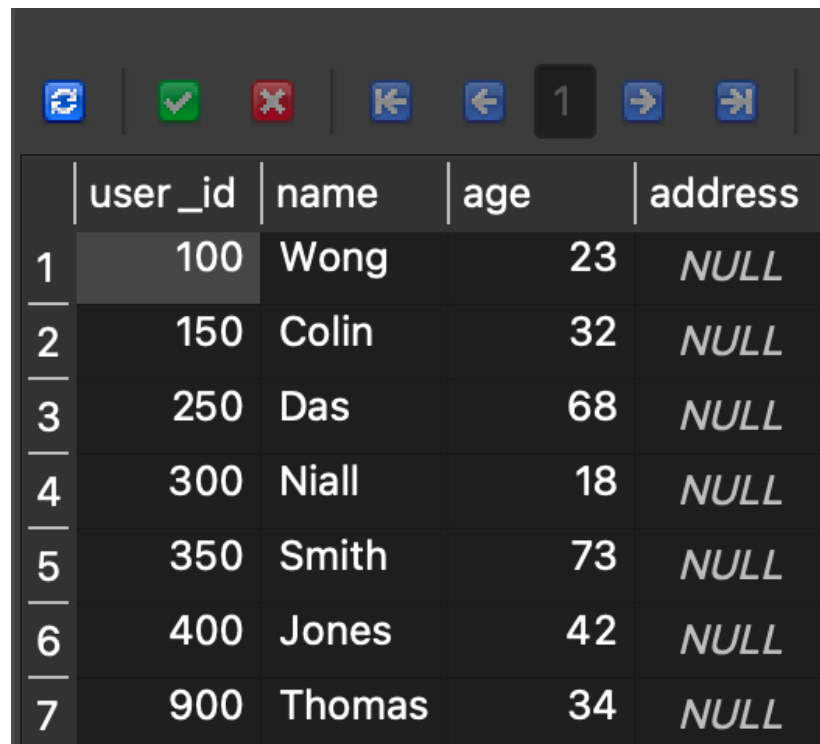
	call_no	title	subject
1	100	Physics Handbook	Physics
2	600	Business Society	Business
3	700	Graphs	Mathematics
4	800	Cell Biology	Biology
5	900	Set Theory	Mathematics

5. DELETE FROM Loan WHERE user_id = (SELECT user_id FROM Patron WHERE name = 'King'); SELECT user_id, name, call_no, fine, paid FROM Loan LEFT JOIN Patron ON Loan.user_id = Patron.user_id;



	user_id	name	call_no	fine	paid
1	100	Wong	100	NULL	yes
2	100	Wong	300	NULL	NULL
3	250	Das	500	NULL	NULL
4	250	Das	600	36.5	yes
5	300	Niall	700	NULL	NULL
6	350	Smith	800	2.9	yes
7	400	Jones	900	NULL	NULL

6. INSERT INTO Patron(name, user_id, age) VALUES('Thomas', 900, 34); SELECT * FROM Patron;



	user_id	name	age	address
1	100	Wong	23	NULL
2	150	Colin	32	NULL
3	250	Das	68	NULL
4	300	Niall	18	NULL
5	350	Smith	73	NULL
6	400	Jones	42	NULL
7	900	Thomas	34	NULL