



Temasek Junior College

2024 JC2 H2 Computing

Database 4 – Basic Operations in DB Browser for SQLite

Section	3	Data and Information
Unit	3.3	Databases and Data Management
Objectives	3.3.8	Write SQL statements and use a programming language to work with SQL databases

1 What is SQLite?

SQLite is a lightweight, self-contained, embedded database engine. Unlike traditional database systems, SQLite doesn't require a separate server process to function. It is a library that developers can integrate into their applications to manage data.

You can find out more about what SQLite is on <https://www.sqlite.org/about.html>.

2 What is the DB Browser for SQLite?

DB Browser for SQLite is a simple and easy to use Graphical User Interface (GUI) - based software for the creation and editing of database files compatible with SQLite. It abstracts and hides the details of complex SQL commands while providing an easy to user interface for performing the same database operations.

You can find out more about what DB Browser for SQLite is on <https://sqlitebrowser.org/>.

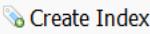
3 File Handling Features

The table below summarises the key file handling features of the DB Browser.

Feature	Description
New Database	<ul style="list-style-type: none"> Creates a new SQLite database file. Available file formats include .db or .db3 or .sqlite or .sqlite3.
Open Database	<ul style="list-style-type: none"> Opens a SQLite database file.
Write Changes	<ul style="list-style-type: none"> Saves changes to database. Equivalent to the COMMIT operation in SQL.
Revert Changes	<ul style="list-style-type: none"> Undo any changes made to the database. Equivalent to ROLLBACK operation in SQL.
Import	<ul style="list-style-type: none"> Located in the File menu Imports one of the following: <ul style="list-style-type: none"> .csv file type as a table in the database. .txt file type as a table in the database. an SQLite database file.
Export	<ul style="list-style-type: none"> Located in the File menu Exports one of the following: <ul style="list-style-type: none"> a table in the database as a new .csv file. database as a .sql file. database as a .json file.

4 Database Modification Features

The table below summarises the database modification features of the DB Browser. These features are found under the Database Structure tab.

Feature	Description
 Create Table	<ul style="list-style-type: none"> Creates a table in the database.
 Create Index	<ul style="list-style-type: none"> Defines a new index (primary key) on an existing database table.
 Modify Table	<ul style="list-style-type: none"> Modifies a table in the database. Not limited to changing table name or field name, adding fields or constraints.
 Delete Table	<ul style="list-style-type: none"> Deletes a table from the database.

4.1 Constraints

Constraints are rules enforced on the data fields and/or table. They limit the type of data that can go into a field and/or table, ensuring the accuracy and reliability of the data in the database.

Table constraints are applicable throughout the entire table while field constraints are only applicable for the field for which it is applied on.

The table below summarises the constraints available in the DB Browser.

Feature	Description
Type	<ul style="list-style-type: none"> Determines the data type accepted for the field.
Not null	<ul style="list-style-type: none"> Ensures that a field cannot have NULL values.
PK	<ul style="list-style-type: none"> Identifies a field or a set of fields as the primary key.
AI	<ul style="list-style-type: none"> Automatically increments the value of the field for each new record. Works for integer values only.
U	<ul style="list-style-type: none"> Ensures that all values in a field are unique.
Default	<ul style="list-style-type: none"> Provides a default value to the field for a new record when none is specified.
Check	<ul style="list-style-type: none"> Ensures that all values in a field satisfy a condition or a set of conditions. If any condition evaluates to false, the record will not be entered into the table.
Foreign Key	<ul style="list-style-type: none"> Identifies a field in one table as the foreign key that references a field in another table.

5 Data Browsing and Modification Features

The table below summarises the data browsing and modification features available in the DB Browser. These features are found under the Browse Data tab.

Feature	Description
	<ul style="list-style-type: none"> Refresh data in the selected table.
	<ul style="list-style-type: none"> Clear all filters.
New Record	<ul style="list-style-type: none"> Creates a new record in the table.
Delete Record	<ul style="list-style-type: none"> Deletes a record in the table.

6 Practice Task

Task 1 – Creation of Database and Database Tables

The task is to create a database, `library.db`, with four tables using DB Browser for SQLite.

The library contains books that can be loaned to borrowers.

- A borrower can take one or more loans.
- Each loan record belongs to only one borrower.
- A book can be loaned many times.
- A publisher can publish one or more books.
- A book can be published by zero or one publisher which may or may not be an official publishing house e.g school lecture notes are not published by an official publishing house.

The first table you will create is the `Borrower` table as shown below.

Field	Type
ID	INTEGER
FirstName	TEXT
Surname	TEXT
Contact	TEXT

The constraints to be applied to the table are:

- `ID` is the primary key of the `Borrower` table. This means that `ID` is used to identify a borrower.
- The value of `ID` should be `AUTOINCREMENT`. This means that the `ID` value increases automatically with each new record inserted.
- All fields are `NOT NULL`. This means each field cannot be empty.

Steps for Task 1

1. Create a folder called `DBTASK`. You will save all your files inside this folder.
2. Open DB Browser.
3. Click on  `New Database`.
4. Save your database file as `library.db`.
The default extension is `.db`.
(Note: other database file extensions are `.sqlite/.sqlite3/.db3`).
5. Under the Database Structure tab, click on  `Create Table` to create the `Borrower` table with the fields and constraints listed above.
6. Click on  `Write Changes` or `CTRL + S` to save all changes to the database.
This saves all the changes made but does not close the database file.

Task 2 – Manual Insertion of Records into Database Tables

The task is to manually add four records to the `Borrower` table.

The records to be added are as follows:

ID	FirstName	Surname	Contact
1	Jerlyn	Lim	91111111
2	Andrel	Quoid	82222222
3	Jiaxuan	Tang	93333333
4	Phuc Truong	Nguyen	84444444

Steps for Task 2

- Under the Browse Data tab, select the `Borrower` table from the dropdown, then click on `New Record`. The `ID` field is auto-populated due to the use of `AUTOINCREMENT`.



- Click on the `FirstName` field of the new record. Enter its value according to the table of records given. Press `Enter` to register the input value.
- Repeat **Step 2** for the `Surname` and `Contact` fields.
- Repeat **Step 1** to **Step 3** to enter the remaining records.
- Click on `Write Changes` or `CTRL + S` to save all changes to the database. This saves all the changes made but does not close the database file.

Task 3 – Updating of Records

The contact number of one of the borrowers, Andrel, is incorrect.

The task is to update her contact number to 81234567.

Steps for Task 3

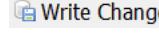
- Click on the `Contact` field of the relevant record.
- Update the field with the new value. Press `Enter` to register the input value.
- Click on `Write Changes` or `CTRL + S` to save all changes to the database. This saves all the changes made but does not close the database file.

Task 4 – Deletion of Records

One of the borrowers, Jiaxuan, has returned all borrowed items.

The task is to delete the record.

Steps for Task 4

1. Select the entire record by clicking on the serial number located before the first field (on the left end of the record).
2. Click on  .
3. Click on  or **CTRL** + **S** to save all changes to the database.
This saves all the changes made but does not close the database file.

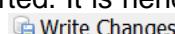
Task 5 – Reverting Changes

The task is to make an erroneous update, then revert the change.

Steps for Task 5

1. Now make an erroneous change to Phuc Truong's contact number by entering the value 12345678. Press **Enter** to register the change.
2. Click on  .
A warning dialog with the message “Are you sure you want to undo all changes to the database file ... since the last save?”
3. Click on “Yes” to undo the changes made.

Important Note

Once  or **CTRL** + **S** has been performed to save all changes to the database, the changes can no longer be reverted. It is hence important to ascertain that all changes made are correct before performing  or **CTRL** + **S**.

Task 6 – Linking Different Tables

The library database shall also contain the Publisher and Book tables.

The Publisher table will be as follows:

Column Name	Type
ID	INTEGER
Name	TEXT

The constraints to be applied to the table are:

- ID is the primary key of the Publisher table.
- The value of ID should be AUTOINCREMENT.
- All fields are NOT NULL.

The records to be added to the Publisher table are shown below:

ID	Name
1	NPH
2	Unprop
3	Appleson
4	Squirrel
5	Yellow Flame

The Book table will be as follows:

Column Name	Type
ID	INTEGER
Title	TEXT
PublisherID	INTEGER
Damaged	INTEGER

The constraints to be applied to the table are:

- ID is the primary key of the Book table.
- ID, Title and Damaged fields are NOT NULL.
- Damaged is an attribute that tracks the condition of the book:
 - A value of 0 means that the book is not damaged.
 - A value of 1 means that the book is damaged.
- PublisherID is a foreign key to ID in the Publisher table.
- The Book table can only be created after the Publisher table because of the foreign key reference to ID in the Publisher table.

The records to be added to the Book table are shown below:

ID	Title	PublisherID	Damaged
1	The Lone Gatsby	5	0
2	A Winter's Slumber	4	1
3	Life of Pie	4	0
4	A Brief History Of Primates	3	0
5	To Praise a Mocking Bird	2	0
6	The Catcher in the Eye	1	1
123	H2 Computing Ten Year Series	NULL	0

The task is to create the Publisher and Book tables.

Steps for Task 6

1. Try this task yourself 😊

Task 7 – Importing Data

Information on loans are contained in a text file loan.txt.

The task is to create the Loan table by importing data from loan.txt into the database.

The Loan table will be as follows:

Column Name	Type
ID	INTEGER
BorrowerID	INTEGER
BookID	INTEGER
DateBorrowed	TEXT

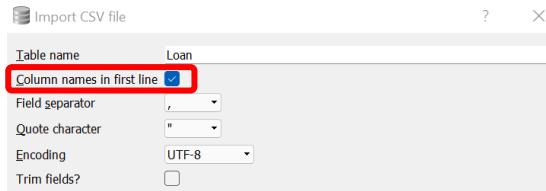
The constraints to be applied to the table are:

- ID is the primary key of the Loan table.
- The value of ID should be AUTOINCREMENT.
- ID, BorrowerID and BookID fields are NOT NULL.
- BorrowerID is a foreign key to ID in the Borrower table.
- BookID is a foreign key to ID in the Book table.

Steps for Task 7

1. Create the `Loan` table using the `Import` feature. This feature allows importing of `.txt` and `.csv` files.
2. Import the text file `loan.txt` from its storage location.
3. Check the option “Column names in first line” in the dialog box that appears. This will create the field names using the first line of the text file.

It is always a good habit to check whether the first line of the data file contains the field names. The option need not be checked if the first row do not contain the field names.



4. Click “OK”.
5. Click “Modify Table”.
6. Adjust the data type according to the description above.

The data type for every field in the table will be defaulted to `TEXT` during an import. Hence it is important adjust the data types where appropriate, after an import.

Name	Type	Not	PK	AI	U	Default	Check
ID	INTEGER						
BorrowerID	INTEGER						
BookID	INTEGER						
DateBorrowed	TEXT						

7. Apply the appropriate constraints to each field.
8. To create the foreign key reference to `ID` in the `Borrower` table, double click on the whitespace for the Foreign Key column that corresponds to `BorrowerID`.
In the dropdowns that appear, select the `BORROWER` table for the first dropdown, then the `ID` field for the second dropdown.
9. Repeat the **Step 8** for `BookID` to create the foreign key reference.
10. View the `Loan` table from Browse Data tab. You should see the following data:

ID	BorrowerID	BookID	DateBorrowed
1	3	2	20180220
2	3	1	20171215
3	2	3	20171231
4	1	5	20180111

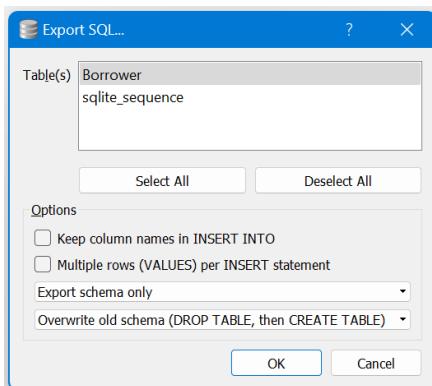
11. Click on `Write Changes` or `CTRL + S` to save all changes to the database.
This saves all the changes made but does not close the database file.

Task 8 – Exporting the Database Schema as a .sql File

The task is to extract the database schema as a .sql file.

Steps for Task 8

1. On the File menu, select “Export”, then “Database to SQL file...”
2. On the dialog box that appears, perform the following:
 - Select the table(s) required. To select more than one table, press and hold the **Ctrl** key, then click on the required tables.
Note that the `sqlite_sequence` table should not be part of the selection as it is not part of the database schema.
 - Ensure that the checkbox options are left unchecked when exporting the schema only.
 - In the dropdowns, make sure that the options selected are “Export schema only” and “Overwrite old schema (DROP TABLE, then CREATE TABLE)”.
As the purpose is to create a new database, overwriting the old schema ensures any existing components are removed, preventing data corruption to the new database being created, as well as any errors during the creation process.
 - Click “OK”.



3. In the next dialog box that appears, save the file as `library.sql` in the DBTASK folder.
4. Upon successful export, a dialog box will appear to inform you that the export has been completed. Click “OK”.
5. Open up your Notepad++ application.
6. From the File menu, open the `library.sql` file.
7. Delete the first line `BEGIN TRANSACTION`.
8. Delete the last line `COMMIT`.
9. Save the changes. This will be the final .sql file.