BBG Open Symbology

**Table of Contents**

Contents

[Document Management 4](#_Toc485768016)

[Document Summary 4](#_Toc485768017)

[Version Control 4](#_Toc485768018)

[BBG Open Symbology 5](#_Toc485768019)

[Benefits 5](#_Toc485768020)

[FIGI 7](#_Toc485768021)

[FIGI Structure 7](#_Toc485768022)

[Characters 1-2 7](#_Toc485768023)

[Character 3 7](#_Toc485768024)

[Characters 4-11 7](#_Toc485768025)

[Character 12 7](#_Toc485768026)

[Open Symbology Fields 8](#_Toc485768027)

[Name (NAME) 8](#_Toc485768028)

[Ticker (TICKER) 8](#_Toc485768029)

[Exchange Code (EXCH\_CODE) 8](#_Toc485768030)

[Security Type (SECURITY\_TYP) 9](#_Toc485768031)

[Security ID Number Description (ID\_BB\_SEC\_NUM\_DES) 9](#_Toc485768032)

[Market Sector Description (MARKET\_SECTOR\_DESCRIPTION) 9](#_Toc485768033)

[FIGI (ID\_BB\_GLOBAL) 9](#_Toc485768034)

[Composite FIGI (COMPOSITE\_ID\_BB\_GLOBAL) 9](#_Toc485768035)

[Share Class FIGI (ID\_BB\_GLOBAL\_SHARE\_CLASS\_LEVEL) 9](#_Toc485768036)

[Unique Identifier (ID\_BB\_UNIQUE) 9](#_Toc485768037)

[Security Type 2 (SECURITY\_TYP2) 9](#_Toc485768038)

[Security Description (SECURITY\_DES) 10](#_Toc485768039)

[Unique Identifier for Future Option (UNIQUE\_ID\_FUT\_OPT) 10](#_Toc485768040)

[Market Sector Number (MARKET\_SECTOR) 10](#_Toc485768041)

[Security Short Description (SECURITY\_SHORT\_DES) 10](#_Toc485768042)

[Technical Design 11](#_Toc485768043)

[API based Work Flow 11](#_Toc485768044)

[FTP based Work Flow 12](#_Toc485768045)

[Technologies 13](#_Toc485768046)

[Server Side 13](#_Toc485768047)

[Local database repository 13](#_Toc485768048)

[User Interface 13](#_Toc485768049)

[Database 13](#_Toc485768050)

[SQLite 13](#_Toc485768051)

[SQLite setup on Windows 14](#_Toc485768052)

[Browser for SQLite 15](#_Toc485768053)

[Important Links 17](#_Toc485768054)

# Document Management

## Document Summary

|  |  |
| --- | --- |
| **Version** | V1 |
| **Document Owner/Author** | Gaurav Agarwal |
| **Reviewed By** |  |

## Version Control

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Date** | **Version** | **Author** | **Section** | **Amendment** |
| 16-May-2017 | v1 | Gaurav Agarwal | All | Initial draft version of documents |
| 21-June-2017 | v2 | Gaurav Agarwal | Technology | SQLite database to be used initially. |
|  |  |  |  |  |

# BBG Open Symbology

Open Symbology is a flexible and open system **for identifying instruments across all global asset classes.** The FIGI identifiers are **available free of charge** with no material impediments on use. The identifiers can be used for research, trading and mapping. Bloomberg, as the Certified Provider, will continuously build, update, and administer FIGI identifiers to ensure their accuracy and effectiveness.

**Symbology** refers to more than a code – it is the methodology and system for defining how data is related, and how that information is conveyed.

**Open** refers to being an open data standard. FIGI is the **first and only open** **data standard for identification of financial instruments**. Open

As such, the FIGI symbology, under the auspice of the **Object Management Group,** is provided free of charge to all, and for the public trust and benefit. There is no cost recovery, licensing or re-use restrictions or hidden fees for access, use, or redistribution of the FIGI symbology.

The FIGI symbology consists of the **unique**, **persistent, unchanging alpha-numeric identifier,** as well as the multiple individual pieces of associated descriptive metadata.

Bloomberg, as the Registration Authority of the FIGI open data standard, is committed to continuously building, updating, and administering its identifiers to ensure they continue to serve as effective symbols for the broad uses required in today’s financial markets.

An effective symbology for any class of instrument must have broad coverage, be freely available, be flexible enough for use in multiple functions, allow mapping to alternative symbologies used in related functions, and be dynamic enough to immediately account for the many instruments that arise, expire, and change on a daily basis.

## Benefits

#### Coverage

The Financial Instrument Global Identifiers (FIGI) provides comprehensive global coverage of the financial markets. The ID will be invaluable for loans, futures and options that lack an identifier.

#### Consistency

Based on the same identifiers used in the Bloomberg Professional service and Bloomberg’s Enterprise Solutions, but now declared as an Open Standard.

#### Delivery

Submission form via **dedicated portal OpenFIGI.com** and the **OpenFIGI API** allows mapping to alternative symbologies.

#### Access

FIGI identifiers are available through the OpenFIGI website with no material impediments on use.

#### Usage

FIGIs can be used for a variety of purposes, including trading, research, and database mapping.

#### Uniqueness

Unique, non-changing identifier that covers all GLOBAL financial instruments with no restrictions on usage.

# FIGI

The Financial Instrument Global Identifier (FIGI) is an established global standard issued under the guidelines of the Object Management Group (OMG.org, an international, non-profit standards organization), founded in 1989.

It is a **12 character,** **alphanumeric, randomly generated ID** covering hundreds of millions of **active and inactive instruments**. In total, there are over **300 trillion potential identifiers available**. The identifier itself acts as a Uniform Resource Identifier (URI) to link to a set of metadata that uniquely and clearly describes the instrument.

FIGI is unique, non-changing symbol. FIGI closes the gap in existing symbology systems. Providing coverage across all global asset classes, real-time availability, and flexibility for use in multiple functions, firms are able to tie together disparate and fragmented symbologies, eliminate redundant mapping processes, streamline the trade workflow and reduce operational risk. FIGI also fills in the gap for asset classes that do not normally have a global identifier, including loans, futures and options.

## FIGI Structure

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **B** | **B** | **G** | **0** | **0** | **0** | **B** | **L** | **N** | **N** | **V** | **0** |
| 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 |

### Characters 1-2

Designates the **Certified Provider** that issued (minted) the corresponding FIGI. The first three positions are always filled with ‘BBG’. The **'BB' is** to identify the Certified Provider of the FIGI, Bloomberg LP.

### Character 3

Always **'G'** to designate it as a Global Identifier

### Characters 4-11

**8 digit randomly** assigned values that complete the reference ID for the set of associated metadata. Alpha-numeric values allowed, excluding vowels.

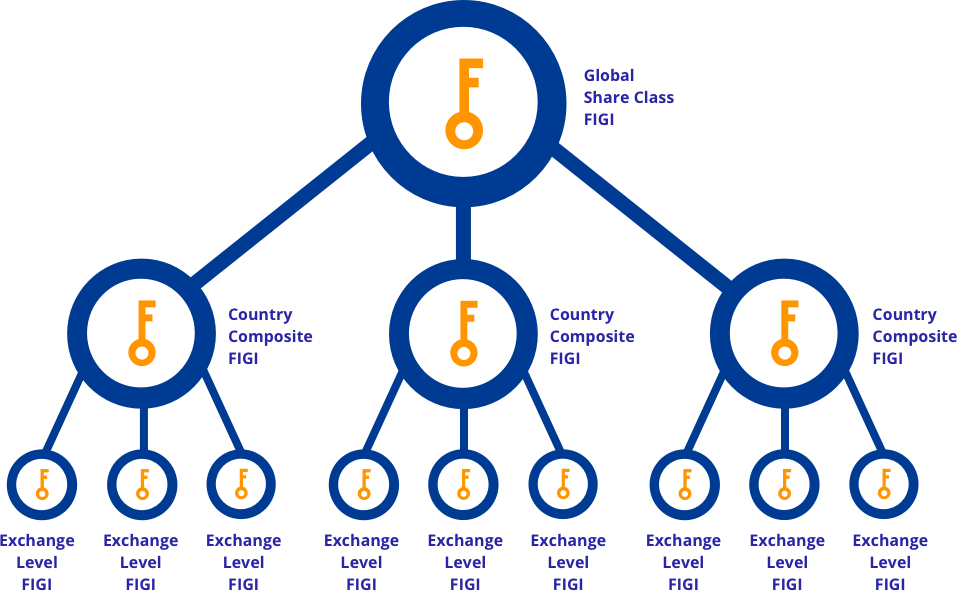
### Character 12

Check digit formula is based on the Modulus 10 Double Add Double technique and will be applied to every FIGI number.

# Open Symbology Fields

FIGI has applicability beyond simple security and instrument identification. The identifiers are used for managing the transition of hierarchal specificity as an instrument moves through the functional aspects of trading, settlement, portfolio management, asset servicing, risk management and regulatory reporting (i.e. the ‘contextual view’). The use of a unique, non-changing and perpetual identifier also enables linking material changes to that instrument, maintains permanent association with those changes and provides a consistent historical perspective of that instrument changing over time through corporate actions and other events.

**FIGI Hierarchy**



## Name (NAME)

Name of the company or brief description of the instrument. The Name of an instrument may change in conjunction with corporate actions.

## Ticker (TICKER)

Ticker is a specific identifier for a financial instrument that reflects common usage. Tickers are not, however, unique to specific exchanges or specific pricing sources. **Tickers may change in conjunction with Corporate Actions.**

## Exchange Code (EXCH\_CODE)

Code for the **trading venue or environment on which the instrument trades.** If an exchange is specified, the code will be for the specified exchange. When not specified, the code will be according to the user default exchange, which can be the composite or primary exchange.

## Security Type (SECURITY\_TYP)

Description of the **specific instrument type** within its market sector.

## Security ID Number Description (ID\_BB\_SEC\_NUM\_DES)

Descriptor for a financial instrument. Equities: Not unique on the exchange level; must be combined with Feed Source (DX282, FEED\_SOURCE) to achieve a unique value at the exchange level.

## Market Sector Description (MARKET\_SECTOR\_DESCRIPTION)

Market Sector refers to the **asset type assigned to the instrument.**

## FIGI (ID\_BB\_GLOBAL)

**Twelve character, alphanumeric identifier**. An identifier is assigned to instruments of all asset classes, is unique to an individual instrument and **once issued will not change for an instrument.** For equity instruments an identifier is issued per instrument per trading venue.

## Composite FIGI (COMPOSITE\_ID\_BB\_GLOBAL)

Twelve character, alphanumeric identifier. The Composite level of assignment is **provided in cases where there are multiple trading venues for the instrument within a single country or market.** The Composite Financial Instrument Global Identifier (FIGI) enables users to link multiple FIGIs at the trading venue-level within the same country or market in order to obtain an aggregated view for that instrument within that country or market.

## Share Class FIGI (ID\_BB\_GLOBAL\_SHARE\_CLASS\_LEVEL)

Twelve character, alpha-numeric identifier. A Share Class level Financial Instrument Global Identifier is assigned to **an instrument that is traded in more than one country.** This enables users to link multiple Composite FIGIs for the same instrument in order to obtain an aggregated view for that instrument across all countries globally.

## Unique Identifier (ID\_BB\_UNIQUE)

A legacy identifier assigned to all instruments. The construction and length of this identifier is different across asset classes. This identifier can also change as a result of corporate actions.

## Security Type 2 (SECURITY\_TYP2)

A description of the security type.

## Security Description (SECURITY\_DES)

A description of the security

## Unique Identifier for Future Option (UNIQUE\_ID\_FUT\_OPT)

Unique ticker with logic for index, currency, single stock futures, commodities and commodity options. This identifier differs from the Unique Identifier in that it is a logical ticker.

## Market Sector Number (MARKET\_SECTOR)

Number of the market sector of the security.

Possible returns are:

1 = Commodity

2 = Equity

3 = Municipals

4 = Preferred

6 = Money Market

7 = Government

8 = Corporate

9 = Index

10 = Currency

11 = Mortgage

## Security Short Description (SECURITY\_SHORT\_DES)

Alternate Short Description for a given security comprised of the ticker, coupon and maturity year (YY). For strips it returns the ticker, coupon, and maturity (M/YY). For corporate securities with Japanese tickers, the series will also be displayed.

# Technical Design

## API based Work Flow

Rest API

**BBG OpenFigi API**

Request

Response

Lookup

Response

Not Found

Store Local

Request

**Local FIGI repository**

## FTP based Work Flow

## Technologies

### Server Side

* Real time service based on Rest based APIs
* Node.js can be used on Server Side
* Jersy for Rest API server
* Grizzly for making standalone application

### Local database repository

* Oracle
* Any Open source RDBMS like MySQL
* NoSQL like MongoDB (Document based)
* HDFS (HADOOP)
* SQLite (Lightweight database)

### User Interface

* HTML\HTML5
* CSS\CSS3
* AJAX
* JQuery
* Angular.js

**Environment**

* Linux
* Windows
* AWS

# Database

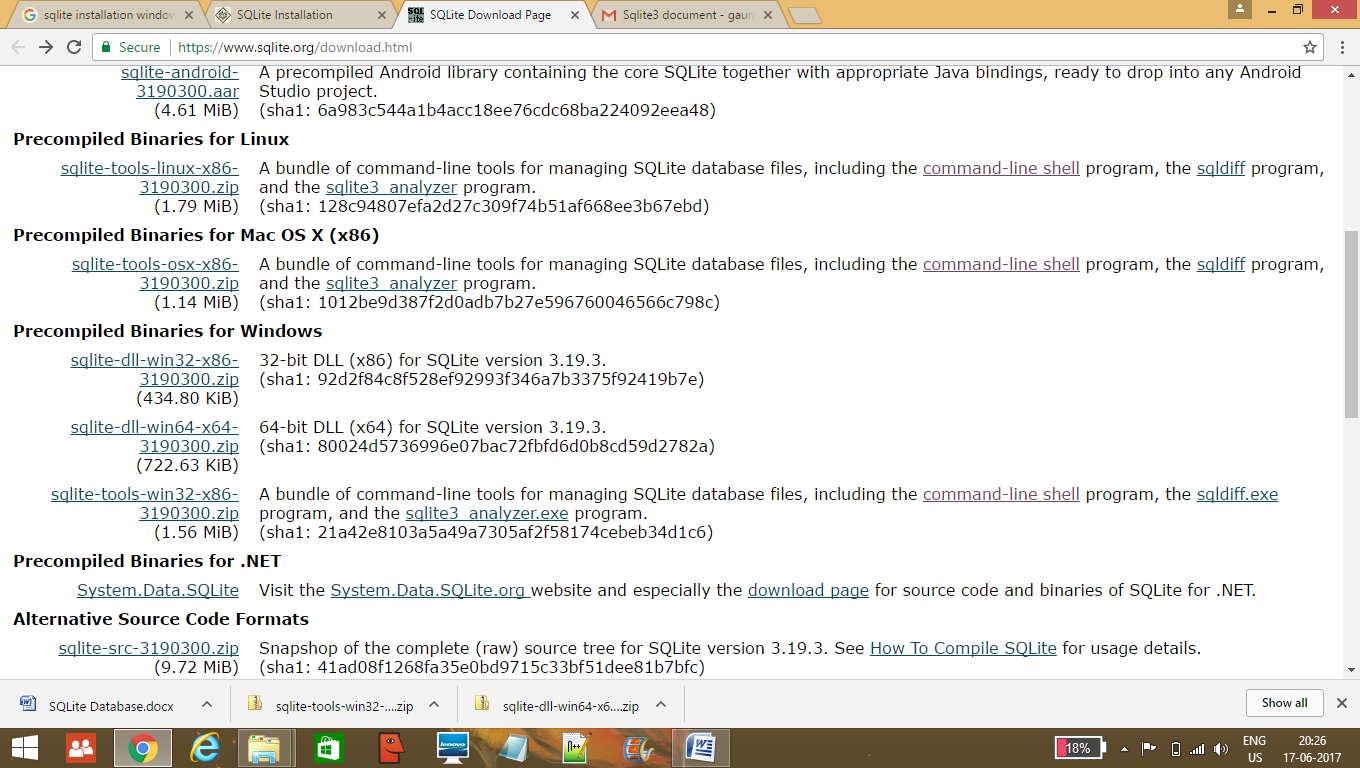
## SQLite

SQLite is a relational self-contained, high-reliability, embedded, full-featured, public-domain, SQL database engine. Some of the important features of SQLite are:-

* Self-contained - Embedded
* Serverless - Doesn't have separate server
* File based database - Reads and write directly into disk
* Almost zero-configuration
* Transactional based
* Open source
* Mode widely deployed

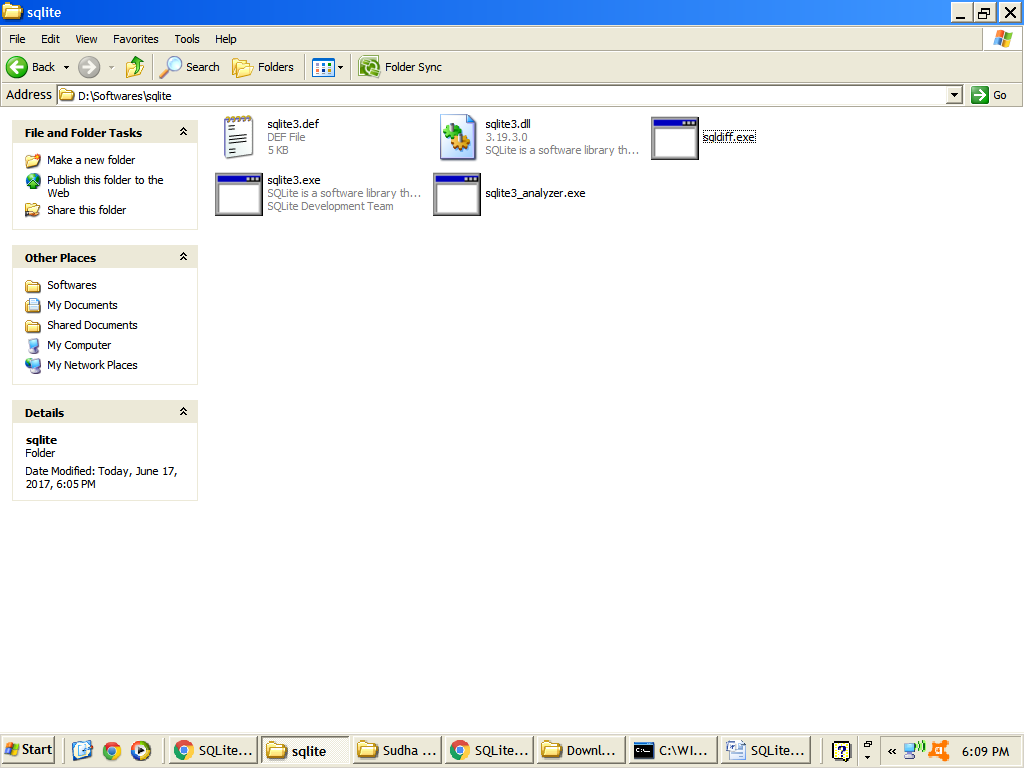
## SQLite setup on Windows

**Step1** : Go to official website of SQLite => <https://www.sqlite.org/download.html> and download sqlite-dll-win32-x86-3190300.zip (for 32 bit), sqlite-dll-win64-x64-3190300.zip (for 64bit) and sqlite-tools-win32-x86-3190300.zip from above site.

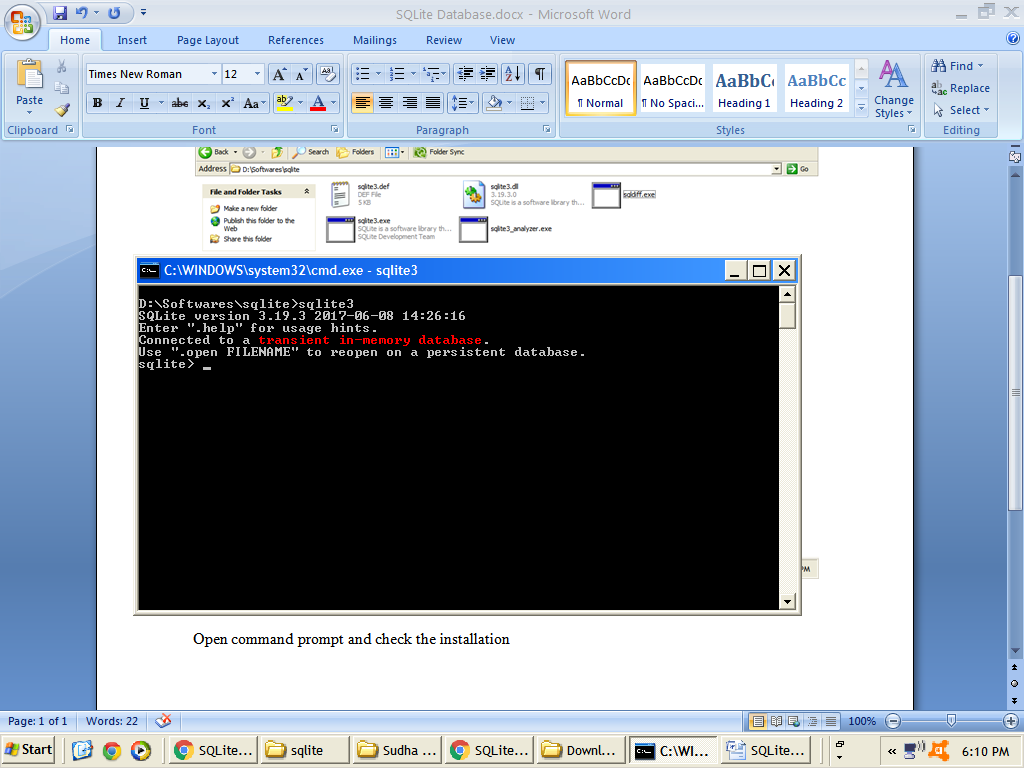


**Step2**

Create folder **C:\>sqlite** and unzip above two zipped files in this folder, which will give us sqlite3.def, sqlite3.dll and sqlite3.exe etc files as shown below:-



**Step3:** Open command prompt and check for the installation



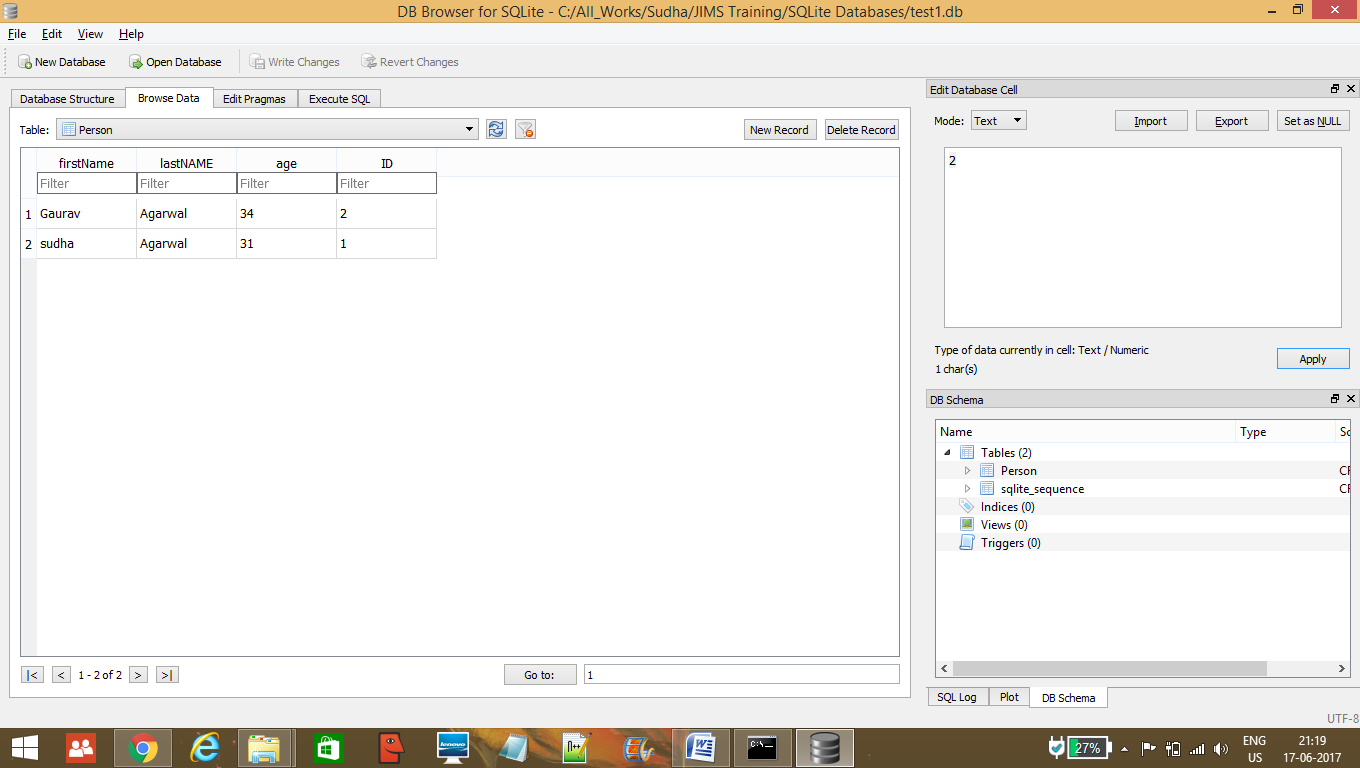
**Step4 (Optional):**

Add above folder location (**C:\>sqlite** ) to PATH environment variable.

## Browser for SQLite

DB Browser for SQLite is a high quality, visual, open source tool to create, design, and edit database files compatible with SQLite.

It can be used to create databases, search, and edit data. It uses a familiar spreadsheet-like interface, and we don't need to learn complicated SQL commands.



This can be downloaded from below location

https://github.com/sqlitebrowser/sqlitebrowser/releases/tag/v3.9.1



# Important Links

https://www.openfigi.com/about/symbology

https://www.openfigi.com/api

<https://github.com/OpenFIGI/api-examples>

<http://www.oracle.com/webfolder/technetwork/tutorials/obe/java/griz_jersey_intro/Grizzly-Jersey-Intro.html#section1s2>

https://www.sqlite.org/index.html

https://github.com/sqlitebrowser/sqlitebrowser/releases/tag/v3.9.1