Dmytro KRASNOSHAPKA
Senior Lecturer at the Department of Computer Technologies, Oles Honchar Dnipro National University, 72 Gagarin ave., Dnipro, Ukraine, postal code 49010 (krasnoshapka_d@fpm.dnu.edu.ua)

ORCID: 0000-0001-8412-960x

Kostantyn ZOLOTKO
Candidate of Technical Sciences, Associate Professor, Associate Professor at the Department of Computer Technologies, Oles Honchar Dnipro National University, 72 Gagarin ave., Dnipro, Ukraine, postal code 49010 (zolt66@gmail.com)

ORCID: 0000-0001-8444-8782

Дмитро КРАСНОШАПКА
старший викладач кафедри комп'ютерних технологій, Дніпровський національний університет імені Олеся Гончара, просп. Гагаріна, 72, Дніпро, Україна, індекс 49010 (krasnoshapka_d@fpm.dnu.edu.ua)

Костянтин ЗОЛОТЬКО
кандидат технічних наук, доцент, доцент кафедри комп'ютерних технологій, Дніпровський національний університет імені Олеся Гончара, просп. Гагаріна, 72, Дніпро, Україна, індекс 49010 (zolt66@gmail.com)

SIMPLE RELATIONAL DATABASE TEXT FORMAT FOR DEVELOPING, DESCRIBING AND EXCHANGING OVER A NETWORK

There is no simple enough text format to describe together the data and the schema of relational database. A comma-separated values (CSV) files are used to represent tabular data. But CSV not describe a relational database schema thus it cannot represent database in complete. New text format for describing database can be CSV files with a simple name convention for naming directories, files, and table columns. The purpose of this work – to give the basic definitions of this format so that on the one hand it was quite simple, and on the other quite complete.

The main idea of this format is to store the data itself, as well as the database schema, together in CSV files. Specifically, information about the database schema, such as the column name, its type, whether it is a key, etc., is encoded in the column name CSV tables. Also, the name of the directory where the CSV files are located is the name of the database, and the names of the CSV files themselves are the names of the database tables.

The article considers an example that contains the main features of a relational database – tabular representation, primary key, foreign key. It is shown that this database is completely described by the new format.

Converters are required to use the new format to exchange databases between different databases. Conventional tools for working with CSV files, which are supported by many databases, are not able to convert the database in new format to another format. To verify the correctness of this format, a simple converter was implemented to convert databases from this format to the Maria DB format and vice versa. As a result of testing the converter it was possible to show that this format is quite suitable for describing together the data and the scheme of the relational database.

These results indicate that this format is quite suitable for creating simple databases, using this format as an intermediate for the transfer of databases between different types of ones and so on.

***Key words:** relational databases, CSV format, file formats, database converter, data transfer over the network.*

ПРОСТИЙ ТЕКСТОВИЙ ФОРМАТ РЕЛЯЦІЙНОЇ БАЗИ ДАНИХ ДЛЯ РОЗРОБКИ, ОПИСУ ТА ОБМІНУ ДАНИМИ ЧЕРЕЗ МЕРЕЖУ

Не існує достатньо простого текстового формату для спільного опису даних і схеми реляційної бази даних. Для представлення табличних даних використовуються файли зі значеннями, розділеними комами (CSV). Але CSV не описує схему реляційної бази даних, тому він не може представляти базу даних повністю. Новим текстовим форматом для опису бази даних можуть бути файли CSV, які використовують просту угоду про іменування каталогів,

файлів і стовпців таблиці. Мета даної роботи – дати основні визначення цього формату так, щоб з одного боку він був досить простим, а з іншого досить повним.

Основна ідея цього формату полягає у зберіганні самих даних, а також схеми бази даних разом у файлах CSV. Зокрема, інформація про схему бази даних, як-от ім'я стовпця, його тип, чи є він ключем тощо, кодується в таблиці CSV із назвою стовпця. Крім того, назва каталогу, де знаходяться файли CSV, є назвою бази даних, а імена самих файлів CSV є назвами таблиць бази даних.

У статті розглянуто приклад, який містить основні ознаки реляційної бази даних – табличне представлення, первинний ключ, зовнішній ключ. Показано, що ця база даних повністю описується новим форматом.

Конвертери повинні використовувати новий формат для обміну базами даних між різними базами даних. Звичайні інструменти для роботи з файлами CSV, які підтримуються багатьма базами даних, не здатні конвертувати базу даних нового формату в інший формат. Щоб перевірити правильність цього формату, було реалізовано простий конвертер для перетворення баз даних із цього формату у формат Maria DB і навпаки. В результаті тестування конвертера вдалося показати, що цей формат цілком підходить для опису разом даних і схеми реляційної бази даних.

Ці результати свідчать про те, що цей формат цілком підходить для створення простих баз даних, використання цього формату як проміжного для передачі баз даних між різними типами баз даних, передачі баз даних через комп'ютерні мережі тощо.

***Ключові слова**: реляційні бази даних, формат CSV, формати файлів, конвертер бази даних, передача даних по мережі.*

Introduction. A comma-separated values (CSV) files are used to represent tabular data. Due to its simplicity and wide popularity, it is widely used in the data exchange between different computer programs, especially between spreadsheets and database management systems. The CSV files standard is represented in RFC 4180 and RFC 7111[1; 2]. There are also many formats for data exchange, for example, in computer networks such as XML, GML, IFC, CityGML, Json, Binary Based Formats and others [3; 4, 5, 6].

But can CSV fully describe a relational database? There are solutions, such as the Tabular Data Package [7], that use CSV to describe tabular-style data and contain not only data but also a database schema.

Tabular Data Package stores data in CSV files and separately stores the schema in a single JSON file.

Our solution offers not to separate data storage and schemas, but to store everything in CSV files.

The main idea of our solution is a database schema and the data itself should not be separated from each other. The directory containing the CSV files is suggested to be named after the database, the CSV files are suggested to be named after the corresponding tables, and the names of the columns in the tables are suggested to contain information about the data type, primary key, and so on.

We propose to name this solution CSVDB – comma-separated values database to call it somehow, as it is based on the CSV format and it describes a relational database.

Realization. The file structure of the CSVDB database is as follows. Database table files are located in a directory whose name is the name of the database. The file names in the directory are the names of the corresponding tables (see Fig. 2).

In the tables themselves, information about the database schema is contained in the column names. Different parts of the scheme information are separated from each other by an underscore "_". The order of the parts of the scheme information in the column name is as follows:

column name_data type_other parameters_primary key_foreign key

Table 1 presents the rules for forming column names, the standard SQL data types used [8].

Table 1

Rules for determining parts of the column name

Part of the column name	Definition
1	2
Column name	Sequence of alphanumeric characters, the first character must be a letter. If the name contains an underscore "_", then the name uses quotation marks, for example, id officeAddress1 "book_id"
Data types, for example, varchar(50) integer int decimal	Corresponds to SQL data types, for simplification brackets in designation of length of type are removed, for example, varchar50 integer int decimal
Not null	notNull
Null	null

Continuation of table 1

1	2
Auto-increment	autoIncrement
Primery key	pk
Foreign key	fk_referenced table name_ referenced field name

Examples of table column names:

id_integer_pk – column has name id, type – integer and is primary key.

postId_int_notNull_fk_Users_userId – column has name postId, type – integer, IS NOT NULL condition, foreign key, referenced table name – Users, referenced field name – userId.

CSVDB Example. For example, consider a simple database of books and book authors - Books. The scheme of the database is presented in the following diagram – Fig. 1:
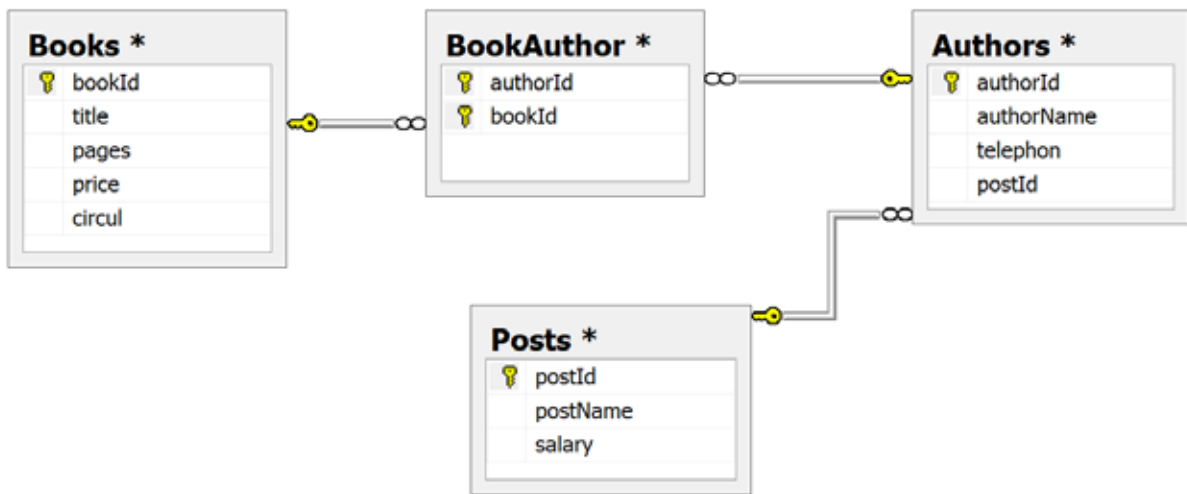

Fig. 1. Scheme of the Books database

This database in CSVDB is represented as follows.

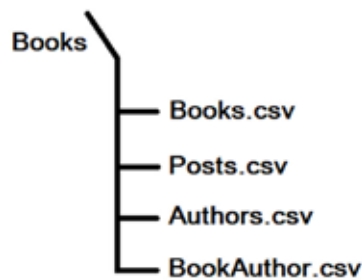The location of the database files on the disk is shown in Fig. 2:

Fig. 2. Allocation the Books database on disk

CSV files with data.

File Books.csv:

bookId_int_notNull_autoincrement_pk,title_varchar100_notNull,pages_int_notNull,price_decimal_null,circul_int_null

1,Now or never,512,16,1024

2,"Don't call us, we'll call you",1024,32,2048

3,Magnum opus,2048,64,4096

Books.csv file data is presented without commas as a table (Table 2) for convenience.

Table 2

Books

bookId_ int_ notNull_ autoIncrement_ pk,	title_ varchar100_ notNull,	pages_ int_ notNull,	price_ decimal_ null,	circul_ int_ null
1,	Now or never,	512,	16,	1024
2,	"Don't call us, we'll call you",	1024,	32,	2048
3,	Magnum opus,	2048,	64,	4096

Notice the phrase "Don't call us, we'll call you". If the data contains a comma, they should be enclosed in quotation marks.

File Posts.csv:

postId_int_notNull_autoincrement_pk,postName_varchar100_notNull,salary_decimal_null
1,Lecturer,8000
2,Professor,9000
3,Assistant,9000

Table 3

Posts

postId_ int_ notNull_ autoIncrement_ pk,	postName_ varchar100_ nutNull,	salary_ decimal_ null
1,	Lecturer,	8000
2,	Professor,	9000
3,	Assistant,	9000

File Authors.csv:

authorId_int_notNull_autoincrement_pk,authorName_varchar100_notNull,telephon_varchar100_null,postId_int_notNull_fk_Posts_postId
1,Blue A,038-450–485,1
2,Green B,038-500–565,3
3,Yellow C,038-565–590,2
4,Orange D,038-590–625,3

Table 4

Authors

authorId_ int_ notNull_ autoIncrement_ pk,	authorName_ varchar100_ notNull,	telephon_ varchar100_ null,	postId_ int_ notNull_ fk_ Posts_ postId
1,	Blue A,	038-450–485,	1
2,	Green B,	038-500–565,	3
3,	Yellow C,	038-565–590,	2
4,	Orange D,	038-590–625,	3

File BookAuthor.csv:

authorId_int_notNull_pk_fk_Authors_authorId,bookId_int_notNull_pk_fk_Books_bookId
1,1
2,1
3,2
3,3
4,2
4,3

Table 5

BookAuthor

authorId_ int_ notNull_ pk_ fk_ Authors_ authorId,	bookId_ int_ notNull_ pk_ fk_ Books_ bookId
1,	1
2,	1
3,	2
3,	3
4,	2
4,	3

The example of the database in CSVDB format is present on gitlab.com [9].

CSVDB converters. Converters are required to use the CSVDB format to exchange databases between different databases. Conventional tools for working with CSV files, which are supported by many databases [10; 11], are not able to convert the database in CSVDB format to another format.

CSVDB converter must convert a database from CSVDB format to a number of other database formats and vice versa.

Introducing a simple converter to convert databases from CSVDB to Maria DB and vice versa [12].

This converter is an executable JAR file and requires a JRE or JDK version 8 or later to work.

For example, to start the converter in Windows, you can call the context menu on the converter file, select "Open with" and select "Java (TM) Platform SE binary" – OK.

To test the converter, we used MariaDB Server, which is included in the packages XAMPP 7.2.11 (https://apachefriends.org/index.html).

Of course, it is desirable to have converters of different types of databases, and even better - conversion tools built into the database for comfortable work with the CSVDB format.

Summary and conclusions. In this paper, we present a simple text format for relational databases, which is based on CSV – CSVDB. The main idea of this format is to store the data itself, as well as the database schema, together in CSV files. Specifically, information about the database schema, such as the column name, its type, whether it is a key, etc., is encoded in the column name CSV tables. Also, the name of the directory where the CSV files are located is the name of the database, and the names of the CSV files themselves are the names of the database tables.

This name convention makes it easy to create database projects in a plain text editor and then import them into any database for which there is a suitable converter.
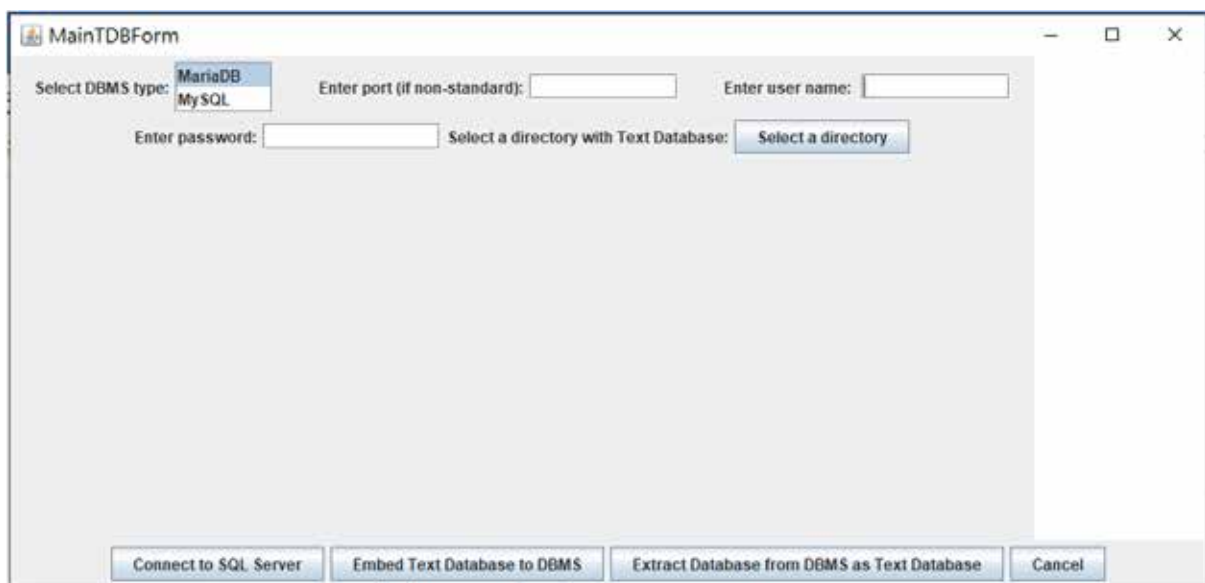

Fig. 3. CSVDB converter user interface

The advantages of this format include:

1. Simplicity of the format, as it is based on CSV.

2. This format can be easily viewed and edited in a text editor.

3. Clarity, ease of understanding, as all information about the database schema is in front of the eyes in the column name.

Cross-platform, as a database from CSVDB format can be converted to any database format for which there is a converter. You can also use this format as an intermediate for the transfer of databases between different types of ones.

Further development of this format is possible in the following areas:

- discussion, improvement, standardization
- creation of converters for various database formats
- creating tools to improve the usability of this format in text editors, such as macros to check data types in tables, check the primary key constraint, etc.

Obviously, this solution has its drawbacks and does not take into account all the features of the database scheme, so we invite all stakeholders to provide their comments and suggestions for improvement.

Bibliography:

1. Y. Shafranovich, Common Format and MIME Type for Comma-Separated Values (CSV) Files. *RFC 4180*, October 2005, DOI 10.17487/RFC4180.

2. M. Hausenblas, E. Wilde, J. Tennison. URI Fragment Identifiers for the text/csv Media Type, *RFC 7111*, October 2014, DOI 10.17487/RFC7111.

3. Xu Feng et al. Research on Cross-network Exchange Method of Enterprise Application Business Process. *Data J. Phys.: Conf. Ser*. 2020. №1693 012037.

4. Greg Charest, Mitch Rogers. Data Exchange Mechanisms and Considerations. Enterprise Architecture (Harvard Unsversity): веб-сайт. URL: https://enterprisearchitecture.harvard.edu/data-exchange-mechanisms (дата звернення: 20.11.2022).

5. Mohammed Jawaluddeen San, Ivin Amri Musliman, Alias Abdul Rahman. IFC to CityGML Conversion Algorithm Based on Geometry and Semantic Mapping. Joint International Conference Geospatial Asia-Europe 2021 and GeoAdvances, 5–6 October 2021, online.

6. Bodlaj, J. (2014). Network Data File Formats. In: Alhajj, R., Rokne, J. (eds) Encyclopedia of Social Network Analysis and Mining. Springer, New York, NY. https://doi.org/10.1007/978-1-4614-6170-8_298.

7. Paul Walsh, Rufus Pollock, Martin Keegan, "Tabular Data Package". [Online]. веб-сайт. URL: https://specs.frictionlessdata.io/tabular-data-package/#language (дата звернення: 20.11.2022).

8. ANSI/ISO/IEC International Standard (IS) Database Language SQL — Part 2: Foundation (SQL/Foundation): веб-сайт. URL: http://web.cecs.pdx.edu/~len/sql1999.pdf (дата звернення: 20.11.2022).

9. GitLab: веб-сайт. URL: https://gitlab.com/dimakrasnoshapka/books.git (дата звернення: 20.11.2022).

10. Understanding CSV Files in Excel. веб-сайт. URL: https://www.spreadsheetsmadeeasy.com/understanding-csv-files-in-excel/ (дата звернення: 20.11.2022).

11. WP Data Access. CSV files: веб-сайт. URL: https://wpdataaccess.com/docs/premium-data-services/csv-files/# (дата звернення: 20.11.2022).

12. Google Drive. CSV files: веб-сайт. URL: https://drive.google.com/file/d/1y-guvxD0r74cJ_bi8yy6J_rHli1N7B6M/view?usp=sharing (дата звернення: 20.11.2022).

References:

1. [RFC4180] Y. Shafranovich, "Common Format and MIME Type for Comma-Separated Values (CSV) Files", RFC 4180, DOI 10.17487/RFC4180, October 2005, <https://www.rfc-editor.org/info/rfc4180>.

2. [RFC 7111] M. Hausenblas, E. Wilde, J. Tennison, "URI Fragment Identifiers for the text/csv Media Type", RFC 7111, DOI 10.17487/RFC7111, October 2014, <https://www.rfc-editor.org/info/rfc7111>.

3. Xu Feng et al (2020). Research on Cross-network Exchange Method of Enterprise Application Business Process. *J. Phys.: Conf. Ser. 1693* 012037.

4. Greg Charest, Mitch Rogers. Data Exchange Mechanisms and Considerations. Enterprise Architecture (Harvard Unsversity). *enterprisearchitecture.harvard.edu/data-exchange-mechanisms*. Retrieved from https://enterprisearchitecture.harvard.edu/data-exchange-mechanisms.

5. Mohammed Jawaluddeen San, Ivin Amri Musliman, Alias Abdul Rahman (2021). Proceedings from 'IFC to CityGML Conversion Algorithm Based on Geometry and Semantic Mapping. Joint International Conference Geospatial Asia-Europe 2021 and GeoAdvances', (pp. 356–369).

6. Bodlaj, J. (2014). Network Data File Formats. *In: Alhajj, R., Rokne, J. (eds) Encyclopedia of Social Network Analysis and Mining*. Springer, New York, NY. https://doi.org/10.1007/978-1-4614-6170-8_298.

7. Paul Walsh, Rufus Pollock, Martin Keegan, "Tabular Data Package". (n.d.) specs.frictionlessdata.io/tabular-data-package/#language . [Online]. Retrieved from https://specs.frictionlessdata.io/tabular-data-package/#language.

8. ANSI/ISO/IEC International Standard (IS) Database Language SQL – Part 2: Foundation (SQL/Foundation). (n.d.) web.cecs.pdx.edu/~len/sql1999.pdf. Retrieved from http://web.cecs.pdx.edu/~len/sql1999.pdf.

9. GitLab. (n.d.) gitlab.com/dimakrasnoshapka/books.git. Retrieved from https://gitlab.com/dimakrasnoshapka/books.git.

10. Understanding CSV Files in Excel. (n.d.) spreadsheetsmadeeasy.com/understanding-csv-files-in-excel. Retrieved from https://www.spreadsheetsmadeeasy.com/understanding-csv-files-in-excel/.

11. WP Data Access. CSV files. (n.d.) wpdataaccess.com/docs/premium-data-services/csv-files/#. Retrieved from https://wpdataaccess.com/docs/premium-data-services/csv-files/# .

12. Google Drive. (n.d.) drive.google.com/file/d/1y-guvxD0r74cJ_bi8yy6J_rHli1N7B6M/view?usp=sharing. Retrieved from https://drive.google.com/file/d/1y-guvxD0r74cJ_bi8yy6J_rHli1N7B6M/view?usp=sharing.