# Система управления базами данных DuckDB: что это, сравнение с SQLite, использование с Python

Воронкин Р.А., Шайдеров Д.В.

**Постановка задачи:** исследование СУБД DuckDB, определение преимуществ и недостатков, сферы применения, возможностей взаимодействия с языком программирования Python. **Цель работы:** исследовать СУБД DuckDB и определить способы взаимодействия с языком программирования Python. **Используемые методы:** метод анализа, метод сравнения. **Результат:** описание СУБД DuckDB, пример реализации использования базы данных DuckDB на языке программирования Python. **Практическая значимость:** DuckDB является наиболее оптимальным выбором среди СУБД для анализа данных.

**Ключевые слова:** DuckDB, СУБД, SQL, SQLite, OLAP, запросы, база данных, таблицы.

## Общая характеристика

DuckDB – это встроенная реляционная система управления базами данных OLAP, поддерживающая SQL. Что это значит:

В отличие от большинства СУБД, DuckDB является не самостоятельным приложением, взаимодействие с которым организовано по принципу клиент-сервер, а подключаемой к программе библиотекой.

Реляционная БД означает, что, благодаря так называемым первичным и внешним ключам, в ней осуществимо установление связей между таблицами.

OLAP (Оперативная аналитическая обработка) – это технология, поддерживающая сложный анализ коммерческих баз данных. Базы данных OLAP предназначены для нагрузок, преимущественно состоящих из операций чтения, с малым числом операций записи, что позволяет оптимизировать извлечение сведений бизнес-аналитики.

## Сравнительная характеристика с SQLite.

DuckDB называют «SQLite для аналитики». В чем же разница?

1. В первую очередь, SQLite поддерживает всего пять типов данных: NULL, INTEGER, REAL, TEXT, BLOB. Это означает, что в некоторых случаях при выполнении запросов придется прибегать к преобразованию типов. DuckDB, в свою очередь, поддерживает более 20 типов данных, таких как, например, DOUBLE, REAL, BOOLEAN, DATE, TIME, VARCHAR, а также вложенные типы LIST, MAP, STRUCT, UNION.

2. DuckDB поддерживает большее количество SQL-выражений, таких как EXCLUDE, REPLACE, GROUP BY ALL.

3. DuckDB позволяет автоматически загружать csv- и json-файлы благодаря функциям read_csv_auto и read_json_auto соответственно.

4. Однако ключевым различием между этими СУБД является способ выполнения запросов. SQLite последовательно обрабатывает каждую строку, в то время как DuckDB использует векторизацию выполнения запросов (ориентацию на столбцы), что значительно увеличивает производительность.

## Преимущества и недостатки DuckDB

Разобравшись с тем, что из себя представляет DuckDB, можно суммировать сильные и слабые стороны СУБД:

**Преимущества**

- Простота – черта, которую система DuckDB унаследовала от SQLite. Подразумевает легкость установки и использования – за счет встроенности в процесс.
- Поддержка рабочих нагрузок аналитических запросов (OLAP). Такие запросы характеризуются длительностью и обрабатывают весомую часть набора данных. Как отмечалось выше, DuckDB использует векторизованные механизмы выполнения запросов, что снижает накладные расходы.
- DuckDB глубоко интегрирован в такие языки программирования, как Python и R, а также предоставляет API для C, C++, Java и других.

- Доступность – DuckDB находится в публичном доступе.

**Недостатки**

- Встроенность и отсутствие сервера являются одновременно как достоинством, так и недостатком. Так, к одной базе данных не могут обращаться несколько разных устройств.

## Использование DuckDB

**Когда стоит использовать DuckDB?**

- При обработке и хранении табличных данных.
- Для интерактивного анализа данных (например, объединение больших таблиц).
- Сложные аналитические запросы (получение большого количества результатов, одновременные изменения в нескольких таблицах).

**Когда не стоит использовать DuckDB?**

- Несколько параллельных процессов, которые ведут запись в одну базу данных или чтение из нее.
- Использование централизованных клиент-серверных установок.

## Возможности DuckDB

Продемонстрируем основные возможные запросы на примере датасета (https://www.kaggle.com/datasets/jealousleopard/goodreadsbooks):

**Импорт данных**

Создадим таблицу при помощи выборки из файла books.csv, для загрузки которого используем вышеупомянутую функцию read_csv_auto. Создание таблицы осуществляется посредством запроса CREATE TABLE.

```
CREATE TABLE books AS (SELECT * FROM read_csv_auto('books.csv') LIMIT 10);
```

Рисунок 1 - Выполненный запрос на импорт данных

**Выполнение запроса select**

Выведем на экран выборку из 10 строк из файла books.csv, для этого используем запрос SELECT.

```
D SELECT * FROM read_csv_auto('books.csv') LIMIT 10;
```

| bookID int64 | title varchar | authors varchar | : | text_reviews_count int64 | publication_date varchar | publisher varchar |
|---|---|---|---|---|---|---|
| 1 | Harry Potter and t: | J.K. Rowling/Mary : | : | 27591 | 9/16/2006 | Scholastic Inc. |
| 2 | Harry Potter and t: | J.K. Rowling/Mary : | : | 29221 | 9/1/2004 | Scholastic Inc. |
| 4 | Harry Potter and t: | J.K. Rowling | : | 244 | 11/1/2003 | Scholastic |
| 5 | Harry Potter and t: | J.K. Rowling/Mary : | : | 36325 | 5/1/2004 | Scholastic Inc. |
| 8 | Harry Potter Boxed: | J.K. Rowling/Mary : | : | 164 | 9/13/2004 | Scholastic |
| 9 | Unauthorized Harry: | W. Frederick Zimme: | : | 1 | 4/26/2005 | Nimble Books |
| 10 | Harry Potter Colle: | J.K. Rowling | : | 808 | 9/12/2005 | Scholastic |
| 12 | The Ultimate Hitch: | Douglas Adams | : | 254 | 11/1/2005 | Gramercy Books |
| 13 | The Ultimate Hitch: | Douglas Adams | : | 4080 | 4/30/2002 | Del Rey Books |
| 14 | The Hitchhiker's G: | Douglas Adams | : | 460 | 8/3/2004 | Crown |

10 rows 12 columns (6 shown)

Рисунок 2 - Выполненный запрос на составление выборки

**Экспорт данных (в формат CSV)**

Экспортируем выборку в файл new_books.csv. Для этого используем функцию COPY(), который передадим запрос SELECT.

```
COPY(SELECT * FROM read_csv_auto('books.csv') LIMIT 10) TO 'new_books.csv' (DELIMITER '|', HEADER);
```

Рисунок 3 - Выполненный запрос на экспорт данных

## Использование DuckDB с Python

Установка DuckDB в виртуальную среду:

```
pip install duckdb
```

Создание базы данных и таблиц

```
import duckdb

conn = duckdb.connect(str(database_path))
cursor = conn.cursor()
cursor.execute(
    """
    CREATE TABLE IF NOT EXISTS types (
        type_id INTEGER PRIMARY KEY,
        type_title TEXT NOT NULL
    )
    """
```

```
    )

# Создать таблицу с информацией о самолетах.
cursor.execute(
    """
    CREATE TABLE IF NOT EXISTS planes (
        plane_id INTEGER PRIMARY KEY,
        plane_destination TEXT NOT NULL,
        type_id INTEGER NOT NULL,
        plane_num INTEGER NOT NULL,
        FOREIGN KEY(type_id) REFERENCES types(type_id)
    )
    """
)

conn.close()
```

Создание последовательностей:

```
cursor.execute(
    """
    CREATE SEQUENCE IF NOT EXISTS type_st START 1
    """
)
cursor.execute(
    """
    CREATE SEQUENCE IF NOT EXISTS plane_st START 1
    """
)
```

Выполнение запроса SELECT:

```
cursor.execute(
    """
    SELECT planes.plane_destination, types.type_title, planes.plane_num
    FROM planes
    INNER JOIN types ON types.type_id = planes.type_id
    WHERE types.type_title = ?
    """,
    (jet,)
)
```

```
+------+---------------------------------+----------------------+------------------+
|  No  |        Пункт назначения         |     Номер рейса      |  Тип самолета    |
+------+---------------------------------+----------------------+------------------+
|    1 | Москва                          | 123                  |         Грузовой |
+------+---------------------------------+----------------------+------------------+
```

Рисунок 4 - Выполненный запрос на составление выборки внутри программы на языке программирования Python

Выполнение запроса INSERT:

```
cursor.execute(
    """
    INSERT INTO planes
    VALUES (nextval('plane_st'), ?, ?, ?)
    """,
    (destination, type_id, num)
)
```

Пример программы Python с интерфейсом командной строки, использующей базу данных DuckDB: https://github.com/dshay-derov/DuckDB_2/blob/main/Project/ind.py

**Вывод:** DuckDB является системой управления базами данных, унаследовавшей все сильные стороны SQLite – встроенность в процесс и

простоту использования, при этом решив её главную проблему – производительность. Таким образом, DuckDB является оптимальным выбором среди СУБД для анализа данных.

## Список используемой литературы


1. Документация DuckDB. URL: https://duckdb.org/docs/
2. Статья на тему: «Почему DuckDB?». URL: https://duckdb.org/why_duckdb.html
3. Статья на тему: «Forget about SQLite, Use DuckDB Instead — And Thank Me Later». URL: https://towardsdatascience.com/forget-about-sqlite-use-duckdb-instead-and-thank-me-later-df76ee9bb777
4. Оперативная аналитическая обработка. URL: https://learn.microsoft.com/ru-ru/azure/architecture/data-guide/relational-data/online-analytical-processing
5. Статья на тему: «DUCKDB: INTRODUCING A NEW CLASS OF DATA MANAGEMENT SYSTEMS». URL: https://ict-research.nl/wordpress/wp-content/uploads/2023/04/IO-magazine-NR1-2023.pdf#page=10