

## **Оглавление**

Часть 1. Краткое описание OrientDB .....	3
Часть 2. Описание модели данных, системы типов .....	4
Часть 3. Функциональные возможности .....	6
Часть 4. Языки запросов, DDL и DML сценарии .....	6
Список используемой литературы .....	8

## Часть 1. Краткое описание OrientDB

OrientDB - это система управления базами данных NoSQL с открытым исходным кодом, написанная на Java. Это мультимодельная база данных, поддерживающая графическую, документальную, ключ/значение и объектную модели. Он поддерживает режимы без схемы, с полной схемой и со смешанной схемой. Он имеет систему профилирования безопасности, основанную на пользователях и ролях, и поддерживает запросы с помощью Gremlin наряду с расширенным SQL для обхода графа [1].

OrientDB является мультимодельной базой данных. Мультимодельная база данных — это база данных, предназначенная для поддержки нескольких моделей данных в одной системе хранения данных. Это означает, что такая система может хранить, индексировать и запрашивать данные в нескольких моделях.

Среди особенностей можно выделить следующие [2]:

1. Полная поддержка ACID-транзакций;
2. Поддержка подмножества языка SQL для выполнения запросов с использованием конструкции SELECT (OrientDB не является реляционной БД, поэтому в полной мере все возможности SQL не поддерживает);
3. Поддержка хранения данных без описания предварительной схемы, с описанием полной структуры или в смешанном режиме;
4. Полностью совместима со стандартом TinkerPop Blueprints для графо-ориентированных БД;
5. Нативно поддерживает HTTP, RESTful и JSON протоколы без использования сторонних компонентов;
6. Поддерживается строгая политика разграничения доступа на основе ролей и полномочий пользователей;
7. OrientDB поддерживает графовую, документную модели данных.

## **Часть 2. Описание модели данных, системы типов**

В данной СУБД поддерживаются 2 основные модели баз данных: документные и графовые [3].

Графовая модель базы данных — это модель, в которой структуры данных для схемы и/или экземпляров моделируются как направленный, возможно, помеченный граф или обобщение структуры данных графа, где манипулирование данными выражается с помощью графо-ориентированных операций и конструкторов типов, а соответствующие ограничения целостности могут быть определены в структуре графа [4].

Для документной модели вместо хранения данных в фиксированных строках и столбцах базы данных документов используют гибкие документы. Документ — это запись в базе данных документов. Документ обычно хранит информацию об одном объекте и любых связанных с ним метаданных. Документы хранят данные в парах поле-значение. Значения могут быть различных типов и структур, включая строки, числа, даты, массивы или объекты [5].

OrientDB поддерживает следующие типы данных [6]:

1. Boolean Булевый тип, который поддерживает только 2 значения True или False
2. String. Тип строки
3. Integer. 32-разрядные целые числа
4. Short. Маленькие 16-битные числа со знаком
5. Long. Большие 64-битные числа со знаком
6. Float. Десятичные числа
7. Double. Десятичные числа с большой точностью
8. Datetime. Любая дата с точностью до миллисекунд
9. Binary. Может содержать любое значение в виде массива байтов
10. Embedded. Запись содержится внутри владельца. Содержащаяся

запись не имеет идентификатора записи

11. Embedded list. Записи содержатся внутри владельца. Содержащиеся записи не имеют идентификаторов записи и доступны только путем навигации по записи владельца
12. Embedded set. Записи содержатся внутри владельца. Содержащиеся записи не имеют идентификатора записи и доступны только путем навигации по записи владельца
13. Embedded map. Записи содержатся внутри владельца в виде значений записей, в то время как ключами могут быть только строки. Содержащиеся в них порядки не имеют идентификаторов записей и доступны только путем навигации по записи владельца
14. Link. Ссылка на другую запись. Это обычная связь "один к одному"
15. Link list. Ссылки на другие записи. Это обычная связь "один ко многим", где хранятся только идентификаторы записей
16. Link set. Ссылки на другие записи. Это обычное отношение "один ко многим"
17. Link map. Ссылки на другие записи в качестве значений записей, в то время как ключами могут быть только строки. Это обычное отношение "Один ко многим". Сохраняются только идентификаторы записей
18. Byte. Однобайтовый. Полезно для хранения небольших 8-разрядных целых чисел со знаком.
19. Transient. Любое значение, не сохраненное в базе данных
20. Date. Любая дата в виде года, месяца и дня. Чтобы узнать больше об этом, ознакомьтесь с Управление датами
21. Custom. Используется для хранения пользовательского типа, предоставляющего методы marshall и unmarshall
22. Decima. Десятичные числа без округления

23. LinkBag. Список идентификаторов записей в качестве спецификации RidBag

### **Часть 3. Функциональные возможности**

Функциональные возможности orientDB следующие [7]:

1. OrientDB поддерживает минимальный движок SQL, позволяющий выполнять пакет команд (batch). Пакетные команды очень полезны, когда вам нужно выполнить несколько действий на стороне сервера, избегая обратного подключения к сети для каждой команды. Batch поддерживает все SQL-команды OrientDB.
2. OrientDB использует кластеры только для хранения ссылок на данные. Кластер - это общий способ группировки записей. OrientDB по умолчанию использует кластер для каждого "класса". Кластер может быть локальным (физическим) или размещенным в памяти.
3. В графовой модели OrientDB база данных представляет данные с помощью концепции графа свойств, который определяет вершину как объект, связанный с другими вершинами, и ребро как объект, связывающий две вершины.
4. OrientDB может обрабатывать индексы таким же образом, как и классы, используя язык SQL и добавляя к имени префикс "index:", за которым следует имя индекса. Индекс подобен классу с двумя свойствами: key - индексный ключ, rid - идентификатор записи, который указывает на запись, связанную с ключом.

### **Часть 4. Языки запросов, DDL и DML сценарии**

В OrientDB таблицы представляются в виде классов, строки - записи в таблице, а свойства класса - поля. В основном OrientDB использует

документальную модель, но также можно использовать графовую модель.

Команды DML сценария существуют следующие [8]:

1. Для создания классов используется следующая команда:  
CREATE CLASS <class\_name>
2. Команда ALTER CLASS используется для изменения существующего класса в OrientDB.
3. Команда DROP CLASS используется для удаления класса из базы данных. Эта операция удаляет класс и всё его содержимое из базы данных.
4. Команда CREATE PROPERTY используется для создания нового свойства внутри класса.
5. Команда CREATE DATABASE используется для создания новой базы данных
6. Команда ALTER PROPERTY используется для изменения свойства класса в схеме.
7. Команда DROP PROPERTY используется для удаления свойства из схемы.
8. Команда CREATE CLUSTER используется для создания нового кластера в базе данных. После создания кластер можно использовать для сохранения записей, указывая его имя во время сохранения.
9. Команда ALTER CLUSTER используется для изменения существующего кластера в базе данных. Она предоставляет возможность переименовать кластер, а также внести другие возможные изменения в конфигурацию кластера.
10. Команда TRUNCATE RECORD используется для удаления записей.

Команды DDL сценарий существуют следующие [9]:

1. Для добавления полей в класс, можно использовать следующую

команду:

INSERT INTO <class\_name> (<field\_name>) VALUES (field\_value)

2. Для удаления записей используется команда DELETE:  
DELETE FROM <target-name>
3. Для изменения значения используется UPDATE:  
UPDATE Profile SET nick = 'Luca'
4. Для выбора результатов используется SELECT:  
SELECT name, age FROM Account

### **Список используемой литературы**

1. [Электронный ресурс] – 2020 г. – Режим доступа: <https://en.wikipedia.org/wiki/OrientDB>, свободный.
2. [Электронный ресурс] – 2015 г. – Режим доступа: <https://www.opennet.ru/opennews/art.shtml?num=41504>, свободный.
3. [Электронный ресурс] – 2014 г. – Режим доступа: <https://habr.com/ru/articles/214837/>, свободный.
4. Survey of graph database models / Renzo Angles, Claudio Gutierrez; — ACM Computing Surveys, vol. 40, 2008. — 1–39 с.
5. [Электронный ресурс] – 2023 г. – Режим доступа: <https://www.mongodb.com/document-databases>, свободный.
6. [Электронный ресурс] – 2024 г. – Режим доступа: <https://orientdb.com/docs/latest/general/Types.html>, свободный.
7. [Электронный ресурс] – 2022 г. – Режим доступа: <https://orientdb.org/docs/3.0.x/>, свободный.
8. [Электронный ресурс] – 2023 г. – Режим доступа: <https://orientdb.org/docs/3.0.x/sql/SQL-Commands.html/>, свободный.
9. [Электронный ресурс] – 2023 г. – Режим доступа: <https://orientdb.org/docs/3.2.x/>, свободный.