**Лабораторная работа № 3**

**ЗНАКОМСТВО С ORIENT DB**

**Цель работы:** ознакомиться с основными понятиями работы с Orient DB, а также с некоторыми сведениями о данной системе.

**ОБЩИЕ СВЕДЕНИЯ**

OrientDB - это система управления NoSQL базами данных с открытым исходным кодом, написанная на языке Java. Это мультимодельная база данных, поддерживающая графики, документы, ключи/значения и объектные модели, но отношения в ней регулируются также как и в графовых базах данных с прямыми связями между записями. Разработка OrientDB по-прежнему зависит от сообщества с открытым исходным кодом во главе с компанией OrientDB LTD, созданной ее автором Лука Гарулли. Проект использует GitHub для управления источниками, вкладчиками и версиями, Google Group и стек переполнения, чтобы обеспечить бесплатную поддержку пользователей во всем мире. OrientDB также предлагает бесплатный курс Udemy для тех, кто надеется узнать основы и начать работу с OrientDB.

**ИСТОРИЯ**

Главный автор Luca Garulli в 2010 году начал переписывать на Java быстрый слой базы данных Orient ОСУБД, написанный на С++. . В течение 2012-2014 годов движок был переделан Андреем Ломакиным. Он получил новое название "plocal", что означает "с локальной разбивкой страниц". Это имя подразумевает, что новый механизм хранения основан на концепции разделения файлов данных по страницам и страница рассматривается как единая атомарная единица изменения. С 2012 года проект финансируется OrientDB LTD (бывший Orient Technologies LTD), некоммерческой компанией с Лука в качестве генерального директора и основателя. В 2013 году Андрей Ломакин присоединился к компании в качестве ведущего инженера по и совладельца компании.

OrientDB была освещена СМИ и является лауреатом премии InfoWorld Bossie 2015 года.

Разработка OrientDB по-прежнему зависит от сообщества с открытым исходным кодом во главе с компанией OrientDB LTD, созданной ее автором Лука Гарулли. Проект использует GitHub для управления источниками, вкладчиками и версиями, Google Group и стек переполнения, чтобы обеспечить бесплатную поддержку пользователей во всем мире. OrientDB также предлагает бесплатный курс Udemy для тех, кто надеется узнать основы и начать работу с OrientDB.

**ДВИЖОК**

OrientDB была основана на мульти-графовой модели. OrientDB позиционируетсая, как графовая БД, но нет никаких причин, по которым хранилище типа ключ-значение не может быть использовано в ней само по себе. OrientDB обрабатывает каждую запись/документ как объект, и связь между объектами/документами осуществляется не через ссылки, а через указатели. Это приводит к быстрому поиску связанных данных по сравнению с поиском в РСУБД. Эта база данных на основе документов, но отношения управляются как в графовых базах с прямым подключением между записями. Она поддерживает схемы: less (слабоструктурированные данные), full (строго задает обязательные поля) и mixed (смешанная: обязательные + необязательные поля). Данная база данных имеет мощную систему профилирования безопасности, основанную на пользователях и ролях, и поддерживает запросы с помощью Gremlin, а также SQL, расширенный для обхода графов. OrientDB использует несколько механизмов индексации на основе B-деревьев и хэширования, планируется реализовать LCM-дерево и Фрактальное дерево на основе индексов. Каждая запись имеет Surrogate ключ, который указывает позицию записи внутри массива или списка, ссылки между записями хранятся либо в виде единичного значения записи сохраненной позиции или как B-дерево запись позиций (так называемых идентификаторов записей или @RID), которое обеспечивает быстрое прохождение (с сложностью O(1)) один-ко-многим отношения и быстрого добавления/удаления новых ссылок. OrientDB является третьей по популярности графической базой данных по рейтингу DB-Engines graph database ranking по состоянию на сентябрь 2017 года.

**ОСОБЕННОСТИ**

Среди особенностей OrientDB можно выделить:

* Полная поддержка ACID транзакций (Atomicity — атомарность,  Consistency — согласованность, Isolation — изолированность, Durability — надежность);
* Поддержка подмножества языка SQL для выполнения запросов c использованием конструкции SELECT (OrientDB не является реляционной БД, поэтому в полной мере все возможности SQL не поддерживает);
* Поддержка "Хранимых Процедур" на языках SQL и JavaScript;
* Поддержка хранения данных без описания предварительной схемы, с описанием полной структуры или в смешанном режиме;
* 100 % совместима со стандартом TinkerPop Blueprints для графо-ориентированных БД;
* Поддержка языка запросов Gremlin;
* Поддерживает HTTP, REST и JSON протоколы без использования сторонних компонентов;
* Возможность работы как в режиме встраивания в другие приложения, так и в качестве выделенного сервера;
* Возможность отката внесённых в документ локальных изменений (ODocument.undo);
* Имеет очень малый размер и не имеет сторонних зависимостей;
* Поддерживается строгая политика разграничения доступа на основе ролей и полномочий пользователей;
* Дистрибутив полностью самодостаточен;
* Дистрибутив содержит Web-Studio (инструмент разработки и администрирования);
* Поддерживает отказоустойчивые конфигурации и репликацию (архитектура OrientDB изначально рассчитана на мультимастер репликацию);
* Поддержка запуска скриптов на стороне сервера (Server Side Scripting);
* Доступна коммерческая поддержка.

**ОСНОВНЫЕ ПОНЯТИЯ**

Главная особенность OrientDB – поддержка многомодельных объектов, то есть он поддерживает различные модели, такие как Document, Graph, Key / Value и Real Object. Он содержит отдельный API для поддержки всех этих четырех моделей.[Источник 4]

***Модель документа.*** Терминология Модель документа принадлежит базе данных NoSQL. Это означает, что данные хранятся в Документах, а группа Документов называется Коллекцией . Технически, документ означает набор пар ключ / значение или также называется полями или свойствами.

OrientDB использует такие понятия, как классы, кластеры и ссылки, для хранения, группировки и анализа документов.

***Модель графика*.** Структура данных графа – это модель данных, которая может хранить данные в форме вершин (узлов), соединенных ребрами (дугами). Идея базы данных графа OrientDB возникла из графа свойств. Вершина и ребро являются основными артефактами модели Graph. Они содержат свойства, которые могут сделать их похожими на документы.

***Модель объекта.*** Эта модель была унаследована объектно-ориентированным программированием и поддерживает наследование между типами (подтипы расширяют супертипы), полиморфизм при обращении к базовому классу и прямое связывание с / на объекты, используемые в языках программирования.

***Запись.*** Наименьшее устройство, которое вы можете загрузить и сохранить в базе данных. Записи могут быть сохранены в четырех типах.

* Документ
* Запись байтов
* темя
* край

***ID записи.*** Когда OrientDB генерирует запись, сервер базы данных автоматически назначает для записи идентификатор устройства, называемый RecordID (RID). RID выглядит как # <cluster>: <position>. <cluster> означает идентификационный номер кластера, а <position> означает абсолютную позицию записи в кластере.

***Документы.*** Документ является наиболее гибким типом записи, доступным в OrientDB. Документы имеют мягкую типизацию и определяются классами схемы с определенным ограничением, но вы также можете вставить документ без какой-либо схемы, т. Е. Он также поддерживает режим без схемы. Документы могут быть легко обработаны путем экспорта и импорта в формате JSON.

***RecordBytes.*** Тип записи совпадает с типом BLOB в RDBMS. OrientDB может загружать и хранить тип записи документа вместе с двоичными данными.

***Темя.*** База данных OrientDB – это не только база данных Document, но и база данных Graph. Новые концепции, такие как Vertex и Edge, используются для хранения данных в форме графика. В графовых базах данных основной единицей данных является узел, который в OrientDB называется вершиной. Vertex хранит информацию для базы данных.

***Край.*** Существует отдельный тип записи, называемый Edge, который соединяет одну вершину с другой. Края являются двунаправленными и могут соединять только две вершины. В OrientDB есть два типа ребер: один обычный, а другой легкий.

***Учебный класс.*** Класс представляет собой тип модели данных и концепцию, взятую из парадигмы объектно-ориентированного программирования. На основе традиционной модели базы данных документов данные хранятся в форме коллекции, а в модели реляционной базы данных данные хранятся в таблицах. OrientDB следует API документа вместе с парадигмой OPPS. Как концепция, класс в OrientDB имеет наиболее тесную связь с таблицей в реляционных базах данных, но (в отличие от таблиц) классы могут быть без схемы, с полной схемой или смешанными. Классы могут наследовать от других классов, создавая деревья классов. Каждый класс имеет свой собственный кластер или кластеры (создаются по умолчанию, если они не определены).

***Кластер.*** Кластер – это важная концепция, которая используется для хранения записей, документов или вершин. Проще говоря, кластер – это место, где хранится группа записей. По умолчанию OrientDB создает один кластер для каждого класса. Все записи класса хранятся в одном кластере с тем же именем, что и класс. Вы можете создать до 32 767 (2 ^ 15-1) кластеров в базе данных.

Класс CREATE – это команда, используемая для создания кластера с определенным именем. После создания кластера вы можете использовать кластер для сохранения записей, указав имя при создании любой модели данных.

***Отношения.*** OrientDB поддерживает два вида отношений: ссылочные и встроенные. Связанные отношения означают, что он хранит прямую ссылку на целевые объекты отношений. Встроенные отношения означают, что они хранят отношения в записи, которая их встраивает. Эти отношения сильнее, чем эталонные отношения.