

Ред База Данных дз

История развития СУБД

Истоки СУБД Ред Базы Данных и Firebird – в сервере баз данных Borland InterBase, зарождение которого произошло еще в 1984 году. Тогда были заложены основы, позволившие до нынешнего времени сохранить в новых СУБД простую файловую архитектуру. Благодаря ней специалисты могут эффективно запускать в работу хранилища новых форматов, эффективно обслуживать данные, легко восстанавливать работоспособность в случае технических сбоев, обеспечивать совместимость практически для всех существующих популярных платформ, включая мобильную ОС Android. Такой совместимостью могут похвастаться очень немногие СУБД.

Долгая история развития не могла не сказаться на самой СУБД. Постепенно в нее добавлялись новые рабочие механизмы и средства совместимости. В начале 2000-х годов Borland объявила о переводе InterBase в статус Open Source. Отдел разработки компании тогда располагался в России, это позволило отечественным программистам с самого начала активно участвовать в развитии этой платформы.

Позднее Borland вернулась к проприетарной версии InterBase 6.0. Но тогда на базе открытых исходных кодов InterBase 6.0 был запущен самостоятельный проект создания свободной СУБД Firebird. В 2006 году появилась корпорация РЕД СОФТ, которая поставила одной из своих целей продолжение разработки СУБД промышленного класса. Так появилась СУБД Ред База Данных, в основу которой были положены свободно доступные исходные коды Firebird.

Развитие Firebird и Ред База Данных продолжается по сей день. Но поскольку российская СУБД появилась не как форк, а как ответвление в развитии Firebird, их разработка ведется параллельно, а все достижения Firebird находят место и в составе Ред База Данных. Разработчики РЕД СОФТ продолжают активно сотрудничать с сообществом Firebird и сами являются контрибьюторами для этой свободной СУБД.

В то же время Ред База Данных содержит усовершенствования, которые отвечают требованиям соответствия российских регуляторов (в первую очередь, ФСТЭК).

В настоящее время в российской СУБД реализованы такие важные функции, как многопоточная архитектура SuperClassic Server, внешние хранимые процедуры, полнотекстовый поиск, улучшенная подсистема безопасности. В результате доработок Firebird 2.5 была обеспечена стабильная работа СУБД под высокой нагрузкой. В 2017 году компания РЕД СОФТ выпустила нынешнюю рабочую версию СУБД Ред База Данных 3.0, основанную на ядре Firebird 3.0.

Инструменты для взаимодействия с СУБД

Помимо стандартных средств взаимодействия типа командой строки существует приложение **Ред Эксперт**.

Его особенности:

- Поддержка любых версий Ред Базы Данных и Firebird "из коробки"
- Поддержка всех типов аутентификации
- Разработка и выполнение SQL команд
- Форматирование и автодополнения SQL команд
- Визуальное проектирование диаграммы БД с возможностями Reverse Engineering
- Создание и редактирование объектов баз данных с помощью графических инструментов
- Трейс менеджер
- Управление пользователями и правами

Для мониторинга: **Ред Мониторинг**

Какой database engine используется в вашей СУБД?

Ядро СУБД «Ред База Данных» построено на основе Firebird — популярной СУБД с открытым кодом

Как устроен язык запросов в вашей СУБД? Разверните БД с данными и выполните ряд запросов

Утилита для работы через командную строку - isql

Язык запросов - sql с некоторым количеством своих наворотов

создание базы:

```
SQL> create database 'path/test_db.fdb' page_size 8192
```

```
CON> user 'sysba' password 'password'
```

подключение к существующей базе:

```
SQL> connect localhost:<путь до бд или алиас> user sysdba password password;
```

CRUD:

```
SQL> CREATE TABLE DWARF(  
ID INT NOT NULL PRIMARY KEY,  
FIRSTNAME VARCHAR(12) NOT NULL,  
ALIAS VARCHAR(15),  
AGE INT NOT NULL
```

```
);  
commit;
```

```
SQL> INSERT INTO DWARF (ID,FIRSTNAME,ALIAS,AGE) VALUES  
(1,'Thorin','Oakenshield',195);  
SQL> commit;
```

```
SQL> INSERT INTO DWARF (ID, FIRSTNAME, AGE) VALUES (2,'Balin', 230);  
SQL> commit;
```

```
SQL> SELECT FIRSTNAME FROM DWARF;
```

```
SQL> UPDATE DWARF SET AGE = 235 WHERE FIRSTNAME = 'Balin ';  
SQL> commit;
```

```
SQL> DELETE FROM DWARF WHERE ID=1;  
SQL> commit;
```

```
SQL> DROP TABLE DWARF;  
SQL> SHOW TABLES;
```

Распределение файлов БД по разным носителям?

Есть такая опция. Можно указать конкретные файлы, куда сохранять данные. Однако управлять тем, куда попадут конкретные данные нельзя.

На каком языке/ах программирования написана СУБД?

Основная часть, включая ядро написана на C++. Однако для создания утилит администрирования и инструментов разработки также использовались языки Python и Java.

Какие типы индексов поддерживаются в БД? Приведите пример создания индексов.

- B-Tree
- Unique
- FullText

```
CREATE [UNIQUE] [ASC[ENDING] | DESC[ENDING]]  
INDEX <имя индекса> ON <таблица>  
{(<столбец> [, <столбец> ...]) | COMPUTED BY (<выражение>)};
```

Как строится процесс выполнения запросов в вашей СУБД?

Процесс обработки SQL-запроса состоит из нескольких взаимосвязанных компонентов:

1. SQL-оператор инициируется в клиентском приложении и передается на сервер базы данных через сеть.
2. После получения SQL-оператора реляционным движком сервера базы данных начинается его обработка.
3. Оптимизатор запросов анализирует дерево запросов и выбирает наиболее оптимальный способ выполнения SQL-оператора, создавая план выполнения.
4. План выполнения передается исполнителю запроса, который использует его для управления операциями чтения или изменения данных в соответствии с SQL-запросом.
5. Движок хранилища использует наиболее эффективные методы доступа для чтения и записи данных.
6. При чтении данных менеджер буферов проверяет, есть ли необходимые данные в кэше памяти, и при необходимости извлекает их с диска.
7. При записи данных с вставкой или обновлением менеджер транзакций гарантирует, что изменения происходят атомарно и сохраняют целостность базы данных.
8. Менеджер блокировок управляет блокировками, чтобы несколько транзакций могли выполняться одновременно, не влияя друг на друга.

Взаимодействие этих компонентов обеспечивает надежную и эффективную обработку запросов в системе управления базами данных.

Есть ли для вашей СУБД понятие «план запросов»? Если да, объясните, как работает данный этап.

План запросов в СУБД RedDB работает следующим образом:

1. При получении SQL-запроса от клиента, сервер RedDB выполняет синтаксический анализ запроса и строит дерево запроса.
2. Затем оптимизатор запросов RedDB анализирует дерево запроса и генерирует один или несколько вариантов плана выполнения запроса.
3. Оптимизатор запросов выбирает наиболее оптимальный план выполнения запроса на основе статистики, собранной о структуре и данных таблиц, вовлеченных в запрос.
4. После выбора оптимального плана выполнения, сервер RedDB компилирует его в двоичный код, который затем кэшируется в памяти для повторного использования.
5. При последующих запросах, если план выполнения уже существует в кэше, сервер Firebird использует его, чтобы ускорить обработку запроса.
6. При выполнении запроса, сервер RedDB использует план выполнения для определения порядка доступа к таблицам, выбора индексов, сортировки и

группировки данных, а также для выполнения других операций, необходимых для получения результата запроса.

7. Если план выполнения запроса не оптимален, оптимизатор запросов RedDB может перестроить его динамически во время выполнения запроса, чтобы улучшить производительность.

При обработке оператора выборки данных SELECT оптимизатор сервера базы данных строит план, который определяет, в каком порядке и с использованием каких механизмов (при помощи индексов или при последовательном переборе с возможным дополнительным упорядочиванием) будут выбираться данные из таблиц для получения выходного набора данных. Как правило, такой план является не худшим решением. При этом у вас есть возможность задать свой план поиска, основываясь на каких-то характеристиках базы данных, например, на размерах различных индексов, количестве строк в таблицах и др. В некоторых случаях грамотно составленный собственный план может повысить производительность системы при выборке данных. Утилита ISQL имеет возможность отобразить пользователю план извлечения данных с помощью команды SET PLAN ON. Для того, чтобы просто изучить план запроса (без выполнения самого запроса) необходимо ввести команду SET PLANONLY ON. Более подробный план можно получить при включении расширенного плана с помощью команды SET EXPLAIN ON.

```
PLAN <выражение для плана поиска>
<выражение для плана поиска> ::=
(<элемент плана> [, <элемент плана> ...])
| SORT (<элемент плана>)
| JOIN (<элемент плана> [, <элемент плана> ...])
| [SORT] MERGE (<элемент плана> [, <элемент плана> ...])
<элемент плана> ::= <основной элемент> | <выражение для плана поиска>
<основной элемент> ::= { <имя/псевдоним таблицы> | <имя представления> } {
NATURAL
| INDEX (<имя индекса> [, <имя индекса> ...])
| ORDER <имя индекса> [ INDEX (<имя индекса> [, <имя индекса> ...]) ] }
```

Поддерживаются ли транзакции в вашей СУБД? Если да, то расскажите о нем. Если нет, то существует ли альтернатива?

Транзакции есть:

```
SET TRANSACTION
```

```
[NAME <имя транзакции>]
```

```
[READ WRITE | READ ONLY]
```

```
[WAIT [LOCK TIMEOUT <кол-во секунд>] | NO WAIT]
[[ISOLATION LEVEL]
{ SNAPSHOT [TABLE STABILITY] | READ COMMITTED [[NO] RECORD_VERSION] } ]
[NO AUTO UNDO]
[IGNORE LIMBO]
[RESERVING <предложение резервирования> | USING <хендл базы данных>];
<предложение резервирования> ::=
<имя таблицы> [, <имя таблицы> ...]
[FOR [SHARED | PROTECTED] {READ | WRITE}]
[, <предложение резервирования>] ...
```

Какие методы восстановления поддерживаются в вашей СУБД. Расскажите о них.

- откат транзакции
- восстановление с фиксацией. Из (инкрементной) резервной копии: утилита **nbackup** с опцией -R(estore) <база данных> <резервный файл 0> <резервный файл 1> ...

Расскажите про шардинг в вашей конкретной СУБД. Какие типы используются? Принцип работы.

Шардинг не поддерживается в этой СУБД, но реализация его поддержки есть в roadmap на 3 года.

Возможно ли применить термины Data Mining, Data Warehousing и OLAP в вашей СУБД?

СУБД в первую очередь ориентирована на OLTP. Однако все термины можно применить, так как существуют инструменты и методы, которые позволяют использовать Ред База Данных в аналитических целях

Какие сообщества развивают данную СУБД? Кто в проекте имеет права на коммит и создание дистрибутива версий? Расскажите об этих людях и/или компаниях.

Развитием этой СУБД занимается компания Ред софт
Российская компания, сертифицирована как отечественный разработчик программного обеспечения

- импортозамещения информационных технологий государственной информатизации;
- совершенствования системы государственного управления средствами информационных технологий;

- развития отрасли информационных технологий, включая экспорт программных продуктов, работ и услуг за рубеж.

Как продолжить самостоятельное изучение языка запросов с помощью демобазы. Если демобазы нет, то создайте ее.

Демобазы для изучения возможностей языка запросов Ред Базы Данных создаются во время прохождения [руководства по Быстрому старту](#) или [руководства по sql](#)

Где найти документацию и пройти обучение

- [документация](#)
- [курс](#) "Администрирование СУБД Ред База Данных"

Как быть в курсе происходящего

- Лекции на [Яндекс дзен](#)
- [Тг канал с анонсами](#)