

# 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  
CREATE INDEX idx\_customer\_lastname ON customers (last\_name);
- Unique  
CREATE UNIQUE INDEX idx\_unique\_email ON users (email);
- FullText  
CREATE FULL TEXT INDEX idx\_ft\_description ON products (description);

## **Как строится процесс выполнения запросов в вашей СУБД?**

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

Существует планировщик запросов, однако запросить план выполнения нельзя.

1. **Анализ запроса:** Оптимизатор анализирует SQL-запрос, чтобы понять его структуру, операторы и условия.
2. **Генерация возможных планов выполнения:** Оптимизатор создает различные варианты планов выполнения запроса, которые могут быть эффективно использованы для извлечения данных.
3. **Оценка стоимости:** Каждый план выполнения оценивается с точки зрения его стоимости, учитывая различные факторы, такие как доступность индексов, объем данных и структуру таблиц.
4. **Выбор оптимального плана:** Оптимизатор выбирает наиболее эффективный план выполнения на основе оценки стоимости и других критериев.

## 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.

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 людей и/или компаниях.**

Развитием этой СУБД занимается компания Ред софт

## **Создайте свои собственные данные для демонстрации работы СУБД.**

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

Демобазы для изучения возможностей языка запросов Ред База 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](#)