Министерство образования и науки Украины
Севастопольский национальный технический университет

АДМИНИСТРИРОВАНИЕ ИНФОРМАЦИОННЫХ СИСТЕМ

Методические указания
к выполнению лабораторных работ №1 — № 2
по дисциплине
“Администрирование информационных систем”
для студентов специальности
7.080401 – "Информационные управляющие системы и технологии"
всех форм обучения

Севастополь
2008

УДК 004.92

Методические указания к выполнению лабораторных работ №1 — № 2 по дисциплине “Администрирование информационных систем”/ Сост. Ю.В. Доронина, А.В. Исаева. — Севастополь: Изд-во СевНТУ, 2008. — 32 с.

Целью методических указаний по дисциплине “Администрирование информационных систем” является выработка у студентов практических навыков администрирования информационных систем и закрепление их на примере реальных информационных систем. Рассматриваются вопросы управления пользователями на сервере, ограничения доступа к базам данных, нарушения целостности файла базы данных, средства сервера и действия администратора ИС по обнаружению повреждений файлов баз данных, а также защите от несанкционированного доступа из внешней сети.

Методические указания составлены в соответствии с требованиями программы дисциплины “Администрирование информационных систем” для студентов пятого курса дневной и заочной форм обучения специальности 7.080401 и утверждены на заседании кафедры информационных систем (протокол №7 от 6 февраля 2008 года).

Допущено учебно-методическим центром СевНТУ в качестве методических указаний.

Рецензент: Г. Г. Сергеев, канд. техн. наук, доцент кафедры КиВТ ;

СОДЕРЖАНИЕ

ОБЩИЕ ТРЕБОВАНИЯ

1. Цель и задачи лабораторных работ

Цель настоящих лабораторных работ состоит в исследовании вопросов нарушения целостности файла базы данных, средства сервера по обнаружению повреждений, а также действия администратора сервера в случае обнаружения повреждений.

Задачами выполнения лабораторных работ являются:

- углубленное изучение основных теоретических положений дисциплины «Администрирование информационных систем»;
- получение практических навыков при работе с базами данных и их администрирование.

2. Описание лабораторной установки

Лабораторные работы выполняются на ЭВМ с частотой работы процессора не менее 200 МГц при наличии оперативной памяти объемом не менее 128 Мб. В качестве операционной системы могут использоваться ОС Windows, начиная с Windows 98, в качестве СУБД – InterBase или FireBird (открытая версия InterBase).

3. Порядок выполнения лабораторных работ

Варианты заданий студенты получают от преподавателя или выбирают самостоятельно по номеру зачетной книжки. Номер варианта определяется как остаток от деления двух последних цифр номера зачетной книжки на количество вариантов задания. Например, две последние цифры 23 и всего 12 вариантов. Остаток от деления 23 на 12 равен 11. Следовательно, номер варианта задания 11.

При выполнении каждой лабораторной работы необходимо:

- изучить соответствующие теоретические положения, методические указания;
- выполнить задание на лабораторную работу на ЭВМ в соответствии с вариантом;
- ответить на контрольные вопросы;
- подготовить и защитить отчет о выполненной работе.

4. Содержание отчета

Отчеты по лабораторной работе оформляются каждым студентом индивидуально. Отчет должен включать: титульный лист с указанием названия и номера лабораторной работы, номера варианта; цель работы; краткие теоретические сведения; постановку задачи; текстовое и графическое описание в виде скриншотов (копий экрана) хода выполнения лабораторной работы; тексты и описания программ, результаты их выполнения; выводы по работе.

ЛАБОРАТОРНАЯ РАБОТА №1

“Управление пользователями ИС. Разграничение доступа к ИС”

1. Цель работы

Изучить основы управления пользователями ИС на примере СУБД Interbase. Исследовать способы ограничения доступа пользователей к информации в СУБД Interbase (4 часа).

2. Основные положения

2.1. Защита СУБД Interbase

СУБД InterBase предоставляет несколько методов для защиты баз данных (БД). В данной лабораторной работе рассмотрены основные возможности администрирования, такие как управление пользователями на сервере БД (добавление, модификация, удаление учетных записей пользователей), а также определение их привилегий относительно БД, содержащихся на сервере, с помощью SQL-команд.

Модель защиты

Защита для InterBase основывается на главной базе данных защиты (БДЗ), она своя для каждого сервера. БДЗ – файл с именем isc4.gdb – содержит учетную запись для каждого пользователя, имеющего разрешение на соединение с базами данных и сервисами InterBase. Запись включает имя пользователя, входящего в систему, и соответствующий зашифрованный пароль.

Перед выполнением любых задач администрирования необходимо сначала войти на сервер, после этого можно соединиться с базой данных, находящейся на сервере.

Для того чтобы войти на сервер, пользователь должен ввести имя (логин), однозначно идентифицирующее его в БДЗ, и пароль. При передаче по сети пароль шифруется. Имя пользователя и пароль сопоставляются с записями в БДЗ. Если соответствие найдено, позволяется вход в систему.

Пользователь SYSDBA – системный администратор

Каждый сервер InterBase имеет пользователя с именем SYSDBA с заданным по умолчанию паролем «masterkey». SYSDBA (System Database Administrator) – специальная учетная запись, на SYSDBA не распространяются ограничения, описанные операторами SQL. SYSDBA может выполнять все задачи администрирования: создавать резервные копии БД, завершать работу сервера и т.д.

Первоначально SYSDBA – единственный пользователь на сервере. Чтобы разрешить доступ другим пользователям, SYSDBA должен присоединиться к InterBase и завести учетные записи для новых пользователей. Только SYSDBA может модифицировать БДЗ, добавлять, удалять, или изменять учетные записи пользователей. Для выполнения своих функций SYSDBA может использовать утилиты gsec или IBConsole. Пользователь SYSDBA имеет право соединиться с любой БД на сервере.

Другие пользователи

Обычно создается отдельная учетная запись для каждого человека, работающего с БД. Для того чтобы упростить администрирование, можно создать одну учетную запись пользователя для группы людей. Например, пользователь FINANCE может использоваться бухгалтерами и финансистами, пользователь MANAGER может использоваться управляющим персоналом. Если пользователи Иван и Петр имеют одинаковые права на доступ к данным, для них можно создать одну учетную запись.

База данных защиты isc4.gdb

Учетные записи пользователей хранятся в БДЗ с именем isc4.gdb. Утилита защиты gsec позволяет отображать, добавлять, изменять, или удалять информацию из isc4.gdb.

Утилита IBConsole обеспечивает графический интерфейс для выполнения тех же самых действий. Следующая таблица описывает содержание isc4.gdb:

Таблица 1 – Содержание учетной записи пользователя

Столбец	Обязателен?	Описание
Имя пользователя	Да	Имя, которое пользователь указывает при подключении, максимальная длина – 31 символ; имя не чувствительно к регистру; пробелы не допустимы
Пароль	Да	Пароль пользователя, чувствителен к регистру. При сравнении учитываются только первые восемь символов. Максимальная длина 32 символа. Пробелы не допустимы
UID	Нет	Целочисленный идентификатор пользователя
GID	Нет	Целочисленный идентификатор группы пользователя
Полное имя	Нет	Полное настоящее имя пользователя

Права пользователя, описанные в операторах SQL

Присоединение к серверу не дает права обычному пользователю на изменение или просмотр данных, содержащихся в базах данных. Право каждого пользователя на выполнение каких–либо операций с данными должно быть предварительно описано для каждой БД. Права, предоставленные с использованием описателя PUBLIC, дают право на доступ любому пользователю. Для назначения и отмены прав пользователей используются операторы SQL (см. параграф 2.2).

Группы пользователей

Interbase поддерживает назначение привилегий группам пользователей. В частности, это реализовано с использованием механизма ролей. Роли SQL назначаются отдельно для каждой БД.

Роли ANSI SQL3

Interbase поддерживает защиту на уровне групп так же, как это описано в ISO-ANSI Working Draft for Database Language.

Чтобы создать роль для определенной БД, надо присоединиться к ней (изначально только как администратор или владелец БД) и запустить специальную утилиту – редактор Interactive SQL (isql). Создание ролей представляет собой процесс из трех шагов.

1. Создать роль с помощью оператора CREATE ROLE
2. Назначить созданной роли права на каждую таблицу или столбцы таблицы с использованием оператора GRANT
3. Назначить роль пользователям с помощью оператора GRANT

Подробное описание используемых операторов SQL находится в параграфе 2.2.

Для получения пользователем прав, назначенных какой-либо роли, существует несколько способов. Первый способ заключается в том, что пользователь присоединяется к БД на сервере с указанием своей роли средствами SQL (в окне Interactive SQL). Например, пользователь с логином user1 и паролем peanuts соединяется с БД, хранящейся в файле foo.gdb, под ролью sales:

CONNECT ’foo.gdb’ USER ’user1’ PASSWORD ’peanuts’ ROLE sales;

При втором способе пользователь, зарегистрировавшись на сервере обычным образом (указав только свой логин и пароль в окне регистрации), заходит на сервер и затем присоединяется к определенной БД с указанием роли для этой БД (опция соединения Databases — Connect As).

Используя первый способ подключения, пользователь может иметь только одну роль за соединение с сервером и не может менять роль, пока он присоединен. Чтобы поменять роль, нужно отсоединиться и снова присоединиться с указанием другой роли.

В этом случае роль пользователя можно указать следующими способами:

- при соединении с помощью IBConsole роль вводится в соответствующее поле окна соединения;
- роль можно указать программно с использованием Interbase API;
- роль можно указать, используя оператор CONNECT.

Дополнительные средства защиты Interbase

Interbase имеет ограничения на использование утилит Interbase для увеличения уровня защиты. Кроме этого, существует несколько предписаний, которые надо выполнять, чтобы не допустить «прорех» в защите.

Ограничение на использование утилит

В Interbase требуется, чтобы утилиты gbak, gstat и gfix мог выполнять только владелец БД (пользователь, который ее создал) или SYSDBA. Несмотря на то, что резервное копирование может производить владелец или SYSDBA, восстановление из резервной копии может сделать любой пользователь, так как не существует понятия «пользователь» для файла резервной копии. Также имеется дополнительное ограничение, в соответствии с которым восстановление из резервной копии для существующей БД может делать только владелец или SYSDBA. Таким образом, хранить резервные копии БД необходимо в защищенном хранилище, так как резервная копия БД может быть восстановлена на другом сервере Interbase любым пользователем.

Управление пользователями с помощью IBConsole

Управление пользователями выполняется с помощью диалогового окна User Information, представленного на рисунке 1. С помощью окна производится добавление новых пользователей, модификация существующих и удаление учетных записей. Чтобы выполнить указанные операции, необходимо установить соединение с сервером.

Диалоговое окно User Information можно вывести на экран несколькими способами:

- выделить сервер в списке серверов, выбрать команду меню Server | User Security, выбрать команду User Security контекстного меню, либо нажать кнопку на панели быстрого доступа;
- выбрать строку Users из древовидного списка сервера и дважды щелкнуть на одном из пользователей.

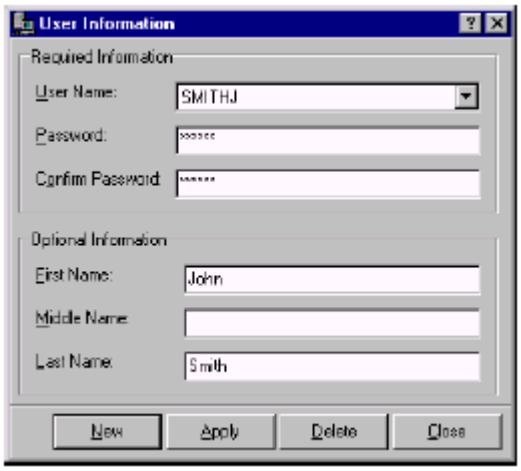

Рисунок 1 – Диалоговое окно User Information

Добавление пользователя

1. Нажмите кнопку «New». Кнопки New и Delete станут недоступными, кнопка Close изменится на Cancel
2. Введите имя нового пользователя
3. Введите пароль (Password), подтверждение пароля (Confirm Password)
4. Введите дополнительную информацию (Имя, Фамилия, Отчество)
5. Нажмите кнопку Apply

Модификация учетной записи пользователя

1. Из выпадающего списка User Name выберите пользователя
2. Внесите изменения в любые поля за исключением имени пользователя
3. Нажмите кнопку Apply

Единственный способ изменить имя пользователя – удалить его и создать нового с требуемым именем.

Удаление пользователя

1. Из выпадающего списка User Name выберите пользователя.
2. Нажмите кнопку Delete

Несмотря на то, что можно удалить пользователя SYSDBA, делать этого не рекомендуется, так как управлять пользователями в дальнейшем станет невозможным. После удаления SYSDBA придется брать БДЗ из чистой инсталляции InterBase (где пользователь SYSDBA есть) и заново заводить всех остальных пользователей.

Управление пользователями с помощью gsec

gsec – это утилита командной строки, по функциональному назначению аналогичная диалоговому окну User Information в IBConsole. Способы запуска:

- с указанием параметров в командной строке
- gsec –user sysdba –password masterkey
- перед запуском утилиты установить переменные окружения ISC_USER и ISC_PASSWORD
- можно запустить gsec, если войти под учетной записью root на UNIX или Administrator на Windows NT.

Для интерактивного использования gsec введите GSEC в командной строке.

Запуск gsec с удаленной машины

Используйте ключ –database с указанием строки соединения с удаленной машиной. Например:

gsec –database jupiter:/usr/interbase/isc4.gdb

Команды gsec

Первую часть команды указывать обязательно. Часть команды, заключенная в квадратные скобки, опциональна. Подробно команды описаны в таблице 2.

Таблица 2 – Команды gsec

Команда	Описание
di[splay]	Отображает все строки из isc4.gdb
di[splay] name	Отображает информацию только о пользователе с именем name
a[dd] name –pw password [option argument] [option argument …]	Добавляет пользователя с именем name в isc4.gdb с паролем password. Каждый параметр option и соответствующий аргумент argument указывает дополнительную информацию о пользователе
mo[dify] name [options]	Подобна команде add, но имя name должно существовать в isc4.gdb
de[lete] name	Удаляет пользователя с именем name из isc4.gdb
h[elp] или ?	Отображает синтаксис и команды gsec
q[uit]	Выходит из интерактивного режима

Опции для команд gsec описаны в таблице 3, а ошибки, которые могут возникнуть при работе с gsec, – в таблице 4.

Таблица 3 – Опции команд gsec

Опция	Назначение
-password или –pa <пароль>	Пароль пользователя, производящего изменения
-user string	Имя пользователя, производящего изменения
-pw string	Пароль пользователя
-uid integer	Идентификационный номер пользователя
-gid integer	Идентификационный номер группы
-fname string	Настоящее имя пользователя
-mname string	Фамилия пользователя
-lname string	Отчество пользователя

Таблица 4 – Ошибки gsec

Сообщение об ошибке	Причины и меры по устранению
Add record error	Ошибка добавления записи. Указан существующий пользователь или неправильный синтаксис, или нет прав на запуск gsec. Измените имя пользователя, используя modify.
<string> already specified	<string> уже указана. Для одного поля в учетной записи данные указаны дважды. Повторите ввод команды.
Ambiguous switch specified	Указана двусмысленная команда. Указанная часть команды не может точно определить желаемое действие.
Delete record error	Ошибка удаления записи. Не хватает прав для использования gsec.
Error in switch specifications	Ошибочный синтаксис команды. Другие сообщения об ошибках скажут, где именно произошла ошибка.
Find/delete record error	Не найдена запись для удаления. Или команда не нашла учетную запись, или не хватает прав для запуска gsec.
Find/display record error	Не найдена запись для отображения. Или команда не нашла учетную запись, или не хватает прав для запуска gsec.
Find/modify record error	Не найдена запись для модификации. Или команда не нашла учетную запись, или не хватает прав для запуска gsec.
Incompatible switches specified	Указаны несовместимые команды.
Invalid parameter, no switch defined	Ошибочный параметр, не указана команда.
Invalid switch specified	Указана неверная команда.
Modify record error	Ошибка модификации записи. Ошибочный синтаксис команды modify, не хватает прав для запуска gsec.

2.2 Команды SQL для реализации разграничения доступа

CREATE ROLE

Создает роль.

Синтаксис: CREATE ROLE rolename;

Ролям, созданным с помощью CREATE ROLE, могут назначаться привилегии (права на данные). Роли могут быть назначены пользователям, после чего пользователь имеет тот же список прав, что и роль. Роль пользователя указывается при соединении. Оператор GRANT используется для назначения привилегий (ALL, SELECT, INSERT, UPDATE, DELETE, EXECUTE, REFERENCES) роли и для назначения роли пользователю. Оператор REVOKE отменяет действие оператора GRANT.

Пример: следующий оператор создает роль с именем “administrator”

CREATE ROLE administrator;

Описание аргументов

rolename – имя роли, уникальное среди других имен ролей в БД.

DROP ROLE

Удаляет роль из БД.

Синтаксис DROP ROLE rolename;

Оператор DROP ROLE удаляет роль созданную оператором CREATE ROLE. Все привилегии, назначенные роли, теряются. Роль может быть удалена создателем роли, пользователем SYSDBA или пользователем с правами root.

Пример: следующий оператор удаляет роль с именем “administrator”

DROP ROLE administrator;

Описание аргументов

rolename – имя существующей роли.

GRANT

Дает пользователю права на указанные объекты БД. Описание аргументов оператора GRANT находится в таблице 5.

Синтаксис:

GRANT < privileges> ON [TABLE] { tablename | viewname} TO { <object> | <userlist> | GROUP UNIX_group}
| EXECUTE ON PROCEDURE procname TO { <object> | <userlist>}
| < role_granted> TO {PUBLIC | < role_grantee_list>};

< privileges> = {ALL [PRIVILEGES] | < privilege_list>}

<privilege_list> = SELECT
| DELETE
| INSERT
| UPDATE [(col [, col …])]
| REFERENCES [(col [, col …])]
[, < privilege_list> …]

<object> = PROCEDURE procname

| TRIGGER trigname
| VIEW viewname
| PUBLIC
[, <object> …]

<userlist> = [USER] username
| rolename
| Unix_user}
[, <userlist> …]
[WITH GRANT OPTION]

< role_granted> = rolename [, rolename …]

<role_grantee_list> = [USER] username [, [USER] username …]
[WITH ADMIN OPTION]

Таблица 5 – Описание аргументов GRANT

Аргумент	Описание
privilege_list	Имена назначаемых привилегий; допустимые значения – SELECT, DELETE, INSERT, UPDATE и REFERENCES (подробное описание см. в таблице 6)
col	Столбец, к которому применяются привилегии
tablename	Имя существующей таблицы, к которой применяются привилегии
viewname	Имя существующего представления (view), к которому применяются привилегии
GROUP unix_group	В операционной системе (ОС) UNIX – имя группы, определенное в /etc/group
object	Имя существующей процедуры, триггера, или представления; PUBLIC является допустимым значением
userlist	Пользователь из isc4.gdb или имя роли, созданной оператором CREATE ROLE
WITH GRANT OPTION	Дает право всем из userlist назначать другим пользователям свои привилегии (перечисленные в операторе GRANT)
rolename	Существующая роль, созданная оператором CREATE ROLE.
role_grantee_list	Список пользователей, которым дается право на rolename; учетные записи пользователей должны существовать в isc4.gdb
WITH ADMIN OPTION	Дает право пользователям из <role_grantee_list>, имеющим роли, перечисленные в <role_granted>, назначать эти роли другим пользователям

Описание: GRANT назначает привилегии и роли на объекты БД пользователям, ролям или другим объектам БД. После создания объекта БД право на него имеет только его создатель, и только создатель объекта может дать право на объект другим пользователям.

Таблица 6 – Привилегии и соответствующие им действия, разрешаемые командой GRANT

Привилегия	Разрешаемые действия
ALL	Выполнять SELECT, DELETE, INSERT, UPDATE и REFERENCES
SELECT	Получать (читать) строки из таблицы или представления
DELETE	Удалять строки из таблицы или представления
INSERT	Добавлять новые строки в таблицы или представления
UPDATE	Изменять текущие значения в определенных столбцах таблиц или представлений; можно указать подмножество столбцов, в которых позволяются изменения.
EXECUTE	Выполнять хранимую процедуру
REFERENCES	Ссылаться на указанные столбцы с помощью внешнего ключа. Если данное право назначается, то разрешение должно выдаваться, как минимум, на столбцы первичного ключа.

Чтобы осуществлять доступ к таблице или представлению, необходимо иметь право на операции SELECT, INSERT, UPDATE, DELETE или REFERENCES для данной таблицы. Права SELECT, INSERT, UPDATE, DELETE и REFERENCES могут быть назначены с помощью единственного описателя ALL.

Пользователь или объект БД должен иметь право EXECUTE для вызова хранимой процедуры из приложения.

Для того чтобы назначить права группе пользователей, создайте роль с помощью оператора CREATE ROLE. Затем используйте оператор GRANT privilege TO rolename для назначения желаемых прав роли. Используйте GRANT rolename TO userlist для назначения роли пользователю или пользователям. Право на роль может назначаться пользователю с помощью оператора GRANT, лишить пользователя права на роль можно с помощью оператора REVOKE. Чтобы воспользоваться привилегиями роли, пользователь должен указать роль во время подключения к БД.

Для того чтобы разрешить другому пользователю ссылаться на поля таблицы с помощью механизма внешних ключей, необходимо дать владельцу ссылающейся таблицы право REFERENCES на всю таблицу или только на столбцы, составляющие первичный ключ. Также необходимо давать право REFERENCES или SELECT на первичный ключ таблицы любому пользователю, осуществляющему запись в таблицу, содержащую внешний ключ.

Замечание: если доступ по чтению не является существенным, можно дать право REFERENCES на первичный ключ таблицы всем (PUBLIC), и этим облегчить администрирование.

Если назначается право REFERENCES, оно должно распространятся на все столбцы первичного ключа. Если право REFERENCES назначается на таблицу целиком, столбцы, не являющиеся частью ключа, не затрагиваются в любом случае.

Когда пользователь объявляет внешний ключ на таблицу, которой владеет другой пользователь, InterBase проверяет, имеет ли пользователь право REFERENCES на данную таблицу. Право REFERENCES можно назначить роли.

Чтобы дать пользователям возможность назначать привилегии другим пользователям, укажите userlist с описателем WITH GRANT OPTION. Пользователи

могут назначать другим пользователям только те привилегии, которыми они владеют сами.

Чтобы дать право всем пользователям, укажите описатель PUBLIC вместо списка пользователей. Описанное таким образом право дает привилегии только пользователям, но не объектам БД.

Привилегии могут быть аннулированы с помощью оператора REVOKE. Если право было дано с использованием ALL, то с тем же описателем оно и должно удаляться. Если право было дано PUBLIC, то и отнять его можно только у PUBLIC.

Примеры:

Следующий оператор дает право SELECT и DELETE пользователю CHLOE на таблицу COUNTRY. Описатель WITH GRANT OPTION дает пользователю право давать привилегии другим пользователям.

GRANT SELECT, DELETE ON COUNTRY TO CHLOE WITH GRANT OPTION;

Следующий оператор SQL дает право SELECT и UPDATE процедуре с именем GET_EMP_PROJ на таблицу JOB:

EXEC SQL
GRANT SELECT, UPDATE ON JOB TO PROCEDURE GET_EMP_PROJ;

Следующий оператор SQL дает право EXECUTE процедуре ADD_EMP_PROJ и пользователю LUIS

EXEC SQL
GRANT EXECUTE ON PROCEDURE GET_EMP_PROJ
TO PROCEDURE ADD_EMP_PROJ, LUIS;

Следующий пример создает роль с именем “administrator”, дает ей право UPDATE на таблицу table1, затем назначает роль пользователям user1, user2, и user3.

CREATE ROLE administrator;
GRANT UPDATE ON table1 TO administrator;
GRANT administrator TO user1, user2, user3;

REVOKE

Аннулирует привилегии пользователя на указанные объекты БД. Описание аргументов оператора REVOKE находится в таблице 7.

REVOKE [GRANT OPTION FOR] < privileges> ON [TABLE]
{ tablename | viewname}
FROM { <object> | <userlist> | < rolelist> | GROUP UNIX_group}
| EXECUTE ON PROCEDURE procname
FROM { <object> | <userlist>}
| < role_granted> FROM {PUBLIC | < role_grantee_list>}};

< privileges> = {ALL [PRIVILEGES] | < privilege_list>}

<privilege_list> = {

SELECT
| DELETE
| INSERT
| UPDATE [(col [, col …])]
| REFERENCES [(col [, col …])]
[, < privilege_list> …]}}

<object> ={
PROCEDURE procname
| TRIGGER trigname
| VIEW viewname
| PUBLIC
[, <object>]}

<userlist> = [USER] username [, [USER] username …]

< rolelist> = rolename [, rolename]

< role_granted> = rolename [, rolename …]

<role_grantee_list> = [USER] username [, [USER] username …]

Таблица 7 – Описание аргументов REVOKE

privilege_list	Имена отменяемых привилегий; допустимые значения SELECT, DELETE, INSERT, UPDATE, REFERENCES (подробное описание см. в таблице 8).
GRANT OPTION FOR	Отменяет право назначать привилегии, указанные в REVOKE; не применяется к объекту
Col	Столбец, право на который отменяется
tablename	Имя существующей таблицы, право на которую отменяется
viewname	Имя существующего представления (view), право на которое отменяется
GROUP unix_group	В ОС UNIX – имя группы, определенное в /etc/group
object	Имя существующего объекта БД
userlist	Пользователь из isc4.gdb или имя роли, созданной оператором CREATE ROLE
rolename	Существующая роль, созданная оператором CREATE ROLE.
role_grantee_list	Список пользователей, которым дается право на rolename; учетные записи пользователей должны существовать в isc4.gdb

Таблица 8 – Привилегии и отменяемые командой REVOKE действия.

Привилегия	Отменяемые действия
ALL	Выполнять SELECT, DELETE, INSERT, UPDATE и REFERENCES
SELECT	Получать строки из таблицы или представления
DELETE	Удалять строки из таблицы или представления
INSERT	Добавлять новые строки в таблицы или представления
UPDATE	Изменять текущие значения в определенных столбцах таблиц или представлений; можно указать подмножество столбцов, в которых позволяются изменения.
EXECUTE	Выполнять хранимую процедуру
REFERENCES	Ссылаться на указанные столбцы с помощью внешнего ключа. Если данное право назначается, то разрешение должно выдаваться, как минимум, на столбцы первичного ключа.

GRANT OPTION FOR отменяет право пользователя давать права другим пользователям.

Для REVOKE существуют следующие ограничения:

- право может отобрать тот пользователь, который его дал;
- один пользователь может назначить права многим другим пользователям; отмена права первого пользователя повлияет только на него, т.е. пользователи, получившие право от данного пользователя, не изменят своих прав;
- право, данное PUBLIC, может быть отнято только у PUBLIC.

Если у пользователя user1 отбирают роль, то все привилегии, розданные пользователем user1 другим пользователям благодаря праву назначать привилегии для роли, также отменяются.

Примеры:

Оператор отбирает право SELECT пользователя MIREILLE на таблицу COUNTRY:

```
REVOKE SELECT ON COUNTRY FROM MIREILLE;
```

Оператор отбирает право EXECUTE на процедуру GET_EMP_PROJ у процедуры ADD_EMP_PROJ и пользователя LUIS:

```
REVOKE EXECUTE ON PROCEDURE GET_EMP_PROJ
FROM PROCEDURE ADD_EMP_PROJ, LUIS;
```

CONNECT

Выполняет соединение с одной или несколькими БД. Оператор доступен в SQL. Подмножество оператора CONNECT доступно в isql.

Синтаксис, используемый в isql:

```
CONNECT ’filespec’ [USER ’username’][PASSWORD ’password’]
[CACHE int] [ROLE ’rolename’]
```

Синтаксис, используемый в SQL:
CONNECT [TO] {ALL | DEFAULT} < config_opts>
| < db_specs> < config_opts> [, < db_specs> < config_opts>…];

< db_specs> = dbhandle
| {’ filespec’ | : variable} AS dbhandle

< config_opts> = [USER {’ username’ | : variable}]
[PASSWORD {’ password’ | : variable}]
[ROLE {’ rolename’ | : variable}]
[CACHE int [BUFFERS]]

Описание аргументов

{ALL | DEFAULT} — соединиться со всеми БД, указанными в SET DATABASE; опции, указанные с CONNECT TO ALL, применяются ко всем БД.

’filespec’ — имя файла БД; может включать полный путь и имя удаленного компьютера. Если задаваемая строка содержит пробелы, ее необходимо заключить в кавычки.

Dbhandle — дескриптор (handle) БД, объявленный в предыдущем операторе SET DATABASE; может использоваться в приложениях на встроенном SQL, но не в isql.

:variable — переменная, содержащая имя БД, имя пользователя или пароль. Не допустимо в isql.

AS dbhandle — соединяется с БД и присваивает предварительно объявленному дескриптору указатель на БД. Не допустимо в isql.

USER {’username’ | :variable} — строка или переменная, содержащая имя существующего пользователя.

PASSWORD {‘password’ | :variable} — строка или переменная, содержащая пароль пользователя.

ROLE {‘rolename’ | :variable} — строка или переменная, содержащая роль пользователя.

CACHE int [BUFFERS] — устанавливает количество буферов в кэш-памяти БД. Данное число (целочисленное, по умолчанию равно 256) определяет количество страниц памяти БД, с которыми программа может работать одновременно. Максимальное значение зависит от параметров операционной системы (ОС) и компьютера. Нельзя использовать назначение кэш-памяти, если имя БД описано с помощью filespec.

Описание оператора CONNECT:

Инициализирует структуры данных БД.

Определяет, находится ли БД на вызывающем компьютере (локальная БД) или на другом компьютере (удаленная БД). Если InterBase не может найти БД, происходит ошибка.

Опционально устанавливает одно или больше имен пользователей, паролей и ролей. Все имена пользователей, пароли и роли должны быть известны. При сравнении паролей учитываются только первые 8 символов. Если установлены переменные окружения ISC_USER и ISC_PASSWORD, и пользователь,

определяемый данными переменными, не существует в БДЗ, выдается ошибка “undefined user name and password.”

Выполняет соединение с БД и проверяет сигнатуру файла БД. Файл БД должен содержать неповрежденную структуру БД, версия дисковой структуры файла должна распознаваться текущей версией InterBase. В противном случае возвращается ошибка.

Опционально создает дескрипторы БД, объявленные в операторе SET DATABASE.

Создает буфер БД для процесса, соединяющегося с БД.

В SQL перед тем, как БД может быть открыта с помощью CONNECT, ее предварительно надо объявить с помощью оператора SET DATABASE. При работе с isql оператор SET DATABASE не требуется. Несмотря на то, что CONNECT может открывать более чем одну БД, лучше использовать несколько операторов для облегчения понимания текста. При соединении используется набор символов (character set) по умолчанию (NONE). Можно определить другой набор символов с помощью оператора SET NAMES, он должен следовать до оператора CONNECT.

Подмножество оператора CONNECT доступно из isql. С помощью isql можно соединиться только с одной БД за один раз. Перед каждым следующим оператором CONNECT происходит отсоединение от предыдущего соединения.

Примеры:

Оператор открывает БД из isql.

```
CONNECT ’employee.gdb’ USER ’ACCT_REC’ PASSWORD ’peanuts’;
```

Оператор встроенного SQL присоединяется БД, (путь к файлу БД хранится в переменной) и определяет предварительно объявленный дескриптор:

```
EXEC SQL
SET DATABASE DB1 = ’employee.gdb’;
EXEC SQL
CONNECT :db_file AS DB1;
```

Оператор встроенного SQL присоединяется к employee.gdb, выделяет 150 буферов:

```
EXEC SQL
CONNECT ’accounts.gdb’ CACHE 150;
```

Оператор встроенного SQL присоединяет пользователя ко всем предварительно объявленным с помощью SET DATABASE БД:

```
EXEC SQL
CONNECT ALL USER ’ACCT_REC’ PASSWORD ’peanuts’
CACHE 50;
```

Оператор встроенного SQL присоединяет employee.gdb с 80 буферами и employee2.gdb с 75 буферами (по умолчанию):

```
EXEC SQL
CONNECT ’employee.gdb’ CACHE 80, ’employee2.gdb’;
```

Оператор встроенного SQL соединяется с двумя БД, для каждой указывая своего пользователя:

```
EXEC SQL
CONNECT
:orderdb AS DB1 USER ’ACCT_REC’ PASSWORD ’peanuts’,
:salesdb AS DB2 USER ’ACCT_PAY’ PASSWORD ’payout’;
```

3. Порядок выполнения работы

1. Ознакомиться с теоретическим материалом.
2. Добавить пользователя к БДЗ с помощью утилиты IBConsole, изменить его параметры, затем удалить;
3. Добавить пользователя к БДЗ с помощью утилиты командной строки gsec, изменить его параметры, затем удалить;
4. Добавить себя как пользователя к БДЗ, зарегистрироваться на сервере как новый пользователь.
5. Создать базу данных по схеме согласно варианту задания. Для создания структуры БД использовать утилиту isql. Занести образцы данных в БД (не менее 8 строк в таблице).
6. Для заданной БД следует создать несколько ролей согласно варианту задания. Далее зарегистрировать на сервере нескольких (4 – 6) пользователей. Назначить пользователям роли.
7. Некоторым пользователям (1 – 2) назначить привилегии без использования ролей, разрешить им наделять своими привилегиями других пользователей.
8. Присоединяясь каждый раз к БД с помощью isql как один из пользователей, продемонстрировать их возможности относительно данной БД.
9. Лишить одного из пользователей всех его привилегий.
10. Оформить результаты работы в виде отчета и защитить его.

4. Контрольные вопросы

1. Методы защиты БД в СУБД Interbase. Что собой представляет БДЗ?
2. Пользователь SYSDBA и его полномочия. Можно ли удалить пользователя SYSDBA?
3. Как можно создать, удалить учетную запись пользователя, редактировать ее параметры?
4. Понятия «роль», «привилегия». Область действия роли. Виды привилегий.
5. Укажите все способы наделения пользователя привилегиями. Какой пользователь может назначить другому пользователю роль?
6. Как полностью/частично аннулировать пользователю его привилегии? Какой пользователь может это сделать?
7. Приведите примеры использования оператора CONNECT средствами встроенного SQL и с помощью isql.

ЛАБОРАТОРНАЯ РАБОТА №2

“Резервное копирование и восстановление Interbase. Проверка целостности и восстановление БД ”

1. Цель работы

Исследовать способы резервного копирования и восстановления данных в СУБД Interbase. Изучить причины и последствия повреждения файла базы данных в информационных системах, использующих СУБД InterBase, и возможности, предоставляемые сервером для проверки целостности и восстановления файла БД.

2. Основные положения

2.1. Резервное копирование и восстановление

В процессе резервного копирования содержимое базы данных копируется в файл. Резервное копирование необходимо производить регулярно, в случае необратимого повреждения данных восстановить содержимое БД можно только из резервной копии. Повреждение данных может наступить при внезапном выключении питания сервера, при возникновении сбоев жесткого диска и т.д. Восстановление есть обратный резервному копированию процесс, при котором содержимое базы данных из файла копируется на сервер.

Преимущества резервного копирования и восстановления

Вся база данных Interbase содержится в одном файле. Таким образом, можно производить резервное копирование и восстановление с помощью средств операционной системы (ОС). Тем не менее, использование встроенных средств (утилита командной строки gbak и IBConcole) имеет несколько преимуществ:

- повышается производительность БД, т.к. операция резервного копирования производит «сборку мусора» и балансировку индексов;
- размер БД уменьшается за счет физического удаления помеченных на удаление записей, существующие записи пакуются;
- можно изменить размер страницы БД и разбить БД на несколько файлов;
- нет необходимости останавливать сервер для выполнения резервного копирования. Процесс может работать одновременно с работой пользователей. В таком случае в резервную копию не попадут те данные, которые пользователи добавили после начала процесса резервного копирования;
- операции независимы от ОС, т.е. резервная копия, полученная в Unix, может быть восстановлена из Windows и т.д.
- резервную копию можно сохранять как на диске, так и на ленточном накопителе;
- можно изменить формат файла БД; файл резервной копии Interbase 5.5 можно успешно восстановить в Interbase 6.5.

Резервное копирование

Операция выполняется с помощью диалогового окна Database Backup. Для того чтобы открыть окно, подключитесь к серверу Interbase, затем воспользуйтесь одним из двух способов:

- команда меню Tools | Backup/Restore | Backup, или кнопка на панели инструментов;
- в контекстном меню, появляющемся по правой кнопке мыши выберите команду Backup/Restore | Backup

Появится диалоговое окно Database Backup, представленное на рисунке 2:

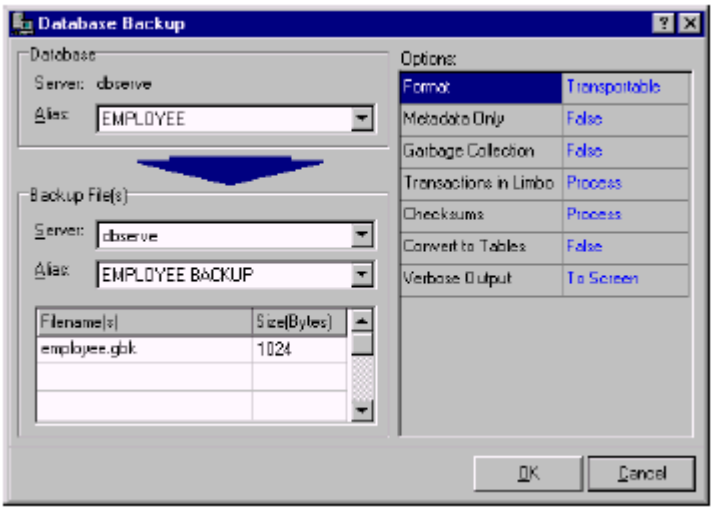

Рисунок 2 – Диалоговое окно Database Backup

Работа с диалоговым окном.

1. Убедитесь, что выбран правильный сервер. Если это не так, закройте окно и вызовите его для другого сервера.
2. Выберите псевдоним БД для резервного копирования. Псевдоним нужен для того, чтобы определить, как называется файл с БД, и где он находится. Если БД состоит из нескольких файлов, заголовочная информация из первого файла указывает, где лежит последующий файл.
3. Выберите сервер, который будет принимать файл с резервной копией.
4. Если резервная копия должна перезаписать существующую БД, выберите псевдоним БД на сервере-приемнике. Если нужно создать новую БД, введите новый псевдоним.
5. Укажите, из каких файлов будет состоять резервная копия. Необходимо ввести имя файла и его размер. Чтобы добавить новую строку в таблицу, находясь на последней строке, нажмите Ctrl-Tab. При указании имени файла можно ввести полный путь к файлу.

В случае если был выбран уже существующий псевдоним, таблица уже содержит список файлов резервной копии. Его можно менять по своему усмотрению.

6. Укажите параметры резервного копирования (Options).
7. Нажмите ОК.

Имена файла БД и ее резервной копии могут быть любыми. Рекомендуется использовать расширение gdb для файлов БД и gbk для резервных копий.

Как правило, файл резервной копии по размеру меньше файла БД, т.к. содержит только текущие версии записей и резервирует меньше места для хранения записи. Также файл резервной копии содержит только объявления индексов, но не сами индексы.

Если указать уже существующее имя, Interbase перезапишет его, не запрашивая подтверждения. Опции отображаются с правой стороны диалога Database Backup. Опции резервного копирования описаны в таблице 9.

Таблица 9 – Опции резервного копирования

Наименование	Варианты	Описание
Format	Transportable, Non-transportable	Transportable используется при переходе с одной ОС на другую.
Metadata Only	True, False	Если выбрать True, будут скопированы только метаданные (структура таблиц, связи между ними, индексы, триггеры и т.д.), но не данные из таблиц. Таким образом, можно получить пустую копию БД. При использовании gbak для той же цели нужно указать ключ – metadata.
Garbage Collection	True, False	По умолчанию во время резервного копирования производится сборка мусора. Сборка мусора помечает старые версии записей как доступные для повторного использования. Отключать данную возможность имеет смысл для ускорения процесса, либо если файл БД поврежден именно в старых версиях записей и Interbase не должен их проверять. Соответствующий ключ для gbak – garbage_collect.
Transactions in Limbo	Process, Ignore	Для того чтобы игнорировать незавершенные (подвисшие) транзакции, нужно установить значение в Ignore. В файл резервной копии не попадают версии записей измененные подобной транзакцией. Записываются версии записей, подтвержденные до начала транзакции. Подвисшие транзакции возникают при сбоях в двухфазном подтверждении транзакции, при сбоях системы, либо во время подготовки обычной транзакции. В случае если есть проблемы с транзакциями, предварительно нужно попробовать восстановить транзакцию с помощью команды меню Tools | Database Maintenance | Transaction Recovery. Соответствующий ключ для gbak– limbo.

Продолжение таблицы 9

Checksums	Process, Ignore	Проверочная сумма есть у каждой страницы данных для контроля ее целостности. Несовпадение проверочной суммы обычно означает физическое повреждение файла. После восстановления необходимо проверить данные. Соответствующий ключ для gbak – ignore.
Convert to Tables	True, False	Преобразует внешние файлы в таблицы. Соответствующий ключ для gbak – convert.
Verbose Output	None, To Screen, To File	Управляет выводом информации о процессе резервного копирования/восстановления. Соответствующий ключ для gbak – verbose.

Восстановление

Операция выполняется с помощью диалогового окна Database Restore. Для того, чтобы открыть окно, подключитесь к серверу Interbase, затем воспользуйтесь одним из двух способов:

- команда меню Tools | Backup/Restore | Restore, или кнопка на панели инструментов;
- в контекстном меню, появляющемся по правой кнопке мыши выберите команду Backup/Restore | Restore

Появится диалоговое окно Database Restore, представленное на рисунке 3:

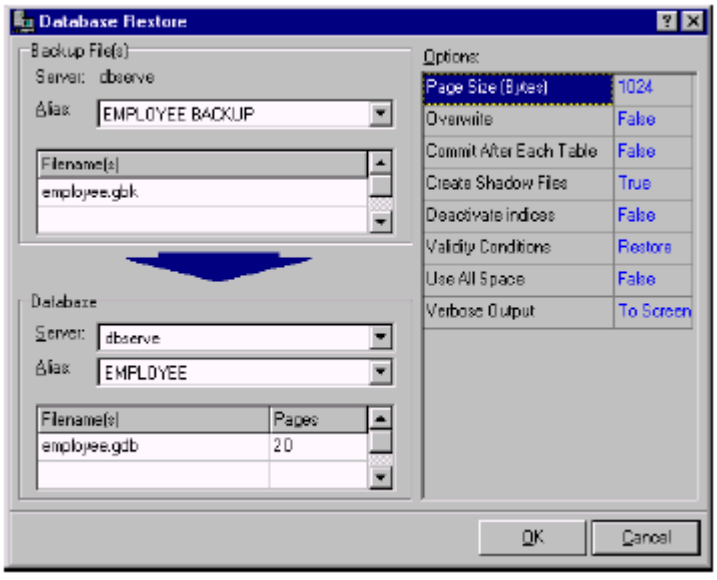

Рисунок 3 – Диалоговое окно восстановления БД

Порядок действий при восстановлении

1. Убедитесь, что сервер с резервными копиями выбран правильно. Если это не так, закройте окно и вызовите его для другого сервера.

2. Выберите псевдоним БД с резервной копией. Псевдоним нужен для того, чтобы определить, как называется файл с резервной копией БД, и где он находится. Если БД состоит из нескольких файлов, заголовочная информация из первого файла указывает, где лежит последующий файл.

3. Если был выбран псевдоним БД с резервной копией, то таблица с именами файлов заполняется автоматически. Можно указать произвольный файл с диска, выбрав Backup File(s) Alias – File… Обязательно нужно указать полный путь к файлу. Обязательно нужно указать все файлы резервной копии. Чтобы добавить новую строку в таблицу, находясь на последней строке, нажмите Ctrl-Tab.

4. Выберите принимающий сервер из списка.

5. Выберите псевдоним БД, в которую будет помещена резервная копия из выпадающего списка.

6. Укажите, в каких файлах должна находиться восстановленная БД. Для больших БД разные файлы целесообразно размещать на разных дисках. Можно добавить новые файлы к БД или удалить существующие, очистив соответствующую строку. Нельзя производить восстановление на сетевые диски.

7. Выберите параметры восстановления.

8. Нажмите OK.

Interbase 6 позволяет успешно завершить процесс восстановления, даже если он по некоторым причинам не сможет построить некоторые индексы (например, из-за недостатка места на диске). В этом случае индексы переводятся в неактивное состояние. После восстановления необходимо командой Alter Index сделать индексы активными.

Принадлежность БД

Несмотря на то, что резервное копирование может быть произведено только владельцем БД или пользователем SYSDBA, восстановление может произвести любой пользователь, если при этом не перезаписывается существующая БД. Восстановленная БД принадлежит пользователю, выполнившему операцию восстановления. Таким образом, резервное копирование и восстановление можно рассматривать как инструмент для смены владельца БД. Резервные копии необходимо хранить в защищенном от несанкционированного доступа месте.

Опции восстановления

Опции отображаются с правой стороны диалога Database Restore. Их описание приведено в таблице 10.

Таблица 10 – Опции операции восстановления

Page Size	1024, 2048, 4096, 8192	По умолчанию – 1024 байтов. Для изменения размера страницы выполните резервное копирование, затем при восстановлении укажите новый размер страницы. Изменение размеров страницы улучшает производительность по следующим причинам: - Запись и чтение полей типа Blob происходит более эффективно, когда поле целиком помещается на одну страницу. Если приложение сохранят множество полей Blob размером более 1К, увеличение размера страницы уменьшит время доступа к Blob. - Производительность выигрывает, если строка таблицы умещается на страницу целиком. - Если в БД есть огромные индексы, увеличение страницы БД уменьшит количество уровней в индексном дереве, что ускорит поиск по дереву. Количество уровней в индексах можно увидеть по команде Tools | Database Maintenance | Statistics. - Если большинство транзакций охватывают только несколько строк данных, меньший размер страницы предпочтительнее. В этом случае пересылаются меньшие объемы данных, и под дисковый кэш используется меньше памяти. Опция для gbak – page_size.
Overwrite	True, False	Если установить этот параметр в False при восстановлении в существующую БД, процесс восстановления не начнется. Нельзя производить восстановление, если к БД есть активные подключения. Лучше всего скопировать файл БД во временный файл, произвести восстановление, и, если все прошло удачно, удалить временный файл. Опция для gbak –replace.
Commit After Each Table	True, False	Обычно Interbase сначала восстанавливает все метаданные, а затем данные. Если установить данный параметр в True, то будет производиться восстановление сначала метаданных таблицы, затем данных таблицы. Опция для gbak -one_at_a_time.
Create Shadow Files	True, False	«Теневые» файлы – это точная копия оригинального файла БД, которая изменяется одновременно с оригиналом.
Deactivate Indexes	True, False	По умолчанию Interbase перестраивает индексы при восстановлении. Если таблицы содержат повторяющиеся значения в уникальном индексе, процесс восстановления завершается с ошибкой. Можно отключить индексы, произвести восстановление, исправить ошибку и снова включить индексы. Отключение индексов ускоряет вставку большого количества строк в таблицу. Отключение и последующее включение индекса заново балансирует дерево индекса. Уникальный индекс не может быть активизирован с помощью ALTER INDEX, его нужно уничтожить и создать заново. Опция для gbak – inactive.

Продолжение таблицы 10

Validity Condition	Restore Ignore	Если ограничения целостности в БД были изменены, старые данные могут не соответствовать новым условиям, и вызовут ошибку при восстановлении. Установив опцию в Ignore, в процессе восстановления будут удалены все ограничения целостности. После восстановления необходимо проверить данные и создать ограничения целостности заново. Опция для gbak – no_validity.
Use All Space	True, False	Для восстановления БД со 100% заполнением страницы (вместо стандартного 80% заполнения) установите опцию в True. Опция для gbak – use_all_space.

2.2. Проверка целостности и восстановление БД

В процессе повседневной работы база данных может быть повреждена. К повреждениям могут привести события, описанные ниже.

- Неправильное завершение пользовательского приложения. Данная причина не может привести к нарушению целостности БД. В случае сбоя приложения Interbase сохраняет данные подтвержденных транзакцией, выполняет откат данных незавершенных транзакций. Если InterBase успел выделить страницы данных для неподтвержденных записей, данные страницы будут рассматриваться как потерянные. Потерянная страница – это ни к кому не относящаяся страница, которая должна быть возвращена в список свободных страниц.

- Ошибки записи ОС и аппаратного обеспечения. Обычно нарушают целостность БД. Ошибки записи могут привести к повреждению структур данных БД. Нарушение структур данных может сделать подтвержденные записи недоступными.

Проверку целостности БД необходимо проводить в следующих случаях:

- если операция резервного копирования/восстановления завершается с ошибкой;

- если пользовательские приложения возвращают ошибку “corrupt database” (повреждение БД);

- периодически, для отслеживания целостности;

- после событий, которые могут привести к нарушению целостности.

Проверка БД требует эксклюзивного доступа к БД. Если с БД работают пользователи, то InterBase вернет ошибку «OBJECT database_name IS IN USE».

Для проверки целостности БД нужно вызвать диалоговое окно Database Validation, показанное на рисунке 4. Для этого надо щелкнуть на псевдониме отсоединенной БД в рабочей панели и выбрать Validation в контекстном меню.

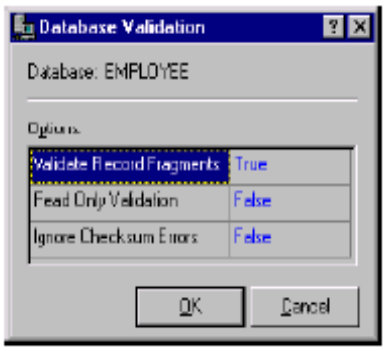

Рисунок 4 – Диалоговое окно проверки целостности БД

Для того чтобы начать процесс верификации, нужно выбрать параметры проверки и нажать на кнопку «ОК». Опции проверки целостности приведены в таблице 11.

При проверке выдаются сообщения о поврежденных структурах данных, об ошибках в выделении страниц и потерянные страницы возвращаются в список свободных.

Таблица 11 – Опции проверки целостности БД

Опция	Назначение
Validate Record Fragments	По умолчанию процесс проверки верифицирует только структуру страниц БД. Если указать данный параметр, будет проверяться как структура страниц, так и структура записей .
Read Only Validation	По умолчанию процесс проверки вносит изменения в БД. Если указать данный параметр, то изменения производиться не будут.
Ignore Checksum Errors	Ошибки контрольной суммы говорят о том, что файл БД был поврежден. Если указать данный параметр, то процесс восстановления завершится успешно, но данные в поврежденных страницах могут быть потеряны.

Даже если удается успешно завершить процесс резервного копирования/восстановления БД с ошибками в контрольной сумме некоторых страниц, повреждение данных, которое повлекли ошибки, бывает трудно обнаружить. Поэтому иногда лучше произвести восстановление с ранней резервной копии и смириться с потерей последних данных.

Если БД содержит ошибки, появляется следующее диалоговое окно:

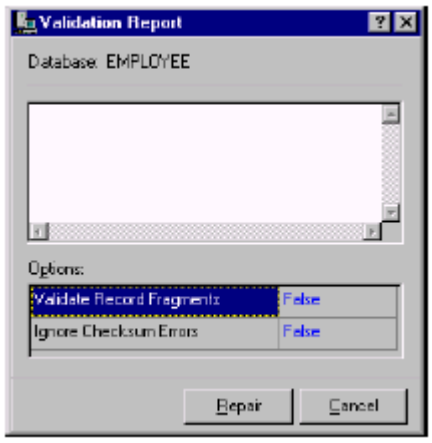

Рисунок 5 – Диалоговое окно с результатами проверки

Обнаруженные ошибки перечисляются в текстовом поле диалогового окна.

Для того чтобы начать процесс восстановления, необходимо выбрать опции восстановления и нажать кнопку “Repair”. В процессе восстановления InterBase помечает поврежденные структуры. При последующем резервном копировании помеченные структуры будут пропущены.

Некоторые ошибки невозможно исправить, например, повреждение стратегических структур (страницы выделения места и др.).

Если предполагается, что БД имеет повреждения, необходимо предпринять следующие шаги:

1. Создать копию файла(ов) БД с помощью команд ОС. Нельзя создавать резервную копию средствами IBConcole или gbak, так как они не работают с поврежденными БД.
2. Выполнить процесс восстановления на копии БД для того, чтобы пометить поврежденные структуры. Если найдены блоки с неправильными контрольными суммами, запустите процесс восстановления заново, установив свойство Ignore Checksum Errors в True. Возможно, потребуется запуск процесса восстановления несколько раз для исправления всех ошибок.
3. Выполнить проверку БД еще раз, с установленной опцией Read Only Validation. Не должно быть сообщений об освобождении страниц, все поврежденные структуры должны быть помечены.
4. Выполнить резервное копирование БД. На этом этапе будут потеряны все данные, ранее помеченные как поврежденные.
5. Выполнить восстановление из полученной резервной копии. Результирующая БД не должна содержать ошибок.

3. Порядок выполнения работы

При выполнении работы используется БД к лабораторной работе № 1.

1. Ознакомиться с теоретическим материалом.
2. Выполнить резервное копирование БД на локальный сервер.
3. Выполнить операцию восстановления из резервной копии.

4. Повредить вручную файл резервной копии, заменив несколько байтов из файла резервной копии по случайному смещению. (Рекомендуется использовать текстовый редактор приложения типа FAR, VC и т.п.).
5. Произвести попытку восстановления из испорченного файла, описать получившиеся ошибки. Попытаться восстановить часть данных, манипулируя параметрами восстановления.
6. Выполнить проверку целостности БД.
7. Физически повредить файл БД.
8. Выполнить проверку целостности и попытаться восстановить данные.

4. Контрольные вопросы

1. Для чего используется резервное копирование и восстановление?
2. Какое расширение используется для резервных копий БД?
3. Опции резервного копирования и их назначение.
4. Опции восстановления и их назначение.
5. Какие пользователи могут выполнять резервное копирование?
6. Какие пользователи могут выполнять восстановление?
7. Для каких целей необходима проверка целостности? Опции проверки целостности и их назначение.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. Ковязин А.Н. Мир Interbase. Архитектура, администрирование и разработка приложений баз данных в Interbase Firebird Yaffil/ А.Н. Ковязин, С.М. Востриков.– Кудиц-Образ, 2006.– 496с.

2. Оптимизация и администрирование БД Microsoft SQL Server 2005: Учебный курс Microsoft/ Перевод. А.А. Исаковский.– BHV, 2007.– 624с.

ПРИЛОЖЕНИЕ А

Варианты заданий

После номера варианта задания в скобках указан номер варианта из лабораторной работы № 4 по курсу ОБД за третий курс. Схему БД надо выбирать по номеру в скобках. В таблицах приведены группы пользователей, права группы пользователей и названия таблиц, на которые распространяются данные права. Если не указано другое, право на таблицу у пользователя – READ (только чтение).

№ 1 (2)

Подразделение	Права	Таблицы
Бюро расписаний	ALL	«Занятие»
Учебный отдел	ALL	«Студент», «Студент_Курс»
Студент	READ	Все

№ 2 (3)

Подразделение	Права	Таблицы
Главный врач	ALL	Операция, Хирург
Хирург	ALL	Препарат_операция, Операция_процедура
Медсестра	ALL	Препарат, Процедура
Пациент	READ	Пациент,Хирург,Операция

№ 3 (5)

Подразделение	Права	Таблицы
Завхоз	READ ALL	Фирма Помещение, Телефон, Типы_Помещений
Менеджер	READ ALL	Фирма Контрагент
Директор	ALL READ	Фирма Все остальные

№ 4 (6)

Подразделение	Права	Таблицы
Курсы повышения квалификации	ALL	Работа, Работа_Курсы, Курс_Обучения
Отдел кадров	ALL READ	Служащий_Работа, Служащий, Тарифная_ставка, Профессия
Министерство	ALL	Тарифная_ставка, Профессия

№ 5 (8)

Подразделение	Права	Таблицы
Деканат	ALL	Факультет
Факультет	ALL	Курс, Студент, Преподаватель,
Студент	READ	Все

№ 6 (9)

Подразделение	Права	Таблицы
Деканат	ALL	ВУЗ, ВУЗ_Специальность, Специальность, Предмет
Приемная комиссия	ALL	Абитуриент, Абитуриент_ВУЗ_Спец, Экзамен
Студент	READ	Абитуриент, Абитуриент_ВУЗ_Спец, Экзамен, Предмет

№ 7 (10)

Подразделение	Права	Таблицы
Транспортный отдел	ALL	Предшествие, Станция
Маршрутный отдел	ALL	Поезд, Рейс
Билетный отдел	ALL	Билет, Пассажир
Пассажир	READ	Предшествие, Станция, Поезд, Рейс

№ 8 (12)

Подразделение	Права	Таблицы
Следователь	INSERT, UPDATE, SELECT	Псевдоним, Персона, Родственник
Следователь	ALL	Дело, Персона_Дело
Комиссар	ALL	Все

№ 9 (13)

Подразделение	Права	Таблицы
Тренер	ALL	Команда, Спортсмен
Министерство	ALL	Соревнование
Судья		Результат_Спортсмен, Результат_соревнований

№ 10 (14)

Подразделение	Права	Таблицы
Филармония	ALL	Концертный_зал, Артист, Концерт
Антерпренер	ALL	Спонсор, Концерт_спонсор, Концерт_артист
Зритель	READ	Концертный_зал, Артист, Концерт, Концерт_артист

№ 11 (17)

Подразделение	Права	Таблицы
Гостиничный трест	ALL	Гостиница, Номер, Фирма
Отдел командировок	ALL	Клиент, Заказ
Клиент	READ	Гостиница, Номер

№ 12 (19)

Подразделение	Права	Таблицы
Завхоз	ALL	МО, Машина, Доп. Комплектующие
Заведующий отделом	ALL	Инженер, Инженер_МО, Инженер_Машина
Директор	ALL	ВЦ

Заказ№…… от «…..» Тираж……экз.
Изд-во СевНТУ