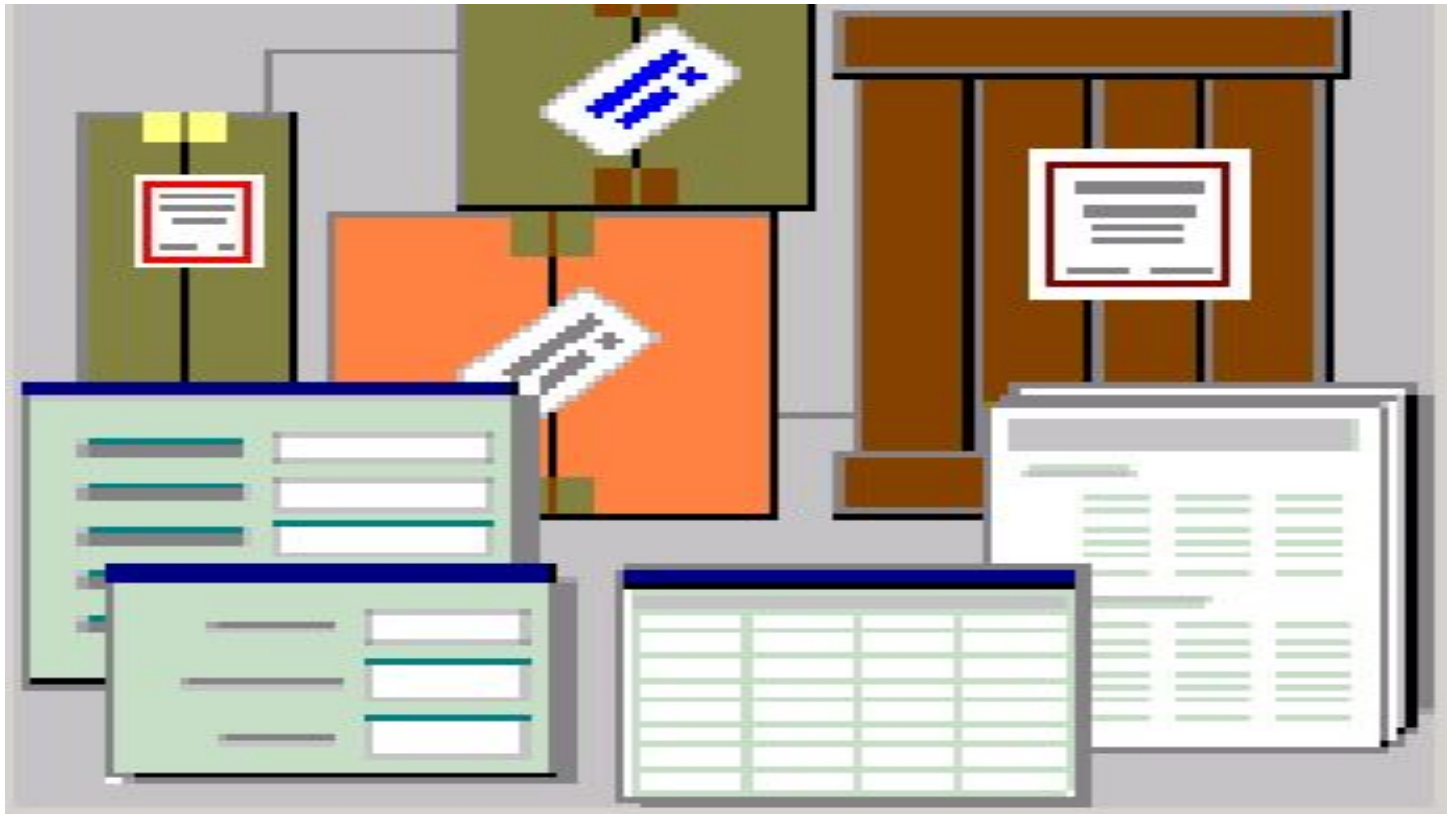
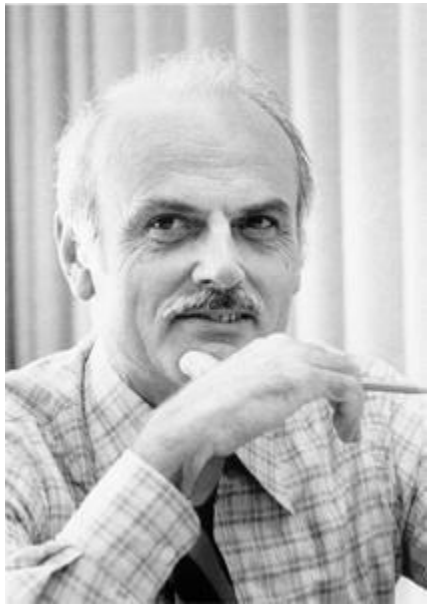
Министерство образования и науки Российской Федерации
Федеральное агентство по образованию
ГОУ ВПО «ОРЕНБУРГСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»
Факультет информационных технологий
Кафедра программного обеспечения вычислительной техники и автоматизированных систем

лекция

Безопасность и целостность баз данных

подготовил
Щелоков Сергей Анатольевич
27-39-72





Эдгар Франк Кодд

Дата рождения:	23 августа23 августа 1923
Место рождения:	Портланд (Дорсет)
Дата смерти:	18 апреля18 апреля 2003 (79 лет)
Гражданство:	Англия
Научная сфера:	информатика
Альма-матер:	Оксфордский университет
Известен как:	Создатель реляционной модели данных.

Введение

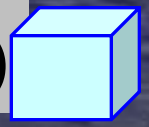
- **<u>Защита данных</u> – исключение возможных случайных или преднамеренных ситуаций, когда существует вероятность потери данных.**

  Способы защиты:

  - **Механизм транзакций;**
  - **Восстановление после сбоев;**
  - **Параллельная обработка данных;**
  - **Блокировка данных.**

- **<u>Безопасность БД</u> – защита БД от несанкционированного разрушения, изменения и модификации.**

- **<u>Целостность БД</u> – обеспечение корректности (исключение аномалий) выполнения операций обработки данных**



1. Безопасность и целостность баз данных

Вопросы безопасности и целостности! - одна из важнейших сторон работы СУБД.

Под **безопасностью** понимают защиту БД от несанкционированного разрушения, изменения и модификации.

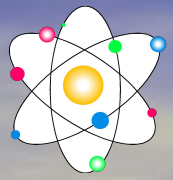
Систему можно считать безопасной только в том случае, если пользователю допускается выполнять только разрешенные действия.

Целостность БД связана с корректным выполнением этих действий.

В СУБД традиционно поддерживаются избирательный или обязательный подходы обеспечения безопасности данных.

При избирательном подходе к управлению безопасностью каждый пользователь обладает различными правами (полномочиями) при работе с тем или иным объектом БД.

В случае обязательного подхода каждому объекту БД присваивается уровень доступа, а пользователям - уровни допуска. Разумеется, для получения доступа к объекту пользователь должен обладать соответствующим уровнем допуска.



Избирательное управление доступом задается правилами, которые должны включать следующее:

******имя правила* **- представляет собой структуру, по которой это правило идентифицируется системой;**

******собственно правила или привилегии* **- набор директив, составляющих способ и возможность доступа, модификации и т п. объектов БД;**

******диапазон применения привилегий*;

******идентификаторы пользователей*, **обладающих вышеперечисленными привилегиями;**

******действие при нарушении правила* **- здесь указывается поведение системы в случае, если пользователь нарушил правило безопасности.**

СОЗДАТЬ ПРАВИЛО БЕЗОПАСНОСТИ RULE1
ДЛЯ МОДИФИКАЦИИ И УДАЛЕНИЯ P
(PN, PNAME, TEACHER, KAFEDRA)
ДЛЯ KAFEDRA = "ФИЗИКА"
ПОЛЬЗОВАТЕЛИ: Ivan, Denis, Andrew
НЕ ВЫПОЛНЯТЬ ПРИ НАРУШЕНИИ ПРАВИЛА

Приведенное выше правило содержит все пять оговоренных элементов. Фактически создано правило безопасности с именем RULE1, позволяющее модификацию и удаление кортежей (PN, PNAME, TEACHER, KAFEDRA) отношения P, относящихся к кафедре физики для пользователей с идентификаторами Ivan, DENIS и Andrew. Если правило будет нарушено (например, запрашивается оговариваемое действие со стороны пользователя Petr), то в запрашиваемом действии будет отказано.

Обязательное управление доступом к БД реализуется при выполнении следующих правил:

- **пользователь имеет возможность работы (но не модификации) с Объектом, если уровень его допуска больше или равен уровню доступа объекта;**

- **пользователь имеет возможность модифицировать объект, если уровень его допуска равен уровню доступа объекта.**

СОЗДАТЬ ПРАВИЛО БЕЗОПАСНОСТИ RULE2
ДЛЯ ПОЛЬЗОВАТЕЛЯ Denis
УСТАНОВИТЬ УРОВЕНЬ ДОПУСКА = 5

А для отношения P, например, вот так:

СОЗДАТЬ ПРАВИЛО БЕЗОПАСНОСТИ RULE3
ДЛЯ МОДИФИКАЦИИ И УДАЛЕНИЯ P
(PN, PNAME, TEACHER, KAFEDRA)
УСТАНОВИТЬ УРОВЕНЬ ДОПУСКА = 5
НЕ ВЫПОЛНЯТЬ ПРИ НАРУШЕНИИ ПРАВИЛА

Тогда пользователь DENIS имеет доступ для модификации и удаления кортежей отношения P, поскольку уровень его допуска и уровень доступа к отношению соответствуют друг другу.

В файле журнала выполняемых операций хранится следующая информация:

- исходный текст запроса;

- имя удаленного терминала, откуда был подан запрос;

- идентификатор пользователя, подавшего запрос;

- дата и время осуществления запроса;

- используемые запросом отношения, кортежи и атрибуты;

- значения данных, с которыми работали до их модификации;

- значения данных, с которыми осуществлялась работа после их модификации;

Другой стороной проблемы безопасности и целостности БД является точность и корректность хранимых в ней данных.

Обычно этот вопрос решают с помощью ограничений целостности.

Традиционно различают два вида ограничений целостности:

немедленно проверяемые

и откладываемые.

При соблюдении обязательного требования поддержания целостности БД возможны следующие уровни изолированности транзакций:

первый уровень - отсутствие потерянных изменении;

второй уровень - отсутствие чтения данных, модифицируемых другой транзакцией;

третий уровень - отсутствие неповторяющихся чтений.

В общем случае ограничение целостности должно содержать три основные части:

- имя ограничения - представляет собой структуру, по которой это ограничение идентифицируется системой;

- собственно ограничения - набор директив и команд, составляющих способ и возможность контроля, и представляющий в конечном итоге логическое выражение. Ограничение удовлетворяется, если оно истинно, и нарушается - если оно ложно;

- действие при нарушении ограничения - здесь предписывается действие системы при нарушении ограничения.

СОЗДАТЬ ОГРАНИЧЕНИЕ ЦЕЛОСТНОСТИ RULE4 ДЛЯ ВСЕХ SP (SP.OCENKA>0 И SP.OCENKA<6) НЕ ВЫПОЛНЯТЬ ПРИ НАРУШЕНИИ ПРАВИЛА

В данном примере ограничение целостности накладывается на атрибут OCENKA отношения SP таким образом, что игнорируются все попытки установить оценку менее 1 и более 5.
Точнее говоря, в ограничении оговорен допустимый интервал оценок (больше 0 и меньше 6).

Различают четыре типа ограничений целостности:

ограничение целостности домена;

ограничение целостности атрибута;

ограничение целостности отношения;

ограничение целостности БД.

СОЗДАТЬ ОГРАНИЧЕНИЕ ЦЕЛОСТНОСТИ RULES ДЛЯ ВСЕХ S(S.KURS >= S'.KURS) НЕ ВЫПОЛНЯТЬ ПРИ НАРУШЕНИИ ПРАВИЛА

Такое ограничение накладывается на атрибут соответственно S'.KURS - до и S.KURS - после выполнения обновления. Приведенный пример ограничивает курс, на котором учится студент так что его можно изменить либо в большую сторону, либо оставить без изменений Действительно - ведь курс не должен уменьшаться. Ограничения состояния и перехода используют только для отношения или БД.

2. Транзакции и параллелизм

Под *транзакцией* понимается неделимая с точки зрения воздействия на БД последовательность операторов манипулирования данными (чтения, удаления, вставки, модификации) такая, что возможны два итога:

- результаты всех операторов, входящих в транзакцию, соответствующим образом отображаются в БД;

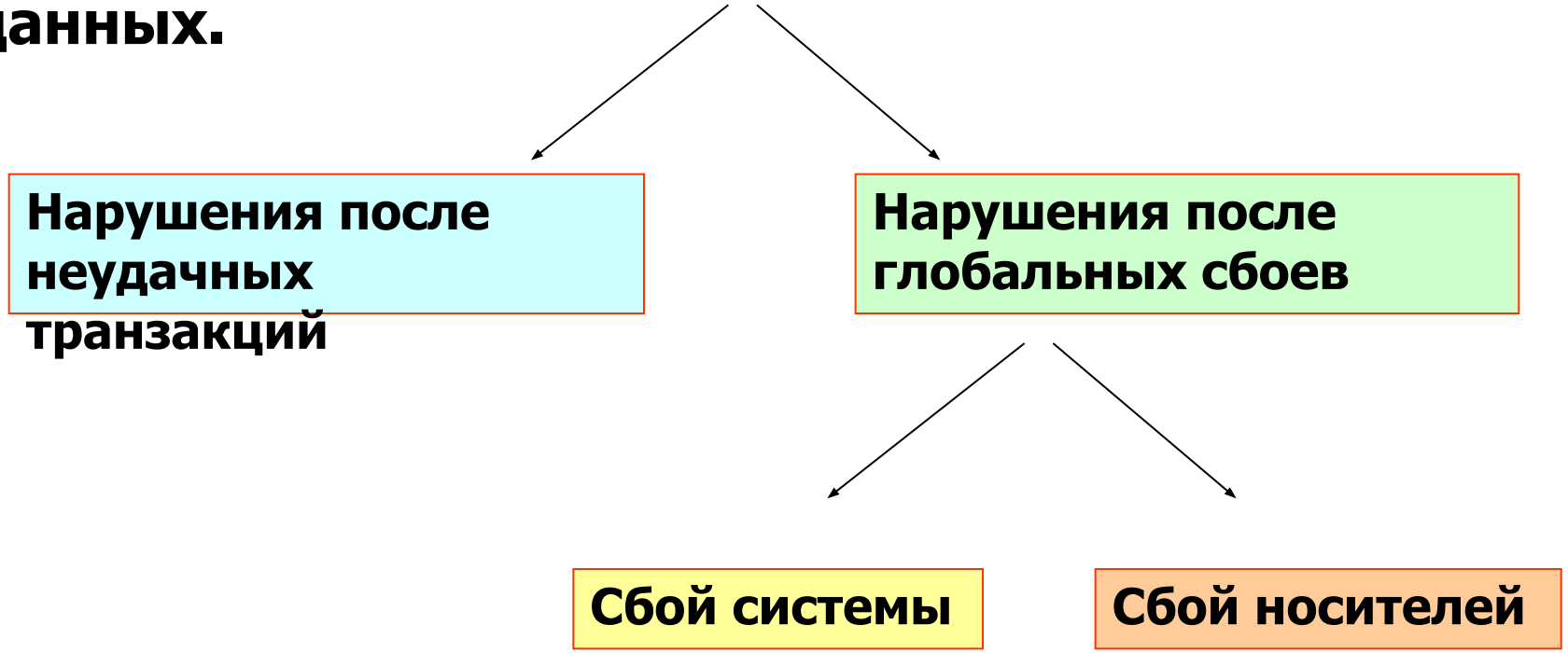
- воздействие всех этих операторов полностью отсутствует.

В СУБД транзакция начинается с оператора BEGIN.

При этом если транзакция завершена оператором COMMIT, то результаты фиксируются во внешней памяти;

при завершении транзакции оператором ROLLBACK результаты отсутствуют во внешней памяти.

Понятие <u>восстановление СУБД</u> - процесс, подразумевающий возвращение БД в правильное состояние, если какой-либо процесс вызвал сбой данных.



Восстановление системы после первого вида глобального сбоя может быть осуществлено по журналу транзакций, в который заносится информация о транзакциях, начавших свое выполнение, и транзакциях, успешно завершившихся Если после перезагрузки системы в журнале будут встречены транзакции, начавшиеся до сбоя, но не закончившиеся, то для всех них выполняется оператор ROLLBACK, в результате чего БД снова будет находиться в целостном состоянии.

Процесс восстановления после сбоя носителей принципиально иной. Восстановление в этом случае осуществляется с резервной копии БД. Понятно, что для реализации этого процесса необходимо, чтобы в СУБД предусматривалось резервное копирование с помощью соответствующей программной реализации.

При параллельной обработке БД возникает три основных проблемы:

- **проблема потери результатов обновления - заключается в**
  первую очередь в том. что транзакция может быть неза-вершена из-за того, что данные, которые она обрабатывает, могут быть модифицированы другой транзакцией;

- **проблема незафиксированности зависимости - состоит в том, что транзакция может использовать для работы данные, которые в настоящий момент модифицируются другой транзакцией. Понятно, что первая из них вполне может работать с данными, которые по завершению второй транзакции в БД просто будут отсутствовать;**

- проблема несовместимого анализа - связана с тем, что в результате модификации БД транзакцией, другая транзакция может внести в БД некую информацию, которая не будет соответствовать целостному состоянию БД.

Для решения этих проблем используют блокировку.

Различают два вида блокировки:

- **блокировка записи - при этом транзакция блокирует кортеж таким образом, что запрос другой транзакции к этому кортежу будет отменен;**

- блокировка чтения - в этом случае транзакция блокирует кортеж так. что запрос со стороны другой транзакции на блокировку записи этого кортежа будет отвергнут, а на блокировку чтения - принят.

В СУБД используют протокол доступа к данным, позволяющий избежать проблем параллелизма. Его суть заключается в следующем:

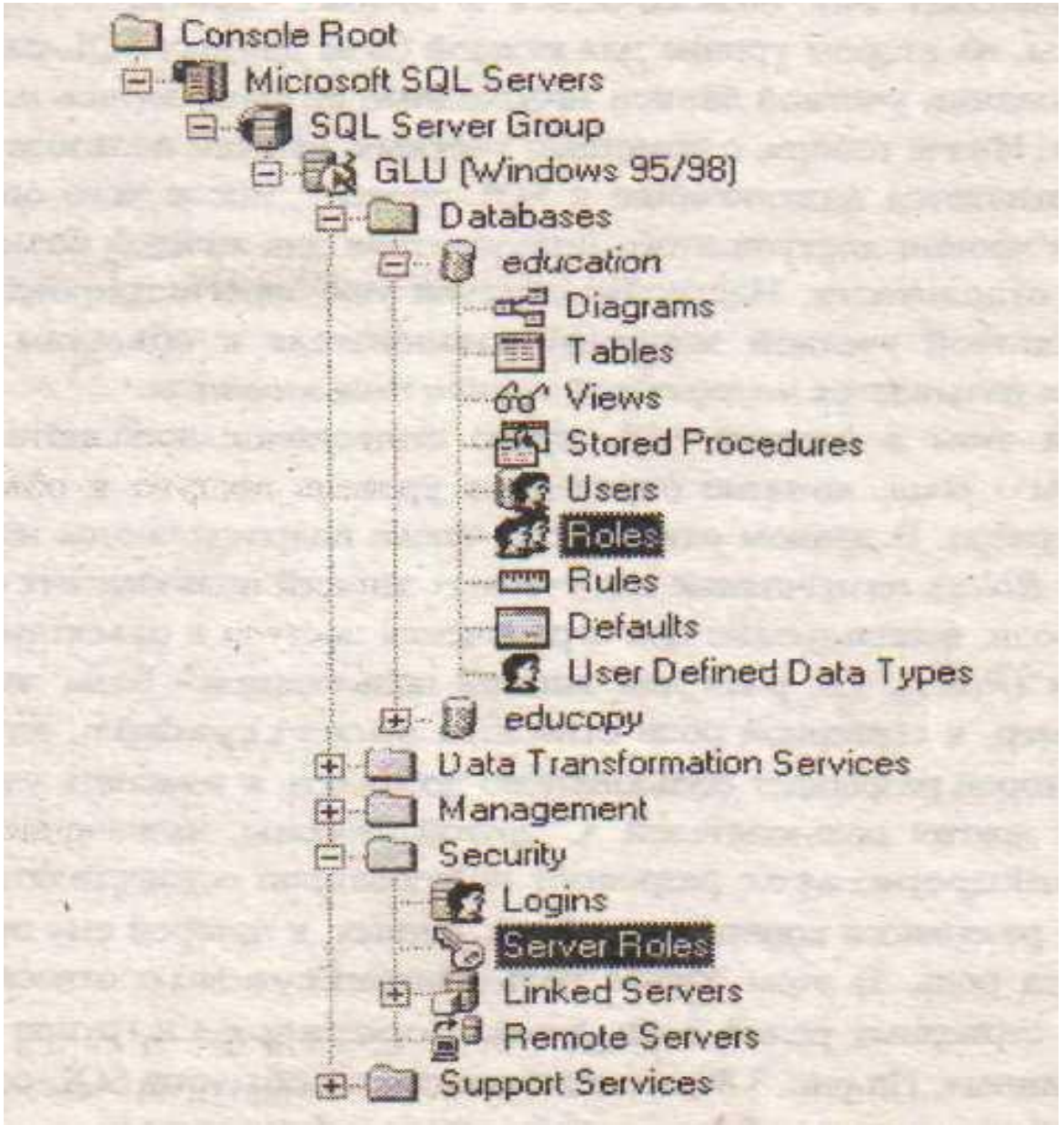
- транзакция, результатом действия которой на кортеж является его извлечение, обязана наложить блокировку чтения на этот кортеж;

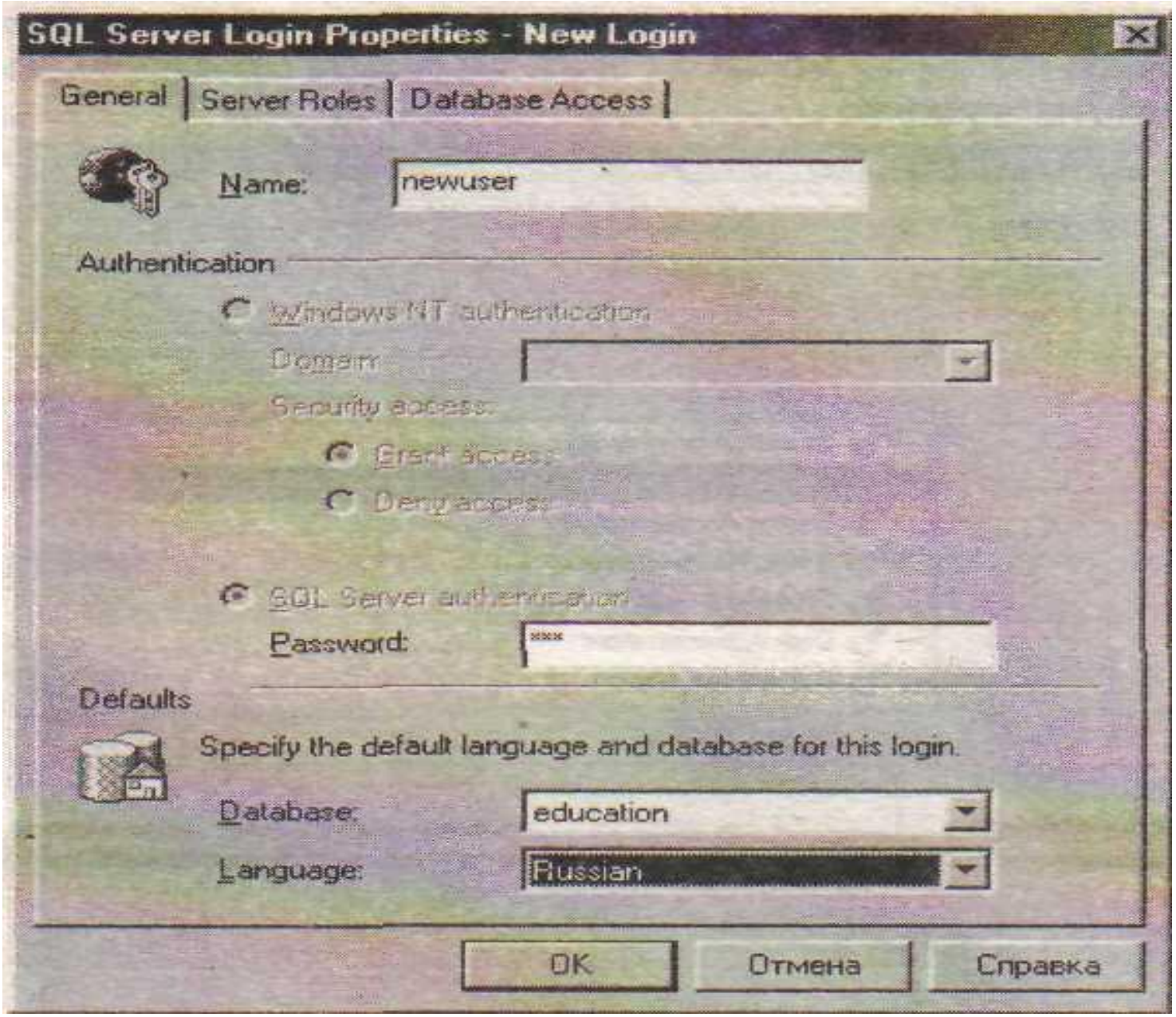
- транзакция, предназначенная для модификации кортежа, обязана наложить блокировку записи на этот кортеж;

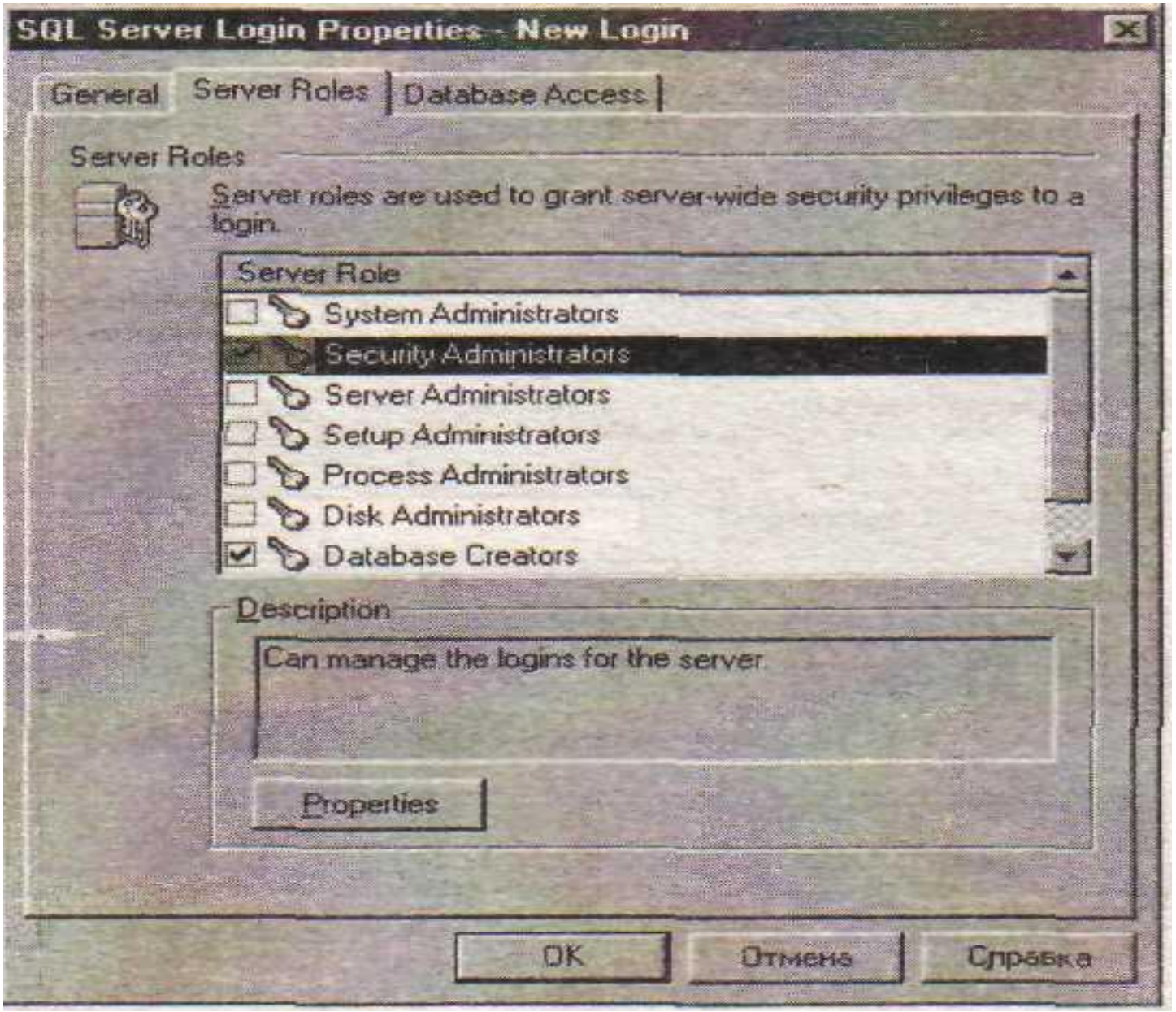
- в случае, если запрашиваемая блокировка на кортеж отвергается из-за того, что на кортеж уже наложена блокировка, то транзакция переводится в режим ожидания до тех пор, пока блокировка не будет снята;

- блокировка записи сохраняется вплоть до конца выполнения транзакции, то есть до выполнения операторов COMMIT или ROLLBACK.

3. Методы защиты данных в SQL сервере




SQL Server Login Properties - New Login
General
Server Roles
Database Access
Name:
newuser
Authentication
Windows NT authentication
Domain:
Security access:
Grant access
Deny access
SQL Server authentication
Password:
***
Defaults
Specify the default language and database for this login.
Database:
education
Language:
Russian
OK
Отмена
Справка


SQL Server Login Properties - New Login
General
Server Roles
Database Access
Server Roles
Server roles are used to grant server-wide security privileges to a login.
Server Role
System Administrators
Security Administrators
Server Administrators
Setup Administrators
Process Administrators
Disk Administrators
Database Creators
Description
Can manage the logins for the server.
Properties
OK
Отмена
Справка

Ниже представлено краткое описание возможностей серверных ролей:

System Administrators - выполнение любых функций администрирования SQL-сервера;

Security Administrators - управление доступом, возможность создания баз данных, доступ к log-файлу ошибок;

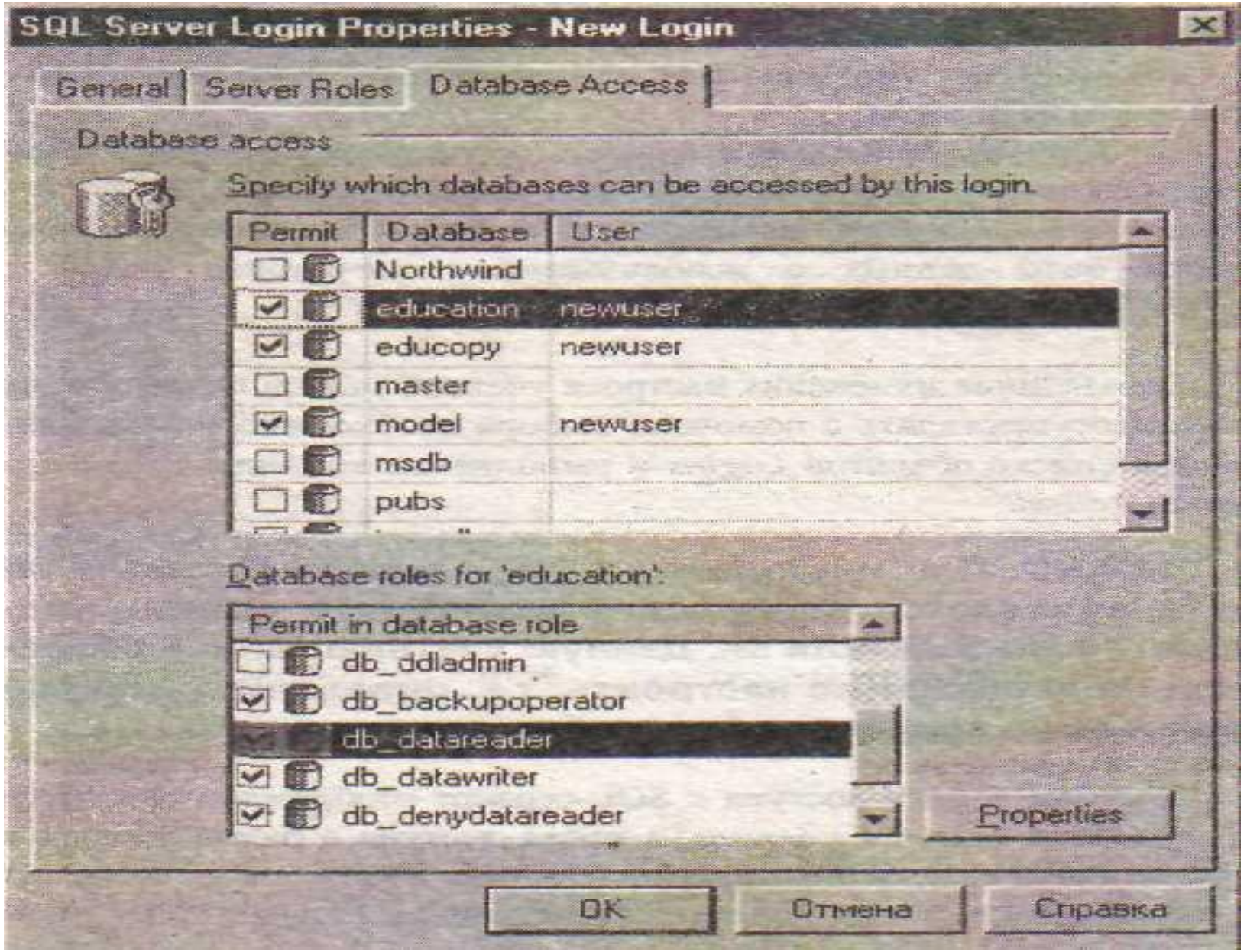
Server Administrators - настройка конфигурации и выполнение функций закрытия SQL-сервера;

Setup Administrators - управление связями между серверами и их процедурами запуска;

Process Administrators - управление процессами, выполняющимися в SQL-сервере;

Disk Administrators - управление файлами SQL-сервера;

Database Creator - управление процессами создания и удаления баз данных.


SQL Server Login Properties - New Login
General
Server Roles
Database Access
Database access
Specify which databases can be accessed by this login.
Permit
Database
User
Northwind
education
newuser
educopy
newuser
master
model
newuser
msdb
pubs
Database roles for 'education':
Permit in database role
db_ddladmin
db_backupoperator
db_datareader
db_datawriter
db_denydatareader
Properties
OK
Отмена
Справка

public - пользователи бет предоставления специальных ролей обладают ролью доступа public;

db_owner - полный доступ к базе данных;

db_accessadmin - возможность добавления и удаления пользователей (не путать с учетными записями севера) в базу данных;

db_securityadmin - возможность управления всеми процессами доступа пользователей к объектам базы данных:

db_ddiadmin - выполнение всех команд DDL (Data definition language - язык определений), кроме GRANT. REVOKE или DENY;

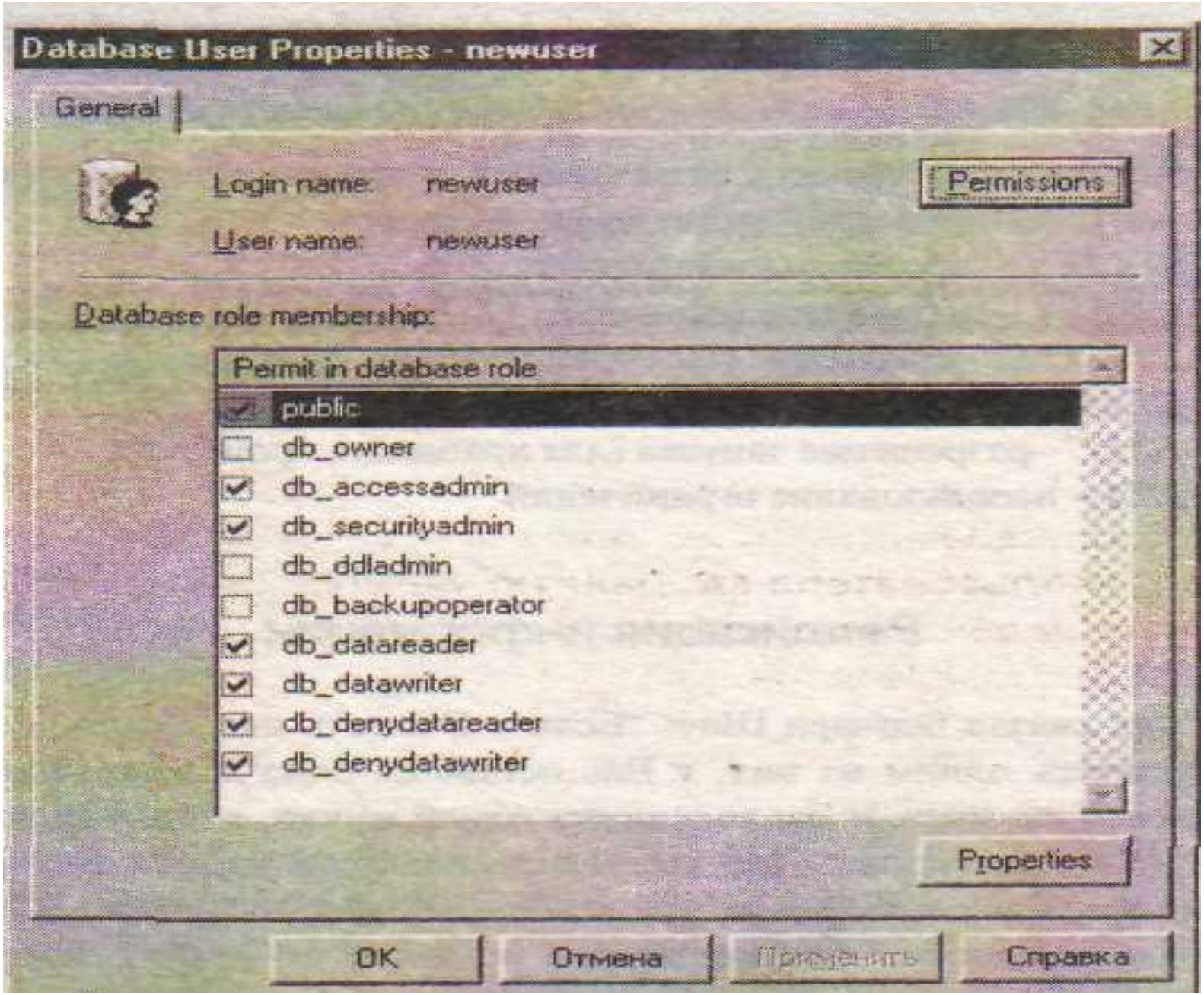
db_backupoperator - функции запуск процедуры резервного копирования базы данных;

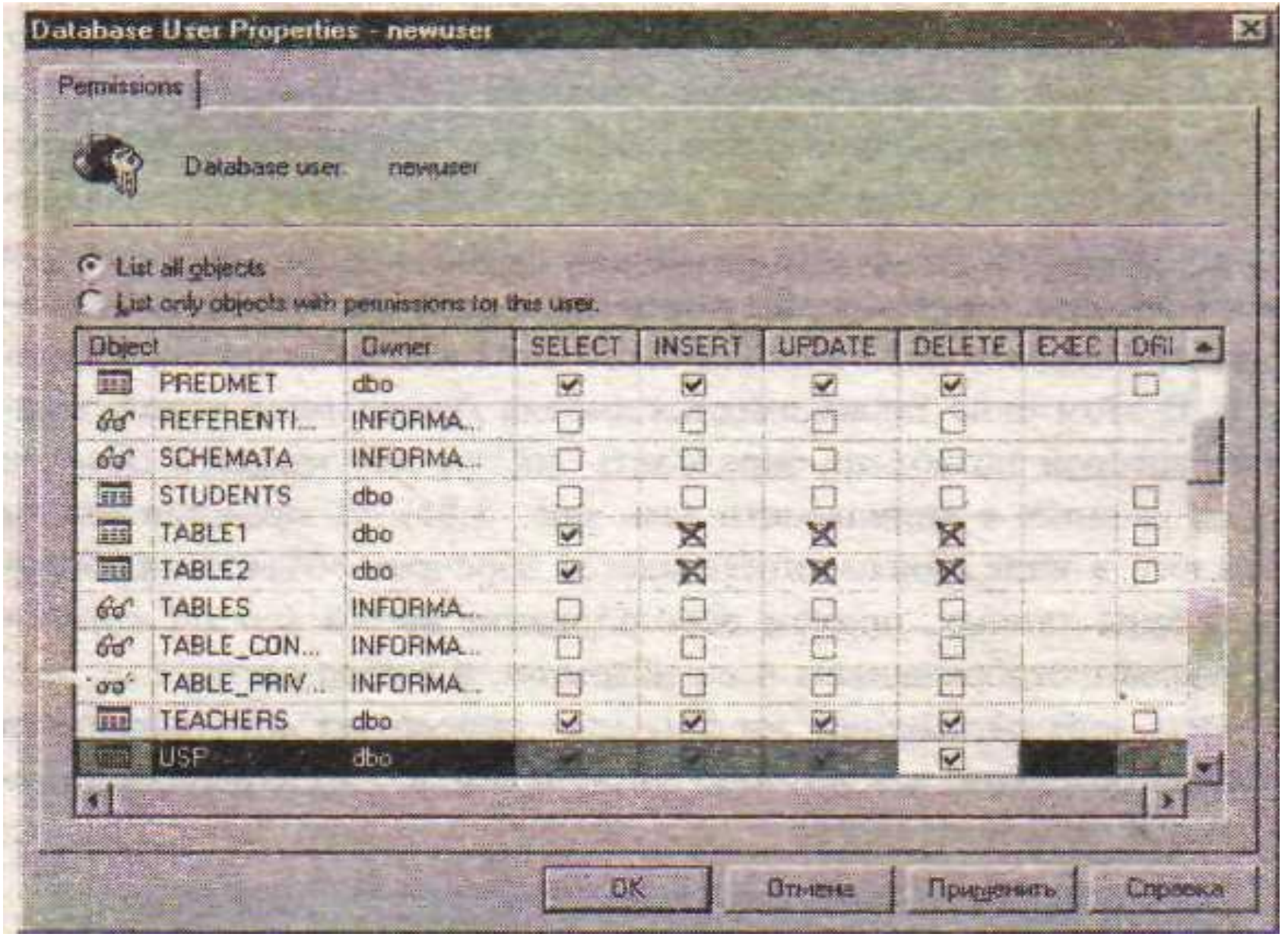
db_datareader - возможность чтения всех данных из любых таблиц базы данных;

db_dalawriter - возможность изменения всех данных из любых таблиц базы данных;

db_denydalareader - возможность ограничения доступа к объектам базы данных с использованием оператора SELECT;

db_denydatuwriter - возможность ограничения доступа к объектам базы данных с использованием операторов INSERT. UPDATE и DELETE.


Database User Properties - newuser
General
Login name: newuser
User name: newuser
Permissions
Database role membership:
Permit in database role
public
db_owner
db_accessadmin
db_securityadmin
db_ddladmin
db_backupoperator
db_datareader
db_datawriter
db_denydatareader
db_denydatawriter
Properties
OK
Отмена
Применить
Справка


Database User Properties - newuser
Permissions
Database user: newuser
List all objects
List only objects with permissions for this user.
Object
Owner
SELECT
INSERT
UPDATE
DELETE
EXEC
DRI
PREDMET dbo
REFERENTI... INFORMA...
SCHEMATA INFORMA...
STUDENTS dbo
TABLE1 dbo
TABLE2 dbo
TABLES INFORMA...
TABLE_CON... INFORMA...
TABLE_PRIV... INFORMA...
TEACHERS dbo
USP dbo
OK
Отмена
Применить
Справка