

МОБИЛЬНАЯ  
РЕЛЯЦИОННАЯ  
СУБД **ЛИНТЕР**<sup>®</sup>

Linter Standard  
Linter Bastion  
Linter RealTime  
Linter Multiversion

**Администрирование средств  
защиты данных**

НАУЧНО-ПРОИЗВОДСТВЕННОЕ ПРЕДПРИЯТИЕ

---

 **РЕЛАКС**<sup>®</sup>

## **Товарные знаки**

РЕЛЭКС™, ЛИНТЕР® , НЕВОД® , LAV™, ЛАКУНА являются товарными знаками, принадлежащими ЗАО НПП «Реляционные экспертные системы» (далее по тексту – компания РЕЛЭКС). Прочие названия и обозначения продуктов являются товарными знаками их производителей, продавцов или разработчиков.

## **Интеллектуальная собственность**

Правообладателем продуктов ЛИНТЕР®, НЕВОД®, LAV™, ЛАКУНА является компания РЕЛЭКС (1990–2011). Все права защищены. Данный документ является собственностью компании РЕЛЭКС. Ни одна его часть не может быть воспроизведена, передана, преобразована, сохранена в системе поиска информации, переведена на другой язык или компьютерный язык в какой-либо форме, какими-либо средствами, электронными, механическими, магнитными, оптическими, химическими, ручными или иными, без предварительного разрешения компании РЕЛЭКС.

## **О документе**

Материал, содержащийся в данном документе, прошел тщательную проверку, но компания РЕЛЭКС не гарантирует, что документ не содержит ошибок и пропусков. Компания РЕЛЭКС оставляет за собой право в любое время вносить в документ исправления и изменения, пересматривать и обновлять содержащуюся в нем информацию.

## **Адрес**

394006, г. Воронеж, ул. 20-летия Октября, 119.  
Тел./факс: (473) 2-711–711, 2-778–333.  
e-mail: market@relex.ru.

## **Адрес для корреспонденции**

394000, г. Воронеж, а/я 137.

## **Техническая поддержка**

Отдел поддержки и сопровождения программных продуктов:

телефон: (473) 2-711–711 с 9:00 до 18:00 мск.  
e-mail: support@relex.ru, market@relex.ru.

С целью повышения качества разрабатываемых программных средств и предоставляемых услуг в компании РЕЛЭКС действует автоматизированная система учёта и обработки рекламаций. Обо всех обнаруженных недостатках и ошибках в программном продукте и/или документации на него просим сообщать нам на Internet–странице [рекламация](#).

# Оглавление

<b>Предисловие</b> .....	<b>1</b>
Назначение документа .....	1
Для кого предназначен документ .....	1
Принятые обозначения и соглашения .....	1
Дополнительные документы .....	2
<b>Архитектура комплекса средств защиты данных</b> .....	<b>3</b>
Основные принципы .....	3
Открытость .....	3
Идентификация и аутентификация .....	4
Контроль доступа в процессе обработки запроса .....	5
Прямая передача прав .....	5
Косвенная передача прав .....	6
Регистрация действий .....	6
Метка доступа .....	6
Метка объекта .....	7
Метка субъекта .....	7
Схема хранения данных .....	8
Категория доступа субъекта контроля .....	9
Роли .....	9
Управление привилегиями доступа .....	9
Группы пользователей .....	9
Уровни доступа .....	11
Управление уровнем доступа пользователей .....	12
Управление уровнем доступа к таблице .....	13
Управление уровнем доступа к столбцу таблицы .....	13
Представление групп и уровней в SQL-выражениях .....	13
Получение информации о метках доступа .....	13
SELECT-выражения .....	14
UPDATE и INSERT операторы .....	14
Управление мандатной защитой на уровне сессии .....	15
Маркировка документов .....	16
Контроль доступа к БД с рабочих станций .....	16
Защита ввода-вывода на внешний носитель .....	20
Контроль целостности средств защиты данных .....	22
Подсчет контрольной суммы .....	23
Диспетчер доступа .....	24
Создание БД .....	24
Системная БД .....	24
Структура системных таблиц .....	25
Рабочая БД .....	26
Инициализация комплекса средств защиты данных .....	27
<b>Мониторинг комплекса средств защиты данных</b> .....	<b>28</b>

## Оглавление

---

Управление мониторингом.....	31
Системные события .....	33
События, связанные с БД.....	33
События, связанные с подсистемой доступа.....	35
События, связанные с таблицами.....	35
События, связанные с пользователями .....	36
<b>Механизм надежного восстановления .....</b>	<b>37</b>
<b>Преобразование данных .....</b>	<b>38</b>
<b>Очистка оперативной/внешней памяти .....</b>	<b>39</b>
<b>Изоляция модулей.....</b>	<b>40</b>
<b>SQL-запросы для работы с системой защиты данных.....</b>	<b>41</b>
Именованые объекты БД.....	41
Предоставление привилегий .....	41
Отмена привилегий .....	42
Создание/удаление пользователя .....	42
Изменение пароля пользователя.....	43
Создание/удаление роли.....	43
Назначение/отмена назначения роли.....	44
<b>Приложение. Синтаксис команд для работы с комплексом средств защиты данных</b>	<b>45</b>
<b>Указатель команд SQL-запросов .....</b>	<b>49</b>

# Предисловие

## Назначение документа

Документ содержит описание процедур администрирования комплекса средств защиты информации (КСЗ) от несанкционированного доступа, описание контролируемых функций и рекомендации по созданию систем защиты:

- описание контролируемых функций;
- руководство по генерации КСЗ;
- описание старта системы и процедур проверки правильности старта;
- описание процедур работы со средствами регистрации;
- руководство по средствам надежного восстановления;
- руководство по работе со средствами контроля модификации;
- руководство по работе со средствами контроля дистрибуции.



Документ может использоваться для работы с любой версией СУБД ЛИНТЕР. Особенности конкретных версий оговариваются по тексту.

## Для кого предназначен документ

Документ предназначен для администратора безопасности системы, а также для администратора защиты.

## Принятые обозначения и соглашения

<u>Обозначение</u>	<u>Пример</u>	<u>Значение</u>
Курсив	<i>Растровым</i> называется изображение...	Новый термин в тексте.
Полужирный шрифт	В этом случае необходимо переносить <b>все</b> физические файлы.	Выделение в тексте.
Подчеркнутый шрифт	Подробную информацию о работе программы можно получить на сайте <a href="http://www.dmk.ru">www.dmk.ru</a> .	Адреса страниц Internet.
Текст, разделенный знаком ⇒	Выполните команду <b>View ⇒ Properties</b> (Вид ⇒ Свойства).	Последовательность выполнения команд.
Текст, заключенный в <>, со знаком + между ними	<Ctrl>+<C>	В <> заключаются клавиши клавиатуры, знак + означает сочетание клавиш.
Крупный моноширинный текст	SQL> _q	Текст командной строки.
Мелкий моноширинный текст	Page Time Count	Текст программы.
Заглавные буквы	BROWSE	Названия команд, слова, зарезервированные в SQL, ключевые слова.

<u>Обозначение</u>	<u>Пример</u>	<u>Значение</u>
Курсив в <>	<return statement>	Определяемый элемент синтаксической конструкции.
Символ ::=		Равенство по определению. Слева от знака стоит определяемое понятие, справа – собственно определение понятия.
Квадратные скобки [ ]	DBSTORE [-d -n -o -p -r -t -u]	Необязательные элементы конструкции. В данном примере ключи не являются обязательными элементами команды.
Вертикальная черта	<return value> ::= <value expression>   <b>NULL</b>	Указывает на то, что все предшествующие ей элементы списка являются необязательными и могут быть заменены любым другим элементом списка после этой черты.
Фигурные скобки { }	CODEPAGE {866   1251   KOI8}	Указывают на то, что все, находящееся внутри них, является единым целым.
Многоточие «...»	Характеристики столбца MAKE CHAR(20) MODEL CHAR(20) ... SQL>	Означает, что предшествующая часть может быть повторена любое количество раз.
Многоточие, внутри которого находится запятая «,...»		Указывает на то, что предшествующая часть оператора, состоящая из нескольких элементов, разделенных запятыми, может иметь произвольное число повторений.
Текст со знаком  на сером фоне	 Если конфигурация страницы-шаблона не учитывала свойств, команда будет выполнена некорректно.	Примечание.

## Дополнительные документы

- СУБД ЛИНТЕР. Модель защиты данных.
- СУБД ЛИНТЕР. Справочник по SQL.

# Архитектура комплекса средств защиты данных

Архитектура комплекса средств защиты данных (КСЗ) включает в себя как информационные компоненты (описания объектов и субъектов в БД), так и программные компоненты (программный КСЗ ядра СУБД ЛИНТЕР).

## Основные принципы

КСЗ СУБД ЛИНТЕР строится на 9 основных принципах:

- 1) открытость;
- 2) идентификация и аутентификация – определяет начало сеанса работы субъекта;
- 3) права на доступ проверяются на всех этапах:
  - на этапе трансляции запроса;
  - на этапе обработки запроса;
  - при обращении к объекту.
- 4) прямая передача прав – прерогатива владельца объекта;
- 5) косвенная передача прав – прерогатива субъекта-администратора;
- 6) все действия, связанные с участием КСЗ, подлежат обязательному протоколированию;
- 7) все объекты контроля снабжаются метками доступа, включающими уровни конфиденциальности и принадлежность к группе субъектов; субъекты разбиты на непересекающиеся группы, каждая из которых изолирована по доступу от всех прочих групп;
- 8) метки объектов (физическая защита) есть неизменная величина, в отличие от меток субъектов (их можно переводить из группы в группу, менять уровень доступа и пр.);
- 9) изменение меток доступа субъектов – прерогатива субъекта (администратора безопасности).

Ниже эти принципы разбираются более подробно.

## Открытость

СУБД ЛИНТЕР – открытая система. По технологии построения открытых систем все ее алгоритмы открыты для всех пользователей и работают для всех одинаково.

Алгоритм интерфейса нижнего уровня устроен таким образом, что установленная связь субъекта и СУБД ЛИНТЕР идентифицируется еще и идентификатором прикладной задачи (каждая операционная система присваивает задаче уникальный идентификатор) (см. стр. 4, «Идентификация и аутентификация»), что исключает возможность перехвата ответов от ядра ЛИНТЕР. Это означает, что ответ, посланный задаче на ее запрос, придет именно к этой задаче и ни к какой другой.

При этом прикладные пользовательские программы отделены от системы. Общение между ядром СУБД ЛИНТЕР и приложением, ее использующим, проходит только через интерфейс нижнего уровня СУБД. Тексты/алгоритмы интерфейса нижнего уровня открыты и не содержат элементов КСЗ.

Все утилиты СУБД ЛИНТЕР также написаны с помощью интерфейса нижнего уровня (т.е. штатных средств) и не используют никаких других скрытых особенностей СУБД.

То же самое можно сказать и о прочих программных интерфейсах, присутствующих в СУБД (например, встроенный SQL (Pci)). В их основе также лежит интерфейс нижнего уровня СУБД ЛИНТЕР.

Это налагает на СУБД дополнительные требования, особенно в части КСЗ.

## Идентификация и аутентификация

Информация о каждом пользователе, имеющем право подключаться к СУБД, находится в системных справочниках БД или в системной БД. Правом изменения системной БД обладает только администратор защиты СУБД.

Каждый пользователь (субъект) имеет в системной БД уникальный идентификатор - идентификатор субъекта.

СУБД связывает субъекта с его идентификатором, исходя из имени и пароля, полученных в результате идентификации и аутентификации.

После этого на всем протяжении работы (до отсоединения пользователя от СУБД) ядру ЛИНТЕР уже не требуется знание имени и пароля субъекта. Для алгоритмов доступа достаточно только идентификатора субъекта.

При хранении информации о правах субъекта (см. стр. 8, «Схема хранения данных») ЛИНТЕР использует идентификатор субъекта, а также идентификаторы объектов, к которым субъект имеет доступ.

Для каждого установленного канала связи с приложением СУБД хранит у себя в памяти следующую информацию:

- идентификатор субъекта;
- идентификатор приложения (процесса и даже нити);
- адрес клиентской станции и тип сетевого протокола (при сетевой работе).

Таким образом, сеанс работы субъекта начинается с идентификации и аутентификации субъекта, при этом СУБД связывает последнего с его идентификатором. Данная связь прекращается только после отсоединения субъекта от СУБД.

Кроме того, СУБД особенно важно, чтобы ответы на все запросы приложения были получены только этим приложением и никаким другим. Это реализуется с помощью идентификатора приложения, который присваивается ему операционной системой в момент запуска. При установлении связи приложение, используя интерфейс нижнего уровня, сообщает ядру ЛИНТЕР свой идентификатор (а при сетевой работе и адрес станции), так что, посылая приложению ответы, СУБД владеет полной информацией о том, кому можно пересылать полный адрес нужного приложения.

В случае, когда исчезает хотя бы один из компонентов связи (удаляется субъект, падает приложение, отключается клиентская станция), ядро, следуя установленным тайм-аутам, разрывает образованную связь.

При попытке пользователя открыть канал связи, СУБД проверяет **имя** пользователя и его **пароль**.

При этом возможны следующие ситуации:

- имя и пароль пользователя введены правильно – пользователь допускается к БД;
- имя пользователя введено неверно – выдается сообщение о том, что имя пользователя введено неверно (**Illegal user name**). Пользователь к БД не допускается;
- пароль пользователя введен неверно – выдается сообщение о том, что пароль пользователя введен неверно (**Illegal password**). Пользователь к БД не допускается.

## Контроль доступа в процессе обработки запроса

Обработка SQL-запроса состоит из нескольких этапов:

- 1) трансляция запроса и привязка его элементов к физическим объектам БД;
- 2) выполнение запроса;
- 3) передача результата выполнения запроса пользователю.

Проверка прав доступа выполняется на каждом этапе.

На первом этапе проверяется, в основном, логическая защита на самом высоком (общем) уровне.

Второй этап, в свою очередь, также состоит из нескольких этапов. При этом объекты, участвующие в запросе, последовательно вовлекаются в процесс обработки.

В СУБД ЛИНТЕР принято, что **права доступа проверяются только в момент привлечения очередного объекта к процессу обработки запроса**. Таким образом, СУБД может потратить довольно продолжительное время на обработку запроса, пока не определит, что она невозможна.

С другой стороны, если в процессе обработки запроса кто-то изменяет права субъекта, то данные изменения могут сказаться уже на обработке текущего запроса. По крайней мере, при таком подходе возрастает вероятность того, что СУБД примет во внимание изменение прав доступа.

Важность того, что проверка доступа перенесена в этап обработки, обусловлена освобождением первых двух этапов от дополнительных проверок. Это позволяет прикладным программам свободно использовать уже оттранслированные и привязанные запросы.

## Прямая передача прав

В СУБД ЛИНТЕР в качестве одного из основополагающих принят принцип прямой передачи прав.

Для того чтобы исключить возможность неконтролируемого распространения прав, принят следующий принцип: **прямая передача прав на объект разрешена только для владельца данного объекта**.

Субъект, получивший какие-либо права на чужой объект, не может осуществить прямую передачу прав.

### Косвенная передача прав

Косвенная передача/отмена прав на объект возможна только через аппарат ролей.

В случае косвенной работы роль должна быть наполнена соответствующими возможностями, предоставленными напрямую владельцами объектов.

После наполнения роли только администратор может назначать ее другим пользователям. Пользователи уже не смогут осуществить передачу имеющихся у них прав.

### Регистрация действий

Все действия приложений, связанные с:

- идентификацией и аутентификацией;
- запросами на доступ к объектам защиты;
- созданием/уничтожением объектов защиты;
- действиями по изменению ПРД,

регистрируются в таблице \$\$\$AUDIT. Кроме того, в данную таблицу заносятся внутренние ошибки СУБД ЛИНТЕР (по требованию), изменения состояния пользовательских событий, установленных в системе, и т.д.

Таблица регистрации \$\$\$AUDIT создается в процессе инициализации подсистемы расширенных функций КСЗ СУБД ЛИНТЕР (см. стр. 27, «Инициализация комплекса средств защиты данных»). Ее наличие – сигнал для включения протоколирования. Более подробно о таблице регистрации изложено на стр. 28.

Для включения протоколирования применяется SQL-запрос:

```
AUDIT START;
```

При этом в нулевой кортеже устанавливается флаг активности подсистемы audit. Для выполнения команды необходимо иметь права DBA.

Останов протоколирования выполняется с помощью SQL-запроса:

```
AUDIT STOP;
```

В нулевой кортеж заносится информация о прекращении протоколирования.

При его возобновлении все события, установленные до остановки протоколирования, будут активизированы.

Для выполнения команды необходимо иметь права DBA.

### Метка доступа

Всем объектам контроля присваивается **метка доступа**.

Метки доступа используются СУБД ЛИНТЕР для проверки доступности информации, а также для возможности организации принудительного управления доступом.

Механизм меток доступа – главное звено обеспечения мандатного принципа контроля доступа (подробнее об этом см. стр. **Ошибка! Закладка не определена.**, «**Ошибка! Источник ссылки не найден.**»).

Метка доступа содержит информацию об уровне конфиденциальности (для объектов, т.е. данных) – **метку объекта**, и уровне доступа (для пользователей, субъектов) – **метку субъекта**.

Кроме иерархии уровней в метке содержится также информация о **группе**. Речь идет о группах пользователей, которые должны быть изолированы друг от друга (в смысле информации).

Соответственно группам пользователей объекты данных тоже разбиваются на группы.

Таким образом, метки предоставляют как иерархию доступа (уровни), так и горизонтальное деление (объектов, субъектов).

Метками доступа снабжается информация всех уровней – от таблицы, до столбца и записи и даже значения полей записи. Т.е. в ЛИНТЕР есть возможность снабжать все объекты доступа специальными метками (об их содержании упоминается ниже).

Все субъекты доступа также снабжаются метками. При проверке доступа субъекта к конкретному объекту СУБД осуществляет дополнительную проверку, не выполняя недоступные действия.

### Метка объекта

Метка объекта обеспечивают физическую защиту данных.

Метка объекта включает:

- группу субъекта, который внес данный объект;
- уровень доступа на чтение: RAL-уровень (Read Access Level);
- уровень доступа на запись: WAL-уровень (Write Access Level).

В отличие от меток субъектов (субъектов можно переводить из группы в группу, менять им уровень доступа и пр.) мир физической защиты данных – неизменный мир.

Появление в БД нового объекта контроля сопровождается присвоением ему соответствующей метки объекта, которая будет связана с ним до момента его уничтожения.

Даже перемещение (копирование) объектов (например, из одной таблицы в другую) не изменит их защитных атрибутов (по крайней мере, не снизит). Так, изменить метку записи таблицы можно только при замене всей информации этой записи, что, собственно, равносильно ее удалению и добавлению вновь.

### Метка субъекта

Метка субъекта включает:

- группу, к которой принадлежит субъект;
- RAL-уровень субъекта, который представляет собой максимальный RAL-уровень доступной субъекту информации;

- WAL-уровень субъекта, т.е. минимальный RAL-уровень объекта, который может быть создан данным субъектом.

В отличие от меток объектов, метки субъектов (пользователей) могут быть изменены. Изменение меток доступа субъектов – прерогатива субъекта (администратора безопасности).

Он может перевести субъекта из группы в группу, изменить уровни доступа и доверия. Естественно, что изменение возможностей субъекта не должно оставаться незамеченным – все подобные действия **протоколируются** в таблице \$\$\$AUDIT.

## Схема хранения данных

Информация обо всех субъектах, ролях, их возможностях по отношению к каждому из объектов, хранится в словаре СУБД.

Для хранения данной информации предназначена таблица привилегий \$\$\$USR. В ней пять столбцов с именами \$\$\$S31, \$\$\$S32, \$\$\$S33, \$\$\$S34 и \$\$\$S35. Таблица \$\$\$USR может содержать записи разных типов (см. таблицы 1-4).

**Таблица 1. Описание пользователя**

\$\$\$S31	\$\$\$S32	\$\$\$S33	\$\$\$S34	\$\$\$S35
ID (> 0)	0	Уровень прав	Имя пользователя	Зашифрованный пароль

**Таблица 2. Описание роли**

\$\$\$S31	\$\$\$S32	\$\$\$S33	\$\$\$S34	\$\$\$S35
ID (< 0)	0	ID владельца	Имя роли	

**Таблица 3. Описание прав на таблицу**

\$\$\$S31	\$\$\$S32	\$\$\$S33	\$\$\$S34	\$\$\$S35
ID (<> 0)	ID таблицы	Тип привилегии	Имя пользователя	

**Таблица 4. Описание назначения роли**

\$\$\$S31	\$\$\$S32	\$\$\$S33	\$\$\$S34	\$\$\$S35
ID (<> 0)	ID роли (< 0)	Тип привилегии	Имя пользователя	

ID пользователя всегда равен RowId его описания, ID роли также равен RowId, но с обратным знаком. **Имена всех пользователей и ролей должны быть уникальны.**

Категории прав: DBA, RESOURCE, CONNECT.

Типы привилегий на таблицу: SELECT, INSERT, DELETE, UPDATE, ALTER, INDEX и ALL (все перечисленные выше).

Категории прав имеют только пользователи. Привилегии – пользователи и роли.

## Категория доступа субъекта контроля

Назначение категории доступа субъекту контроля КСЗ НЗД СУБД ЛИНТЕР может быть осуществлено двумя способами:

- 1) создание субъекта контроля с неявным заданием категории доступа Connect:

```
CREATE USER <имя пользователя> IDENTIFIED BY <пароль>;
```

- 2) создание субъекта контроля с явным заданием категории доступа:

```
GRANT <категория> TO <имя пользователя> IDENTIFIED BY <пароль>;
```

## Роли

Аппарат ролей используется для разграничения доступа субъектов контроля СУБД ЛИНТЕР к объектам:

- 1) создание роли:

```
CREATE ROLE <имя роли>;
```

- 2) удаление роли:

```
DROP ROLE <имя роли>;
```

- 3) присвоение роли пользователю:

```
GRANT ROLE <имя роли> TO {<список пользователей>| PUBLIC};
```

```
<список пользователей>::=<имя пользователя> [, ...]
```

- 4) отмена присвоения роли

```
REVOKE ROLE <имя роли> FROM <имя пользователя>;
```

## Управление привилегиями доступа

Привилегии доступа могут быть назначены как пользователям, так и ролям:

- 1) назначение привилегии доступа пользователю:

```
GRANT <привилегия> ON <имя объекта> TO <имя пользователя>;
```

- 2) отмена привилегии доступа пользователю:

```
REVOKE <привилегия> ON <имя объекта> FROM <имя пользователя>;
```

- 3) назначение привилегии доступа роли:

```
GRANT <привилегия> ON <имя объекта> TO <имя роли>;
```

- 4) отмена привилегии доступа роли:


```
REVOKE <привилегия> ON <имя объекта> FROM <имя роли>;
```

## Группы пользователей

Все пользователи БД разделяются на непересекающиеся группы.

Группа описывает область доступных членам группы данных. Для каждой группы существует администратор группы (DBA-группы), созданный администратором SYSTEM (группа 0). Принадлежность группе задается командой ALTER.

Пользователи одной группы не видят данных пользователей другой группы.

 Группы не считаются заранее данными, каждую из них перед использованием необходимо создать.

Для управления группами используются следующие запросы:

1) создание группы:

```
CREATE GROUP <имя группы>[=<числовой идентификатор группы>];
```

Пример.

```
create group "1-й отдел";
```

При создании группы в таблицу \$\$\$GROUP заносится запись с RowId, равным числовому идентификатору группы; если он не указан, то присваивается первый свободный.

2) изменение имени группы:

```
ALTER GROUP <имя группы> SET <новое имя группы>;
```

Пример.

```
alter group "первый отдел" set "группа маркетинга";
```

3) назначение группы пользователю:

Назначение группы пользователю возможно через команду определения пользователя или через команду модификации определения пользователя с помощью дополнительной конструкции GROUP (<имя группы>):

```
CREATE USER <имя пользователя> IDENTIFIED BY <пароль>
```

```
GROUP <имя группы>;
```

```
ALTER USER <имя пользователя> GROUP <имя группы>;
```

Пример.

```
CREATE USER a IDENTIFIED BY 'ASD' GROUP "1-й ОТДЕЛ";
```

```
ALTER USER b GROUP "АДМИНИСТРАТОР";
```

4) предоставление доверия группе:

Пользователи группы А могут увидеть данные пользователей группы Б в случае, если DBA-группы Б установил флаг доверия группе А:

```
GRANT ACCESS ON <группа-доверитель> TO {<группа-приемник>|ALL};
```

<группа-приемник> ::= идентификатор группы, которой «оказывают доверие».

<группа-доверитель> ::= идентификатор группы, к данным которой могут обращаться пользователи <группы-приемника>.

5) отмена доверия группе:

```
REVOKE ACCESS ON <группа-доверитель>
```

```
FROM {<группа-приемник>|ALL};
```

<группа-приемник> ::= идентификатор группы, которой «отказывают в доверии».

<группа-доверитель> ::= идентификатор группы, к данным которой не могут обращаться пользователи <группы-приемника>.

Уровни доверия не могут быть вложенными.

Группа представляет собой числовое значение в диапазоне [1-250]. Значения в диапазоне [251-255] зарезервированы.

Группа 0 – группа администратора SYSTEM. При создании пользователя ему автоматически присваивается группа создателя.

Только пользователь SYSTEM может создать пользователя в группе, отличной от своей.

Все данные, созданные от имени пользователя, помечаются его группой.

Описания групп хранятся в таблице групп \$\$\$GROUP (см. таблицу 5).

**Таблица 5. Схема таблицы групп**

№	Имя	Тип	Длина	Содержание
1	\$\$\$ID	int	4	Номер группы
2	\$\$\$STATUS	int	4	Статус группы (используется 1-ый бит – запрет группе доступа к СУБД)
3	\$\$\$NAME	char	66	Имя группы
4	\$\$\$DESCR	byte	128	Описание группы
5	\$\$\$INFO	byte	32	Описание уровней доверия (описание доверия всем оставшимся группам – Bit-mask)

Группа может идентифицироваться по номеру или имени, если таковое задано.

Описание уровней доверия зарезервировано для дальнейшего использования.

Имя группы, как и описание, может быть изменено либо DBA-группы, либо пользователем SYSTEM при помощи SQL-предложения ALTER GROUP.

Номер группы входит в метку всех объектов БД.

## Уровни доступа

Уровни доступа (см. таблицу 6) вводятся для проверки на уровне ядра СУБД ЛИНТЕР прав на осуществление чтения/записи информации.

Вводятся следующие уровни доступа:

- 1) для пользователя (субъекта):
  - RAL-уровень доступа. Пользователь может получать (читать) информацию, RAL-уровень которой **не выше** его собственного уровня доступа;
  - WAL-уровень доверия на понижение уровня конфиденциальности. Пользователь не может вносить информацию с уровнем доступа (RAL-уровнем) более низким, чем данный WAL-уровень пользователя. Т.е. пользователь не

может сделать доступную ему информацию менее конфиденциальной, чем указано в данном параметре.

2) для информации:

- RAL-уровень чтения. Пользователь может получать (читать) информацию, RAL-уровень которой не выше его собственного RAL-уровня (может читать менее конфиденциальные данные);
- WAL-уровень ценности или уровень доступа на запись (модификацию, удаление). Пользователь может модифицировать (удалять) информацию, WAL-уровень которой не выше его RAL-уровня.

Всего вводится 10 уровней (0-10). Значения 11-15 – резерв. Уровни по умолчанию равны 0 (все имеют возможность читать и модифицировать доступные по дискреционному принципу данные).

Создать пользователя с произвольными уровнями может только пользователь SYSTEM. Остальные администраторы (DBA) способны создавать пользователей (или изменять для них уровень) только в пределах отведенных им уровней (на чтение – не выше, на запись – не ниже).

Пользователь может принудительно пометить вводимые данные, указав в списке атрибутов уровни доступа для соответствующих записей и полей (при операции insert или update).

По умолчанию вносимые данные наследуют уровни пользователя, вносящего/изменяющего данные.

Защищаемые объекты: пользователи, таблицы, столбцы, записи (вносятся при insert), поля записей (изменяются при update).

**Таблица 6. Схема таблицы уровней**

№	Имя	Тип	Длина	Содержание
1	\$\$\$ID	int	4	Номер уровня
2	\$\$\$NAME	char	66	Имя уровня
3	\$\$\$DESCR	varchar	128	Описание

## Управление уровнем доступа пользователей

Для управления уровнями доступа пользователей используются следующие запросы:

1) создание уровня:

```
CREATE LEVEL <имя уровня>=<номер уровня>;
```

2) изменение названия уровня:

```
ALTER LEVEL <имя уровня> SET <новое имя уровня>;
```

Выполнение данного запроса доступно только пользователю SYSTEM.

3) назначение уровня доступа пользователю:

Назначение уровня доступа пользователю возможно через команду определения пользователя либо команду модификации определения пользователя с помощью дополнительной конструкции LEVEL (<RAL>,<WAL>):

```
CREATE USER <имя пользователя> IDENTIFIED BY <пароль>
LEVEL (<RAL-уровень>,<WAL-уровень>);
```

```
ALTER USER <имя пользователя>
LEVEL (<RAL-уровень>,<WAL-уровень>);
```

Примеры.

```
CREATE USER a IDENTIFIED BY 'ASD'
LEVEL ("СОВ.СЕКРЕТНО", "для сл.пользования");
```

```
ALTER USER b LEVEL ("СОВ.СЕКРЕТНО", "для сл.пользования");
```

Значения групп и уровней доступа можно задавать в числовом виде.

## Управление уровнем доступа к таблице

В операторах ALTER TABLE и CREATE TABLE можно указать уровень доступа для всей таблицы через конструкцию LEVEL (<RAL>, <WAL>), например:

```
CREATE TABLE t (h INT, j CHAR(10))
LEVEL ("СОВ.СЕКРЕТНО", "для сл.пользования");
```

## Управление уровнем доступа к столбцу таблицы

В операторах ALTER TABLE и CREATE TABLE можно указать уровень доступа для столбца через конструкцию LEVEL (<RAL>, <WAL>), например:

```
CREATE TABLE t(h INT LEVEL ("СОВ.СЕКРЕТНО", "для
сл.пользования"), j CHAR(10));
```

## Представление групп и уровней в SQL-выражениях

Выше было определено, что метка данных (или метка пользователя) всегда состоит из трех составляющих: номера (или имени) группы и двух номеров (имен) уровней доступа (RAL, WAL):

```
<метка доступа> ::= # [<группа>] # [<RAL>] # [<WAL>]
```

<метка доступа> может сопровождать как всю таблицу (строку), так и столбец таблицы (поле строки).

## Получение информации о метках доступа

Для получения информации о метке доступа используется встроенная в SQL функция.

### Синтаксис

```
SECURITY({*|<имя столбца>}, {'R' | 'W' | 'G'});
```

### Описание

1) 'R' | 'W' | 'G' – тип части метки доступа или указание для выдачи значений RAL, WAL и GROUP соответственно;


2) функция всегда возвращает значение типа `integer`, являющееся значением указанной части метки. При этом если:

- в качестве первого параметра употреблен символ звездочка (\*), то результатом функции будет служить значение указанной (вторым параметром) части метки **текущей строки**;
- в качестве первого параметра употреблено имя столбца, то результат функции – значение указанной части метки соответствующего столбцу **поля** текущей строки.

3) функцию можно использовать в тех местах SQL-запроса, где по синтаксису допустимо использование скалярной функции. Это может быть получение справочной информации – на выходе `select`-запроса или в поисковом условии (в `where`-предложении).

## SELECT-выражения

Кроме вышеописанной функции `SECURITY` в `select`-выражениях нет специальных возможностей для использования меток, конфиденциальности и пр. Все действия по проверке доступа производятся ядром СУБД ЛИНТЕР прозрачно для пользователя.

 Выполняя поисковые операции, ядро СУБД будет принимать во внимание **только доступные (видимые)** пользователю данные. Таким образом, у каждого пользователя будет свое представление о содержании данных в БД или ее таблицах.

Следовательно, субъекты, облеченные большим доверием, будут **видеть** больше информации там, где другие пользователи **найдут** лишь небольшую часть данных (доступных им).

Здесь исключены все возможные, даже косвенные, лазейки, так как правило видимости касается и таких конструкций, как агрегатные функции. Так, информация о минимальном значении данных какого-либо столбца будет выдана, исходя из доступного множества значений этого столбца. То же касается и остальных агрегатных функций (`MIN`, `MAX`, `AVG`, `COUNT` и др.).

## UPDATE и INSERT операторы

<Метка\_доступа> в этих операторах может сопровождать как всю таблицу (строку), например,

```
INSERT INTO Auto<метка доступа> VALUES (...);
```

так и отдельные поля, например,

```
INSERT INTO Auto (PersonID<метка доступа>) VALUES (10000);
```

В первом случае считается, что метка доступа относится ко всей строке, во втором – только к указанному полю.

Если пропущена <метка доступа> строки, то по умолчанию она берется равной метке доступа текущего пользователя.

Если пропущена <метка доступа> поля, то по умолчанию она берется равной метке доступа строки.

Пример.

```
UPDATE Auto SET
Make#"1-ый Отдел"#"СЕКРЕТНО"#"СОВ. СЕКРЕТНО"='ШКОДА'
WHERE Make='FORD';
```

## Управление мандатной защитой на уровне сессии

Для того чтобы не указывать группу, RAL и WAL в каждом INSERT- или UPDATE-запросе, выполняемом пользователем в текущей сессии, можно использовать команду

```
SET SESSION DEFAULT SECURITY #[<группа>]#[<RAL>]#[<WAL>];
```

Команда распространяется на все открытые дочерние каналы (которые уже открыты ранее или будут открыты впоследствии).

Пример.

```
create LEVEL "NS" = 1;
create LEVEL "DSP" = 2 ;
create LEVEL "C" = 3;
create LEVEL "CC" = 4;

drop user "TEST" cascade;
create user "TEST" identified by 'TEST';
grant DBA to "TEST";
username TEST/TEST
create or replace table a (i int, j int);
! TEST FOR 'INSERT'
insert into a values(1,1);
username SYSTEM/MANAGER
alter user TEST level ("DSP","DSP");
username TEST/TEST
insert into a values(2,2);
!ok
set session default security ##C#C;
insert into a values(3,3);
!ok
set session default security ##CC#CC;
insert into a values(4,4);
!2 rows
select security(*,'R'),security(*,'W') from a;
username SYSTEM/MANAGER
alter user TEST level ("CC","CC");
username TEST/TEST
!4 rows
select security(*,'R'),security(*,'W') from a;
! TEST FOR 'UPDATE'
update a set i=10,j=10 where i=1;
!4 rows
select security(*,'R'),security(*,'W') from a;
!error: MandatoryViolation
set session default security ##NS#NS;
```

### Маркировка документов

Согласно архитектуре открытых систем, СУБД ЛИНТЕР не имеет средств вывода информации на документы, однако ЛИНТЕР предоставляет разработчику приложения все возможности для реализации маркировки документов.

Функция SECURITY, описанная выше, обеспечивает не только получение справочной информации для администратора средств защиты, но и является главным инструментом, позволяющим приложению маркировать документы (выводимые данные).

На самом деле, главную роль при маркировке играет уровень конфиденциальности (чтения) – RAL-уровень получаемых данных. При получении разноуровневых (по RAL-уровню) данных маркировка документов должна основываться на уровне самых конфиденциальных из них, т.е., на максимальном уровне полученной информации.

Следовательно, при маркировке документов основным аппаратом программирования приложения станет функция SECURITY(..., 'R').

### Контроль доступа к БД с рабочих станций

СУБД ЛИНТЕР представляет собой многопользовательскую систему. Пользователи могут получать удаленный доступ к БД с различных сетевых терминальных станций. СУБД ЛИНТЕР отличает различные станции сети по их **протоколу обмена** данными и **уникальному сетевому адресу** в пределах одного протокола.

СУБД ЛИНТЕР позволяет администратору системы безопасности регулировать доступ пользователей к БД по следующим критериям:

- 1) по времени работы пользователя;
- 2) по количеству одновременно активных логических соединений к БД;
- 3) по списку разрешенных для доступа станций;
- 4) по уровням доступа;
- 5) по списку разрешенных для доступа групп.

Основополагающим понятием в процессе сопоставления пользователя с устройством является понятие *сетевой станции*. Сетевой станцией СУБД ЛИНТЕР считает любой компьютер, имеющий уникальный идентификатор – адрес в сети.

СУБД ЛИНТЕР поддерживает несколько типов сетей, что требует различного подхода к интерпретации сетевых адресов.

Рассмотрим структуру сетевого адреса. В общем случае сетевой адрес состоит из *адреса подсети* (уникального в пределах всей сети) и *адреса станции* (уникального в пределах подсети). СУБД ЛИНТЕР позволяет управлять доступом как на уровне конечной станции, так и на уровне подсети. В последнем случае ограничения, наложенные на всю подсеть, влияют и на все станции, расположенные в данной подсети.

Каждый сетевой адрес в СУБД ЛИНТЕР характеризуется следующими параметрами:

- тип сети (определяет внутреннюю структуру адреса);
- тип адреса (указывает на подсеть или конкретный узел);
- адрес в сети (собственно сетевой адрес, в зависимости от типа сети, может включать адрес подсети, а может не включать его);

- маска разрешенных групп (стандартная битовая маска, описывающая, разрешен ли доступ данной группы к станции);
- уровень мандатного доступа (проверяется возможность пользователя выполнять соответствующие операции по отношению к данной станции);
- маска разрешенного времени доступа (с точностью до получаса описывается время, разрешенное для доступа со станции).

Создать новую сетевую станцию (изменить характеристики существующей) может только администратор безопасности БД.

Право на доступ с конкретной сетевой станции может быть предоставлено администратором безопасности. В таком случае возможно установление временных ограничений для доступа данного пользователя с предоставляемой сетевой станции.

При попытке установления соединения подсистема защиты СУБД выполняет следующие действия:

- 1) проводит стандартные проверки идентификации и аутентификации пользователя;
- 2) проверяет наличие у пользователя категории Connect;
- 3) получает у операционной системы тип сети и адрес сети – источника запроса на установку соединения;
- 4) проверяет, ограничен ли доступ для данного пользователя (по временным показателям);
- 5) проверяет, существует ли для данного пользователя список разрешенных или запрещенных сетевых станций;
- 6) если такой список существует, подвергается проверке совпадение меток доступа для пользователя и разрешенной станции (группа пользователя должна содержаться в маске групп пользователей у станции; RAL-уровень пользователя **не должен быть выше** RAL-уровня станции, WAL-уровень пользователя **не должен быть ниже** WAL-уровня станции);
- 7) при успешной проверке рассматривается возможность данного пользователя работать с этой станции в текущий момент времени;
- 8) если запрещенных комбинаций не обнаружено, доступ разрешается.

В таблице 7 приведена структура таблицы сетевых станций (\$\$\$STATION).

**Таблица 7. Схема таблицы сетевых станций**

№	Имя	Тип	Длина	Содержание
1	\$\$\$ID	int	4	Идентификатор станции
2	\$\$\$NAME	char	66	Имя станции
3	\$\$\$DESCR	byte	200	Характеристики станции

Идентификатор станции предназначен только для ускорения поиска при установлении соединения (смысловой нагрузки с точки зрения подсистемы защиты не несет).

Имя станции представляет собой идентификатор, который может использоваться администратором безопасности для разрешения (запрещения) доступа пользователей с данного устройства.

Характеристики станции включают:

- общее число неуспешных попыток обращения со станции;
- текущее число неуспешных попыток обращения со станции;
- флаги доступа (доступ запрещен, требуется проверка группы);
- уровни доступа с устройства для мандатного доступа;
- маска разрешенных для доступа групп;
- маска временного доступа: битовое поле разрешенного времени работы со станции (с точностью до получаса на неделю);
- время последнего неудачного доступа;
- время последнего успешного доступа;
- дата и время, начиная с которого доступ с данной станции разрешается;
- дата и время, до которого доступ с данной станции разрешается;
- маска разрешенных для доступа дней недели.

Уровни мандатного доступа для сетевой станции представляют собой два числа (со значениями из отрезка [0...15]).

Маска разрешенных для доступа групп представляет собой битовую маску из 256 бит, первые 250 бит которой кодируют разрешение соответствующей группы на доступ к станции.

Дата и время, начиная с которого доступ с данной станции разрешается, представляют собой одностороннее ограничение на дату начала разрешения доступа с данной станции.

Дата и время, начиная с которого доступ с данной станции запрещается, представляют собой одностороннее ограничение на дату прекращения доступа с данной станции.

Маска разрешенных для доступа дней недели представляет собой битовую маску разрешенных для доступа дней недели.

Перед использованием станции ее необходимо сделать видимой для СУБД ЛИНТЕР. Это можно сделать двумя способами:

- 1) создать станцию (SQL-синтаксис приведен ниже), т.е. включить ее описание в список станций, в таблицу \$\$\$STATION;
- 2) перевести СУБД ЛИНТЕР в режим беспрепятственной работы с неизвестными станциями (UNLISTED STATIONS).

Для управления рабочими станциями используются следующие команды:

- 3) создание станции:

```
CREATE [IF NOT EXISTS] STATION <имя станции>  
PROTOCOL <сетевой протокол>  
ADDRESS <адрес станции>  
[LEVEL (<RAL>, <WAL>)]  
[<рабочее время>]  
<имя станции>::=<идентификатор>
```

```

<сетевой протокол> ::= <символьный литерал>
<адрес станции> ::= <символьный литерал>
<RAL> ::= <идентификатор>
<WAL> ::= <идентификатор>
<рабочее время> ::=
  {ENABLE |DISABLE} LOGIN
  {ALWAYS|<график по времени>|<график по дням>}
<график по времени> ::=
  FROM <время начала работы>
  TO <время окончания работы> [FOR <дни работы>]

<график по дням> ::= {SINCE <дата начала>} {UNTIL <дата окончания>}
<время начала работы> ::= 'HH:MM'
<время окончания работы> ::= 'HH:MM'
<дни работы> ::= <день недели> {[,<день недели>] ...}
<день недели> ::= 'MON'|'TUE'|'WED'|'THU'|'FRI'|'SAT'|'SUN'
<дата начала> ::= 'DD.MM.YYYY'
<дата окончания> ::= 'DD.MM.YYYY'

```

Опция IF NOT EXISTS отменяет выполнение оператора, если указанная станция уже зарегистрирована в БД.

4) удаление станции:

```
DROP STATION <имя станции>;
```

5) разрешить доступ к станции:

```
GRANT ACCESS ON UNLISTED STATION TO {<имя группы> | ALL};
```

Команда разрешает работу со станций, не перечисленных в списке станций СУБД ЛИНТЕР.

6) запретить доступ к станции:

```
REVOKE ACCESS ON UNLISTED STATION FROM {<имя группы> | ALL};
```

Команда запрещает работу со станций, не перечисленных в списке станций СУБД ЛИНТЕР.

7) изменение параметров станции:

```
ALTER STATION <имя станции> [SET<новое имя станции>]
ADDRESS <адрес станции>
[LEVEL(<RAL>,<WAL>)] [<рабочее время>];
```

8) разрешить группе/всем пользователям доступ к станции:

```
GRANT ACCESS ON STATION <имя станции> TO {<имя группы>|ALL};
```

9) запретить группе/всем пользователям доступ к станции:

```
REVOKE ACCESS ON STATION <имя станции> FROM {<имя группы>|ALL};
```

## Защита ввода-вывода на внешний носитель

СУБД ЛИНТЕР использует внешние устройства постоянного хранения информации для размещения таблиц данных и временных рабочих файлов.

Расположение конкретного объекта БД (файла таблицы, временного рабочего файла) внутри БД однозначно идентифицируется *четырёхсимвольным идентификатором* – ЛИНТЕР-именем устройства. ЛИНТЕР-имя используется при создании новых таблиц или изменении расположения файлов старых.

Соответствие ЛИНТЕР-имени устройства и реального физического устройства описывается таблицей \$\$\$DEVICE (см. таблицу 8). Данная таблица создается на уровне администратора безопасности БД и доступна только ему.

**Таблица 8. Схема таблицы устройств**

№	Имя	Тип	Длина, байт	Содержание
1	\$\$\$ID	integer	4	Идентификатор устройства.
2	\$\$\$LNAME	char	4	ЛИНТЕР-имя устройства (имя данного физического устройства в составе БД).
3	\$\$\$PNAME	char	256	Физический путь к устройству.
4	\$\$\$NAME	char	128	Описание устройства. Может содержать комментарии администратора безопасности, служащие для понимания назначения устройства.
5	\$\$\$DESCR	byte	40	Описание доступа к устройству.

Описание доступа к устройству (столбец \$\$\$DESCR) включает следующие составляющие:

### Метка доступа

Набор из двух значений: **RAL** и **WAL**. Служит для принудительного контроля над созданием файлов БД (файлов таблиц и временных файлов) на описываемом устройстве:

- **RAL**-устройства представляет собой минимальный уровень доступа пользователя (**RAL**-пользователя), необходимый для создания этим пользователем объектов (таблиц и временных файлов) на данном устройстве;
- **WAL**-устройства представляет собой максимальный **WAL**-уровень пользователя, необходимый для удаления таблицы.

### Маска признаков доступа

Проводится или не проводится проверка доступа к соответствующему устройству, запрещен или нет доступ к нему целиком для системы. В последнем случае все запросы на создание новых (получение информации из существующих) объектов будут отвергаться.

*Маска разрешения доступа для групп пользователей*

Описываются все 250 групп.

Существует дополнительный параметр, позволяющий управлять доступом для устройств, **отсутствующих в таблице устройств**. Он указывает, будет ли запрашиваться информация о неописанных ЛИНТЕР-устройствах у операционной системы, или запросы на работу с данными устройствами будут отвергаться.

Для управления контролем над устройствами используются следующие команды:

1) создание устройства:

```
CREATE [IF NOT EXISTS] DEVICE <имя устройства> DIRECTORY <каталог>
[COMMENT <комментарий>]
[LEVEL (<RAL-устройства>,<WAL-устройства>)];
```

<имя устройства>

ЛИНТЕР-имя, используемое в качестве имени устройства при указании расположения файлов создаваемых таблиц.

<каталог>

Символьная строка (длиной до 256 символов), определяющая местоположение (путь) (в терминах соответствующей ОС) файлов таблиц при их создании (в команде CREATE TABLE) или модификации местоположения этих файлов (в командке ALTER TABLE).

Оператор CREATE DEVICE создает новое ЛИНТЕР-имя, которое впоследствии может использоваться в запросах на создание (модификацию) таблиц. При создании устройства может быть указан комментарий к нему (пояснение о назначении устройства) и уровни доступа к устройству.

Опция IF NOT EXISTS отменяет выполнение оператора, если указанное ЛИНТЕР-имя устройства уже существует в БД.

2) удаление устройства:

```
DROP DEVICE <имя устройства>;
```

Команда удаляет ЛИНТЕР-имя устройства из списка разрешенных имен.

3) модификация параметров устройства:

```
ALTER DEVICE <имя устройства> [DIRECTORY <каталог>]
[LEVEL (<RAL-устройства>,<WAL-устройства>)];
```

Команда позволяет изменить значение ЛИНТЕР-имени (путь к физическому устройству) и уровни защиты устройства.

При использовании операторов ALTER DEVICE и DROP DEVICE необходимо учитывать, что существующие файлы, расположенные на новых устройствах, могут стать недоступными для некоторых пользователей.

Значение (содержание) устройства SY00 определяется в момент старта системы и **не может быть переопределено** в произвольный каталог, но все ограничения, наложенные на него, будут выполняться обычным образом, за исключением отношений, необходимых для функционирования СУБД ЛИНТЕР и соответствующих рабочих файлов.

4) разрешить группе/всем пользователям доступ к устройству:

```
GRANT ACCESS ON DEVICE <имя устройства> TO {<имя группы>|ALL};
```

5) запретить группе/всем пользователям доступ к устройству:

```
REVOKE ACCESS ON DEVICE <имя устройства> FROM {<имя группы>|ALL};
```

При выполнении команды **REVOKE ACCESS** необходимо учитывать, что существующие объекты, расположенные на новом месте, могут стать недоступными для соответствующих групп.

6) разрешить группе/всем пользователям доступ к незарегистрированным устройствам:

```
GRANT ACCESS ON UNLISTED DEVICE TO {<имя группы>|ALL};
```

Команда разрешает работу с ЛИНТЕР-именами устройств средствами операционной системы (СУБД будет запрашивать информацию о физическом расположении устройства у операционной системы).

7) запретить группе/всем пользователям доступ к незарегистрированным устройствам:

```
REVOKE ACCESS ON UNLISTED DEVICE FROM {<имя группы>|ALL};
```

Команда запрещает работу с ЛИНТЕР-именами устройств средствами операционной системы.

При использовании команды **REVOKE ACCESS** необходимо учитывать, что существующие объекты, расположенные на новом месте, могут стать недоступными для соответствующих групп.

Информация о возможности работы с неописанными ЛИНТЕР-именами устройств располагается в описании БД.

## Контроль целостности средств защиты данных

СУБД ЛИНТЕР обладает широким набором функциональных средств защиты данных:

- подсистему дискреционного доступа;
- подсистему мандатного доступа;
- диспетчер доступа;
- подсистему очистки памяти;
- подсистему контроля над внешними физическими устройствами хранения информации;
- подсистему контроля доступа с внешних устройств;
- подсистему идентификации и аутентификации;
- подсистему регистрации событий.

КСЗ СУБД ЛИНТЕР реализован в виде части исполняемого кода ядра СУБД. Ядро системы безопасности – диспетчер доступа, который получает управление на ранних этапах инициализации СУБД до запуска процедур обслуживания пользовательских запросов.

Дальнейшая работа СУБД происходит под контролем КСЗ.

Целостность КСЗ в СУБД ЛИНТЕР эквивалентна целостности программного кода ядра СУБД ЛИНТЕР.

Для проверки целостности СУБД применяется утилита подсчета контрольных сумм, которая создает на выходе файл, содержащий контрольные суммы проверяемых файлов. Целостность СУБД считается подтвержденной в случае, если результирующие контрольные суммы совпали с контрольными суммами, приведенными в документации.

Проверка выполняется перед каждым запуском ядра СУБД.

Подсчет контрольных сумм ведется только для указанных в таблице 9 файлов с помощью утилиты count. Запуск данной утилиты (поставляемой в составе дистрибутива СУБД ЛИНТЕР) производится с помощью командной строки:

```
count <имя файла>
```

Значение контрольной суммы выдается на терминал, и его нужно только сравнить с приведенными в таблице 9 значениями.

**Таблица 9. Содержание каталога исполняемых модулей \Linter\Bin**

Имя файла	Размер (в байтах)	Назначение	Контрольная сумма
gendb		Генератор БД	
intsrt		Процессор сортировки ответов	
linter		Ядро СУБД ЛИНТЕР	
sql		SQL-транслятор	

## Подсчет контрольной суммы

### Назначение

Для контроля целостности КСЗ СУБД ЛИНТЕР используется утилита count, которая подсчитывает 32-битную контрольную сумму файла. Ее использование также целесообразно для контроля неизменности всех исполняемых модулей, контрольные суммы для которых прилагаются.

### Описание

Производится расчет 16-байтной последовательности символов, однозначно идентифицирующих заданный файл. Расчет осуществляется с использованием распространенного алгоритма вычисления аутентифицирующих кодов Message Digest в режиме сцепления по промежуточному результату вычислений.

Затем результат суммируется со сдвигом для получения результирующей 32-битной контрольной суммы.

### Использование

Командная строка:

```
count <имя файла>
```

Значение контрольной суммы выдается на терминал, и его нужно только сравнить с приведенными в таблице 9 значениями.

## Диспетчер доступа

Диспетчер доступа СУБД ЛИНТЕР реализован в виде нескольких взаимосвязанных программных фрагментов, каждый из которых контролирует свою область действий СУБД.

Первая часть обслуживает подсистему **дискреционного доступа**. Она получает управление после разбора запроса. В этот момент полностью определены все объекты доступа и запрашиваемые действия. Диспетчер выбирает все правила, касающиеся запрашивающего пользователя, и, в соответствии с запросом, проверяет, доступны ли требуемые объекты в данном запросе. Также проверяется возможность работы с внешними устройствами. Данный фрагмент может обращаться к подсистеме идентификации за подтверждением (запретом) действий, связанных с категориями пользователей (Connect, Resource, Db).

Вторая часть обслуживает подсистему идентификации и аутентификации. Она получает управление в момент открытия логической связи с СУБД при изменении прав пользователей и таблицы сетевых станций. Данный фрагмент обеспечивает идентификацию, аутентификацию, проверку категорий доступа, проверку сопоставления пользователей с устройством.

Третий фрагмент получает управление при каждой операции манипуляции с конкретными данными – при получении их из БД, проведении операций над ними, записи в таблицы БД, выдаче данных пользователю и т.п. В этом случае проверяются **мандатные правила доступа**.

Четвертый фрагмент диспетчера доступа СУБД ЛИНТЕР обеспечивает изоляцию параллельно выполняющихся запросов. Он также осуществляет планирование обработки запросов, контролирует их квантование и отслеживает все возможные нарушения изоляции процессов, выполняющихся по разным логическим связям.

Все фрагменты диспетчера доступа в процессе работы обращаются к подсистеме регистрации событий СУБД ЛИНТЕР, которая может принять решение согласно своим параметрам о регистрации соответствующих событий в таблице событий.

## Создание БД

Создание БД ЛИНТЕР выполняется с помощью с помощью утилиты gendb (см. документ «СУБД ЛИНТЕР. Создание и конфигурирование базы данных»). В процессе генерации создаются файлы системной и рабочей БД.

## Системная БД

Системная БД представляет собой набор из трех таблиц, содержащих метаинформацию, т.е. данные о данных:

- 1) `$$$SYSRL` (файлы с именами 1.\*) – таблица таблиц. Содержит информацию о таблицах: имя, идентификатор владельца, число столбцов, размеры файлов и пр. Это каталог таблиц БД, выполненный также в виде таблицы;
- 2) `$$$ATTRI` (файлы с именами 2.\*) – таблица столбцов. Содержит информацию о столбцах таблиц: имя, принадлежность к таблице, тип данных, информация об индексах, ограничения целостности и пр.;
- 3) `$$$USR` (файлы с именами 3.\*) – таблица полномочий. Содержит информацию о пользователях и их полномочиях при работе с таблицами, ролях и пр.

Кроме описаний пользовательских таблиц, системные таблицы содержат и свои собственные описания.

После генерации системной БД с помощью утилиты `gendb` она будет содержать следующие таблицы:

- `$$$SYSRL` – описания трех системных таблиц: `$$$SYSRL`, `$$$ATTRI`, `$$$USR`;
- `$$$ATTRI` – описания 13 столбцов системных таблиц;
- `$$$USR` – описание единственного пользователя БД, имеющего полномочия на чтение при работе со всеми системными таблицами (администратора БД).

При создании новых таблиц и пользовательских представлений или при изменении полномочий состояние системной БД меняется.

Создание пользовательской таблицы/представления влечет за собой добавление ее описания в `$$$SYSRL` и описаний ее столбцов в `$$$ATTRI`. При удалении таблицы СУБД выполняет обратные действия.

Доступ к любой таблице (пользовательской/системной) БД определяется по информации из системных таблиц.

Системные таблицы содержат жизненно важную для работы БД информацию.

Изменение данной информации может привести к **катастрофическим последствиям**. Во избежание изменения пользователями системных таблиц доступ к ним **запрещен для всех пользователей** (даже администратор БД имеет лишь привилегию `SELECT`). Все необходимые изменения в системных таблицах в процессе работы производит только ядро СУБД ЛИНТЕР.

## Структура системных таблиц

Системные таблицы, так же, как и пользовательские, состоят из столбцов, по значениям которых возможен поиск.

Однако среди столбцов системных таблиц есть столбцы, содержащие в байтовом виде интегрированную информацию о включенном в эту таблицу объекте БД. Приложению, не являющемуся системным, эта информация чаще всего не требуется. Клиентские приложения работают с СУБД на уровне SQL-запросов к пользовательским таблицам. Всю прочую работу выполняет СУБД.

Только системным приложениям может потребоваться информация о расположении файлов БД, о том, как отличить базовую таблицу от представления, о типе, длине столбца и пр.

Схема `$$$SYSRL` приведена в таблице 10.

**Таблица 10. Схема таблицы таблиц**

Имя	Тип	Длина, байт	Содержание
\$\$\$S11	int	4	Системный идентификатор таблицы
\$\$\$S12	int	4	Идентификатор владельца таблицы
\$\$\$S13	char	66	Имя таблицы
\$\$\$S14	byte	262	Вспомогательная системная информация

Схема \$\$\$ATTRI приведена в таблице 11.

**Таблица 11. Схема таблицы столбцов**

Имя	Тип	Длина, байт	Содержание
\$\$\$S21	int	4	Идентификатор таблицы, содержащей данный столбец
\$\$\$S22	smallint	2	Идентификатор столбца
\$\$\$S23	char	66	Имя столбца
\$\$\$S24	byte	80	Вспомогательная системная информация

Схема \$\$\$USR приведена в таблице 12.

**Таблица 12. Схема таблицы привилегий**

Имя	Тип	Длина, байт	Содержание
\$\$\$S31	int	4	Идентификатор пользователя таблицы
\$\$\$S32	int	4	Идентификатор таблицы
\$\$\$S33	int	4	Набор привилегий данного пользователя на эту таблицу
\$\$\$S34	char	66	Имя пользователя
\$\$\$S35	char	240	Пароль пользователя и прочая дополнительная информация

## Рабочая БД

В рабочей БД содержатся минимум четыре файла (при наличии нескольких файлов системного журнала их может быть гораздо больше):

- 1) 1.31 – рабочий файл бит-векторов. Предназначен для свопинга бит-векторов ответов одно- и многопеременных запросов;
- 2) 1.41 – рабочий файл хранимых процедур и быстрой загрузки;

3) 1.51 – рабочий файл сортировки. Предназначен для выполнения сортировки ответов и предложений типа group by в SQL-операторах;

4) 000000.61, 000001.61, ... – файлы системного журнала. Предназначены для ведения протокола обо всех изменениях, произведенных СУБД в системной и пользовательской БД (сеансы чтения в журнале не регистрируются).

Файл бит-векторов используется ядром СУБД для хранения информации о том, какие записи вошли/не вошли в ответ (может быть промежуточным).

Программа sort последовательно обрабатывает запросы на сортировку, поэтому каждый раз в файле 1.51 записываются новые данные, которые сортируются до конца.

Размеры файлов должны быть определены до запуска СУБД. Однако они не остаются фиксированными на протяжении всего сеанса работы ядра СУБД, а расширяются по мере необходимости.

## Инициализация комплекса средств защиты данных

Без инициализации КСЗ СУБД ЛИНТЕР не будет поддерживать большинство из описанных выше функций (управление группами, уровни доступа, защиту устройств и пр.).

В процессе инициализации КСЗ создаются необходимые для её работы ресурсы:

- 1) для поддержки механизма меток должны быть созданы таблицы:
  - \$\$\$LEVEL;
  - \$\$\$GROUP.
- 2) для работы аппарата сопоставления пользователя с сетевым устройством необходимо создать таблицу \$\$\$STATION;
- 3) для работы подсистемы защиты ввода-вывода на внешний носитель требуется создать таблицу \$\$\$DEVICE;
- 4) для работы подсистемы регистрации событий нужно создать таблицу \$\$\$AUDIT.
- 5) для установления «отношений» между конкретными пользователями и конкретными объектами следует создать таблицу \$\$\$RELATION.

Все вышеперечисленные таблицы создаются с помощью SQL-файла `systab.sql`, который должен быть выполнен (с помощью утилиты `inl`) администратором безопасности.

Для этого необходимо запустить ядро СУБД ЛИНТЕР, затем, пользуясь утилитой `inl`, запустить на выполнение SQL-файл `security.sql`:

```
inl -u SYSTEM/MANAGER -f security.sql
```

После выполнения данного пакета запросов (из файла `systab.sql`) СУБД ЛИНТЕР следует выгрузить. Только при следующей загрузке СУБД она включит в работу указанные механизмы КСЗ и протоколирования.

# Мониторинг комплекса средств защиты данных


СУБД ЛИНТЕР позволяет хранить одновременно информацию, принадлежащую различным пользователям, поддерживать сложную иерархическую структуру доступа к данным. Для контроля над доступом пользователей к данным и для ликвидации ошибок или противоречий в структуре защиты БД служит подсистема регистрации событий (аудит).

Данная подсистема является неотъемлемой частью КСЗ СУБД ЛИНТЕР.

СУБД ЛИНТЕР позволяет регистрировать следующие события:

- включение механизмов идентификации и аутентификации;
- запросы на доступ к ресурсам БД;
- создание и уничтожение объектов БД;
- действия по изменению правил разграничения доступа (ПРД);
- все попытки доступа к БД;
- действия администратора БД.

Таблица регистрации \$\$\$AUDIT имеет 8 столбцов, представленных в таблице 13. Более подробно их содержание рассматривается ниже.

 Средства протоколирования СУБД ЛИНТЕР предоставляют администратору всего лишь возможность сбора информации об интересующих его событиях; для ее обработки и анализа должны разрабатываться специальные приложения.

**Таблица 13. Структура таблицы регистрации \$\$\$AUDIT**

Имя столбца	Тип данных	Содержание
EVENTTYPE	smallint	Источник события (пользователь, ядро СУБД, КСЗ ядра СУБД)
EVENTID	smallint	Тип события (запрос на доступ, изменение ПРД и т.п.)
USERNAME	char(18)	Имя пользователя, инициирующего событие
SOURCEADR	byte(24)	Сетевой адрес источника события
OBJNAME	char(38)	Полное имя объекта, обращение к которому вызвало событие
OBJTYPE	smallint	Тип объекта, к которому относится событие
BODY	byte(56)	Дополнительная информация о событии
USERTEXT	char(240)	Пользовательское сообщение

Таблица \$\$\$AUDIT имеет следующие особенности:

- 1) она не создается автоматически при создании БД. Для ее создания администратору защиты необходимо выполнить файл `sysstab.sql`;

- 2) запуск мониторинга КСЗ (аудита) осуществляется командой `AUDIT START`, при этом в нулевом кортеже (первая запись системной таблицы `$$$SYSRL`) устанавливается флаг активности подсистемы аудита;
- 3) останов аудита происходит по команде `AUDIT STOP`, при этом в нулевой кортеж системной таблицы `$$$SYSRL` заносится информация об остановке протоколирования. При возобновлении протоколирования все события, установленные до остановки протоколирования, будут активизированы;
- 4) отметка в таблице регистрации может быть произведена пользователем с помощью команды `AUDIT MESSAGE <текст сообщения>`. Это бывает необходимо в случае, если пользователь хочет оставить администратору безопасности некое сообщение (в данной версии СУБД не реализовано);
- 5) специальной характеристикой таблицы `$$$AUDIT` является параметр `RECORDS_LIMIT` – число записей в данной таблице, при превышении которого ядро СУБД `ЛИНТЕР` полностью очистит таблицу `$$$AUDIT`. Для исключения потери накопленной информации должно быть разработано специальное приложение, контролирующее степень заполнения таблицы `$$$AUDIT` и автоматически выгружающее данные из нее при угрозе переполнения;
- 6) изменять значение параметра `RECORDS_LIMIT` может только администратор защиты;
- 7) для доступа к таблице `$$$AUDIT` необходимо иметь соответствующие права доступа и категорию пользователя `DBA`;
- 8) СУБД `ЛИНТЕР` не позволяет архивировать таблицу `$$$AUDIT` отдельно от всей БД (т.е. ее можно сохранить только в архивном файле всей БД);
- 9) при работе с таблицей `$$$AUDIT` СУБД `ЛИНТЕР` использует только один режим работы транзакций – `AUTOCOMMIT`.

Описание столбцов таблицы `$$$AUDIT`:

- 1) `EventType` – источник события, подлежащего регистрации. В данной версии источником регистрируемых событий могут быть ядро СУБД `ЛИНТЕР` (например, при создании нового объекта) и КСЗ (например, при регистрации обращения к недоступному объекту);
- 2) `EventId` – идентификатор типа события. В данной версии СУБД этот столбец может содержать целые значения, определяющие попытку идентификации и аутентификации, запрос на доступ к защищаемому объекту, создание/удаление объекта, действия по передаче прав;
- 3) `SourceType` – тип источника. Этот столбец может принимать значения, показывающие вид работы приложения (вызвавшего событие) с системой, 0 – локальная работа, не нулевое значение указывает тип сети (`TCP/IP`, `IPX/SPX`);
- 4) `SourceAdr` – сетевой адрес станции клиента. Это значение несет реальную информацию только при не нулевом значении типа источника (при сетевой работе);
- 5) `UserName` – имя пользователя. Это имя пользователя, который инициировал событие (например, обратился к недоступному объекту);
- 6) `ObjName` – полное имя объекта. Это имя объекта, при обращении к которому возникло регистрируемое событие;
- 7) `Body` – дополнительная информация о событии (см. таблицу 14);

8) UserText – пользовательское сообщение. В данной версии СУБД ЛИНТЕР не используется.

**Таблица 14. Структура поля BODY таблицы \$\$\$AUDIT**

Поле структуры	Тип данных	Комментарий
EventTime	byte[16]	Дата и время события
Reserved	byte[16]	Зарезервировано
SourceType	word	Тип события
SourcePid	longint	Идентификатор процесса-сервера
SourceRPid	longint	Идентификатор процесса-клиента
SourceSocket	longint	Сетевой порт (сокет) процесса-клиента
EventStatus	longint	Состояние выполнения СУБД ЛИНТЕР
SourceStatus	longint	Зарезервировано
SourceSystemStatus	longint	Состояние операционной системы

Для удобства пользователей при выполнении файла `sysstab.sql` создается представление `AUDIT_EVENTS` (см. таблицу 15).

**Таблица 15. Структура представления AUDIT\_EVENTS**

Столбец	Тип данных	Комментарий
Event_time	date	Дата и время события
Event_type	char(19)	Тип события
Username	char(18)	Имя пользователя
Eventid	char(19)	Имя события
Networkaddress	char(24)	Сетевой адрес клиента
Objectname	char(38)	Имя объекта БД
Sourcepid	int	Идентификатор процесса-сервера
Sourcereapid	int	Идентификатор процесса-клиента
Socket	int	Сетевой порт (сокет) процесса-клиента
Status	int	Состояние выполнения СУБД ЛИНТЕР
Osstatus	int	Состояние выполнения ОС
Usertext	char(240)	Пользовательское сообщение

Коды типов событий приведены в таблице 16.

Таблица 16. Коды типов событий EVENT\_TYPE для \$\$\$AUDIT

Код	Мнемоника	Комментарий
1	SYSTEM EVENT	Системные события
2	RESOURCE EVENT	События, связанные с изменением структуры БД
3	AUTHORIZATION EVENT	События, связанные с подсистемой авторизации
4	TABLE EVENT	События, связанные с конкретными таблицами
5	CHANNEL EVENT	Канальные события

## Управление мониторингом

Для управления мониторингом предназначены следующие команды:

- 1) управление протоколированием

```
AUDIT {ENABLE|DISABLE|CLEAR} [<имя события> [ON <объект БД>]]
[FOR <имя пользователя>] [ BY {SESSION|STATEMENT|ACCESS }
[WHEN [NOT] SUCCESS];
```

<имя события>

Имя события, для которого устанавливается протоколирование.

<объект БД>

Пользователи, таблицы, представления.

<имя пользователя>

Имя пользователя. Если указан конкретный пользователь, то событие возникает при условии, что текущий пользователь совпадает с <Именем пользователя>.

**BY SESSION**

Событие протоколируется один раз для одного подключения (CONNECTION).

**BY STATEMENT**

Событие протоколируется один раз для каждой SQL-операции.

**BY ACCESS**

Событие протоколируется для каждой записи, данный режим является режимом по умолчанию. Зарезервировано для дальнейшего использования.

**WHEN NOT SUCCESS**

Событие протоколируется только при неудачном завершении, по умолчанию протоколируется всегда.

**WHEN SUCCESS**

Событие протоколируется только в случае успешного завершения.

Если имя события явно не задано, устанавливается протоколирование для всех возможных событий.

В подсистеме аудита СУБД ЛИНТЕР предусмотрено три типа установок:

- *глобальные* – установки, действующие для всех пользователей и объектов БД;
- *персональные* – установки для конкретных пользователей или объектов БД;
- *локальные* – установки для конкретных пользователей и объектов БД, то есть установки между конкретными пользователями и конкретными объектами.

При работе сначала проверяются глобальные установки, затем персональные и локальные. Таким образом, глобальные установки обладают наименьшим приоритетом, локальные – наибольшим.

### AUDIT ENABLE

Команда разрешает протоколирование событий. Если при этом не задано имя пользователя и имя объекта, то изменяются глобальные установки.

Если задано имя пользователя или объекта, изменяются персональные установки.

### AUDIT DISABLE

Команда запрещает протоколирование событий. При этом ее работа аналогична работе команды AUDIT ENABLE.

### AUDIT CLEAR

Команда сбрасывает (очищает) изменения установок, выполненные AUDIT ENABLE и AUDIT DISABLE:

- AUDIT CLEAR ... WHEN SUCCESS – очищает только «успешные» события.
- AUDIT CLEAR ... WHEN NOT SUCCESS – очищает только «неуспешные» события.
- AUDIT CLEAR ... – очищает все события.

2) протоколирование пользовательского сообщения:

AUDIT MESSAGE <текст сообщения>;

Команда создает в таблице \$\$\$AUDIT новую запись с типом «пользовательское сообщение» и <текстом сообщения> в поле USERTEXT.

3) начать комментирование протоколируемых событий:

AUDIT SET MESSAGE <комментарий>;

Для всех событий, протоколирующихся по соединению, в котором подана эта команда, в поле USERTEXT добавляется текст <комментария>.

Команда доступна всем пользователям БД.

Пример.

```
audit set message 'Приложение "Склад"';
```

4) отменить комментирование протоколируемых событий:

AUDIT CANCEL MESSAGE;

Команда доступна всем пользователям БД.

5) параметры протоколирования

AUDIT {SET |CANCEL} <параметры протоколирования>;

<параметры протоколирования>::=  
 RECORDS LIMIT <количество записей>|DAYS LIMIT <срок хранения>

<количество записей>

Ограничивает количество записей в таблице AUDIT.

<срок хранения>

Указывает количество последних дней, в которые необходимо хранить записи протоколирования. Резервировано для дальнейшего использования.

 Конструкция AUDIT CANCEL RECORDS LIMIT запрещена.

## Системные события

Системные события представлены в таблице 17.

**Таблица 17. Системные события**

ИД события	Имя события	Пояснение
4	STARTUP	Старт ядра СУБД
5	WARM RESTART	Теплый рестарт ядра СУБД
3	SHUTDOWN	Останов ядра СУБД
7	AUDIT START	Старт подсистемы протоколирования
8	AUDIT STOP	Останов подсистемы протоколирования
6	INTERNAL DIAGNOSTIC	Регистрация диагностики СУБД ЛИНТЕР

## События, связанные с БД

События, связанные с БД, представлены в таблице 18.

**Таблица 18. События, связанные с БД**

ИД события	Имя события	Пояснение
9	CREATE TABLE	Создание таблицы
10	CREATE VIEW	Создание представления
52	CREATE SYNONYM	Создание синонима
11	CREATE PROCEDURE	Создание процедуры
16	DROP TABLE	Удаление таблицы
17	DROP VIEW	Удаление представления
18	DROP SYNONYM	Удаление синонима
19	DROP PROCEDURE	Удаление процедуры
20	DROP TRIGGER	Удаление триггера

<b>ID события</b>	<b>Имя события</b>	<b>Пояснение</b>
53	USER MESSAGE	Сообщение пользователю
58	AUDUT COMMAND	Подана команда подсистемы аудита
59	CREATE STATION	Создание станции
60	ALTER STATION	Изменение станции
61	DROP STATION	Удаление станции
62	CREATE DEVICE	Создание устройства
63	ALTER DEVICE	Изменение устройства
64	DROP DEVICE	Удаление устройства
65	ALTER PROCEDURE	Изменение процедуры
66	CREATE TRIGGER	Создание триггера
67	ALTER TRIGGER	Изменение триггера
68	GRANT PROCEDURE	Назначение прав доступа к процедуре
69	REVOKE PROCEDURE	Отмена прав доступа к процедуре
70	EXECUTE PROCEDURE	Выполнение процедуры
71	EXECUTE TRIGGER	Выполнение триггера
72	CREATE SEQUENCE	Создание последовательности
73	DROP SEQUENCE	Удаление последовательности
74	CREATE CHARACTER SET	Создание кодировки
75	DROP CHARACTER SET	Удаление кодировки
76	CREATE TRANLATION	Создание правила трансляции
77	DROP TRANSLATION	Удаление правила трансляции
78	SET DATABASE NAMES	Установка кодировки БД по умолчанию
79	SET RECORD SIZE LIMITS	Ограничение размера записи
80	SET NAMES	Установка кодировки соединения по умолчанию
81	CREATE ALIAS	Создание алиаса
82	DROP ALIAS	Удаление алиаса
83	CREATE DESCRIPTION	Создание описания кодировки
84	DROP DESCRIPTION	Удаление описания кодировки
85	SET DEFAULT CHARACTER SET	Установка кодировки по умолчанию
86	TRUNCATE TABLE	Усечение таблицы
87	ALTER SEQUENCE	Модификация последовательности

## События, связанные с подсистемой доступа

События, связанные с подсистемой доступа, представлены в таблице 19.

**Таблица 19. События, связанные с подсистемой доступа**

ИД события	Имя события	Пояснение
21	CREATE USER	Создание пользователя
22	DROP USER	Удаление пользователя
23	ALTER USER	Изменение привилегий пользователя
24	ALTER PASSWORD	Изменение пароля пользователя
25	CREATE ROLE	Создание роли
26	DROP ROLE	Удаление роли
31	GRANT ROLE	Назначение роли
32	REVOKE ROLE	Отмена назначенной роли
27	CREATE GROUP	Создание группы
28	ALTER GROUP	Изменение группы
29	CREATE LEVEL	Создание уровня
30	ALTER LEVEL	Изменение уровня
33	GRANT ACCESS	Разрешение доступа к группе
34	REVOKE ACCESS	Отмена доступа к группе

## События, связанные с таблицами

События, связанные с конкретными таблицами, представлены в таблице 20.

**Таблица 20. События, связанные с таблицами**

ИД события	Имя события	Пояснение
13	INSERT	Добавление строки в таблицу
14	UPDATE	Изменение строки таблицы
12	SELECT	Выборка строк из таблицы
15	DELETE	Удаление строк таблицы
35	INSERT BY PROCEDURE	Добавление строки в таблицу из процедуры
37	UPDATE BY PROCEDURE	Изменение строки таблицы из процедуры
38	SELECT BY PROCEDURE	Выборка строк таблицы из процедуры
36	DELETE BY PROCEDURE	Удаление строк из таблицы из процедуры

ИД события	Имя события	Пояснение
40	UPDATE BY REFERENCE	Изменение строки в таблице при каскадном изменении
39	DELETE BY REFERENCE	Удаление строк из таблицы при каскадном удалении
42	DROP INDEX	Удаление индекса таблицы
41	CREATE INDEX	Создание индекса таблицы
44	REBUILD TABLE	Восстановление таблицы
43	PRESS TABLE	Сжатие таблицы
45	ALTER FILE	Изменение файла таблицы
48	LOCK TABLE	Блокировка таблицы
49	UNLOCK TABLE	Деблокировка таблицы
46	GRANT TABLE	Передача привилегий на таблицу
47	REVOKE TABLE	Отмена привилегий на таблицу
56	ADD COLUMN	Добавление столбца
57	ALTER TABLE	Модификация таблицы

## События, связанные с пользователями

События, связанные с пользователями, представлены в таблице 21.

**Таблица 21. События, связанные с пользователями**

ИД события	Имя события	Пояснение
50	COMMIT	Фиксация транзакции
51	ROLLBACK	Откат транзакции
1	CONNECT	Подключение пользователя к БД
2	DISCONNECT	Отсоединение пользователя от БД
54	OPEN CURSOR	Открытие подчиненного канала
55	CLOSE CURSOR	Закрытие подчиненного курсора

## **Механизм надежного восстановления**

Механизм надежного восстановления обеспечивается СУБД ЛИНТЕР. Его основой является ведение системного журнала, где отображаются изменения, которые производятся с БД всеми пользователями системы.

Действия, связанные с изменениями в системе защиты, также отображаются в системном журнале (создание/удаление нового пользователя/группы и т.д.).

Если пользователь получил уведомление о том, что его изменения перенесены в БД, то сбой оборудования не может привести к нарушению системы защиты.

# Преобразование данных

Преобразованию в БД ЛИНТЕР подлежат данные любой таблицы, содержащиеся в:

- файлах данных;
- BLOB-файлах.

Целью преобразования является затруднение доступа к данным БД средствами, отличными от средств СУБД ЛИНТЕР, в момент, когда ядро СУБД не работает (не загружено).

При загруженном ядре СУБД файлы таблиц монополюбно блокируются и доступны только ядру СУБД ЛИНТЕР.

Шифрация данных представляет собой операцию «исключающее ИЛИ» с константой для всех машинных слов, содержащихся в данных.

В файлах индексов шифрация не требуется, т. к. сжатие, которому подвергаются ключи таблицы, настолько серьезно, что может потребовать месяцы упорного труда для получения реальной информации.

Кроме того, страницы всех файлов таблицы снабжены проверочной информацией, не позволяющей незаметно изменить данные.

Пароли пользователей, прежде чем записываться в БД, подвергаются необратимому преобразованию. При этом используется алгоритм MAC (Message Authentication Code).

## Очистка оперативной/внешней памяти

Перераспределение внешней памяти возможно в двух случаях: при удалении/модификации таблицы (все/некоторые ее файлы удаляются), и при выгрузке системы (усекаются ее рабочие файлы).

В обоих случаях освободившееся дисковое пространство очищается с помощью трехкратной записи маскирующей информации.

Во время работы СУБД ЛИНТЕР отслеживает занятое ею место в оперативной памяти.

При выгрузке СУБД ЛИНТЕР обнуляет весь объем оперативной памяти, который она занимала.

# Изоляция модулей

При рассмотрении механизмов работы КСЗ СУБД ЛИНТЕР необходимо отдельно исследовать работу механизма параллельной обработки запросов различных пользователей. Контроль над изолированностью запросов отдельных пользователей производит диспетчер доступа СУБД ЛИНТЕР. Он получает управление:

- при установке логических связей (каналов) с СУБД различными пользователями и при их разрыве;
- при подаче запроса пользователем;
- при выдаче ответа пользователю;
- в процессе параллельной обработки запроса;
- при окончании кванта текущего запроса.

В момент установления логической связи с СУБД и при условии выполнения всех процедур проверки доступа диспетчер доступа выделяет под обработку запросов от данного пользователя управляющую структуру, располагающуюся в адресном пространстве СУБД ЛИНТЕР. В случае если с СУБД установлено несколько логических соединений, для каждого из них будет выделена своя собственная управляющая структура. Управляющая структура канала полностью описывает пользователя, установившего соединение, и содержит всю информацию о текущем состоянии выполнения запросов данного пользователя.

СУБД ЛИНТЕР в процессе обработки запросов оперирует не данными пользователя, а только управляющей информацией, характеризующей их. Эта информация представляет собой, в основном, массивы ссылок на файлы данных. Вся управляющая информация хранится в управляющей структуре соответствующего канала.

При параллельном исполнении запросов диспетчер доступа просто выбирает очередную структуру управления каналом в соответствии с приоритетами исполнения запросов. Дальнейшая работа СУБД ЛИНТЕР, вплоть до передачи управления диспетчеру доступа, ведется в контексте параметров и информации данной структуры.

Все модули СУБД ЛИНТЕР (кроме диспетчера доступа) получают на входе и выдают на выходе управляющую структуру, соответствующую каналу, для которого эти модули будут осуществлять свои функции. В процессе выполнения запроса диспетчер доступа контролирует невозможность использования или повреждения структур, относящихся к другим каналам. Любая подобная попытка считается ошибочной ситуацией, и попытка доступа к БД соответствующего пользователя будет завершена с ошибкой, указывающей на нарушение изоляции.

При корректном закрытии или аварийном разрыве логической связи диспетчер доступа производит процедуру освобождения канала. В этом случае уничтожается вся управляющая информация, касавшаяся данного канала, освобождаются все использованные каналом ресурсы СУБД, и уничтожается структура управления каналом.



- списки субъектов и их паролей могут различаться по числу элементов. При этом список субъектов должен быть не уже списка паролей. Соответствие «пользователь-пароль» производится слева направо. Пользователи, оставшиеся без соответствия, получают «пустой» пароль (18 пробелов);
- строковый пароль может содержать до 18 символов;
- привилегии могут передаваться только владельцем указанной таблицы/представления;
- вводить нового пользователя (с определением его категории) может только администратор (категория DBA).

## Отмена привилегий

Отмена привилегии выполняется оператором Revoke, схема которого приведена на рис. 2.

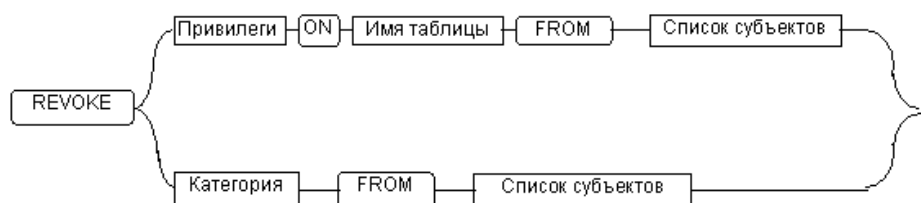


Рис. 2. Схема оператора Revoke

Общие правила:

- синтаксические правила те же, что и для определения привилегий;
- привилегии может отменять только владелец таблицы;
- отнять (понизить) категорию пользователя может только администратор БД (категория DBA);
- при этом считается, что  $CONNECT \subset RESOURCE \subset DBA$ , т.е. категории упорядочены. Когда отнимается какая-либо категория, это означает снижение категории на один уровень. Например:  
 $DBA > DBA = RESOURCE$ ;  
 $DBA > RESOURCE = CONNECT$ ;  
 $RESOURCE > RESOURCE = CONNECT$
- снятие привилегии CONNECT (лишение всех прав) равноценно удалению пользователя.

## Создание/удаление пользователя

Схемы создания/удаления пользователя представлены на рис. 3, 4.

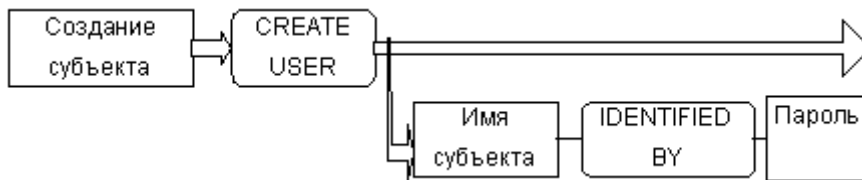


Рис. 3. Схема создания пользователя

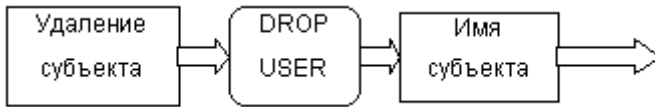


Рис. 4. Схема удаления пользователя

Общие правила:

- для создания/удаления нового пользователя необходимо иметь уровень прав DBA;
- по умолчанию пользователь создается без пароля. Уровнем прав ему назначается CONNECT;
- нельзя удалить пользователя, если в БД после этой операции будут оставаться созданные им таблицы или роли;
- вместе с пользователем удаляются все его привилегии и назначения ролей.

## Изменение пароля пользователя

Схема изменения пароля пользователя приведена на рис. 5.

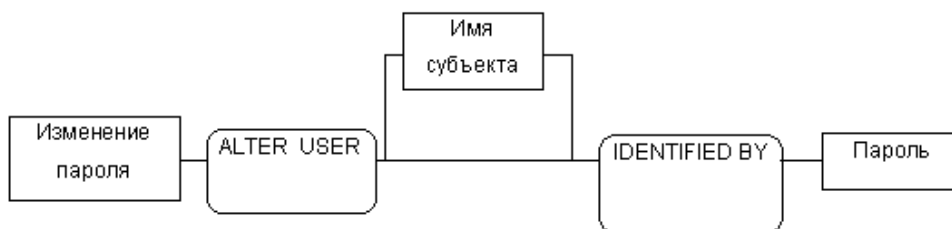



Рис. 5. Схема изменения пароля пользователя

Общие правила:

- изменить пароль могут администратор (имеющий права DBA) и пользователь (сам себе);
- если имя пользователя не задано, то подразумевается пользователь, подавший эту команду.

 Администратор не может незаметно выполнить изменение пароля, т.к. не имеет возможности вернуть обратно старый пароль. Если он (администратор) изменит пароль пользователя и получит доступ к конфиденциальной информации с именем пользователя и его новым паролем, это не останется незамеченным. Пользователь не будет знать измененного пароля (старый восстановить администратор не сможет), не сможет получить доступ к БД и забьет тревогу.

## Создание/удаление роли

Схемы создания/удаления роли приведены на рис. 6.

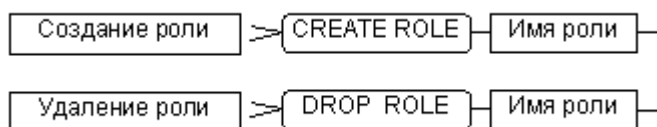


Рис. 6. Схемы создания/удаления роли

Общие правила:

- для создания роли необходимо иметь уровень прав RESOURCE;
- удалить роль может только ее создатель.
- вместе с ролью удаляются записи обо всех назначенных ей привилегиях и обо всех ее назначениях пользователям и другим ролям.

## Назначение/отмена назначения роли

Схемы назначения/отмены назначения роли приведены на рис. 7.

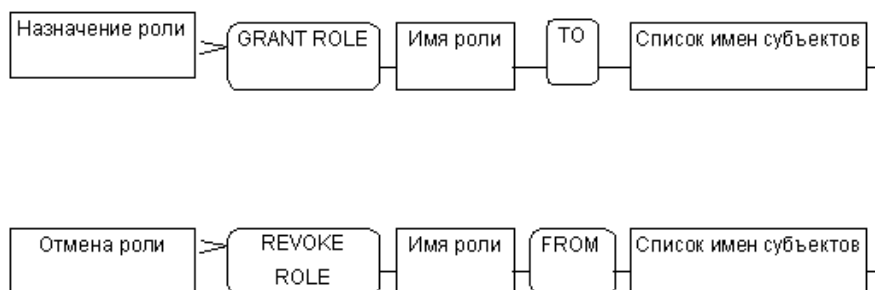


Рис. 7. Схемы назначения/отмены назначения роли

Общие правила:

- назначить роль (отменить назначение роли) может только ее владелец.

# Приложение

## Синтаксис команд для работы с комплексом средств защиты данных

### Уровень доступа

Создание:

```
CREATE LEVEL <имя уровня> = <№ уровня>;
```

Изменение имени уровня:

```
ALTER LEVEL <имя уровня> SET <новое имя уровня>;
```

### Группы доступа

Создание:

```
CREATE GROUP <имя группы> [= <числовой идентификатор группы>];
```

Изменение имени группы:

```
ALTER GROUP <имя группы> SET <новое имя группы>;
```

Уровни доверия между группами:

```
GRANT ACCESS ON <группа-доверитель> TO {<группа-получатель>|ALL} ;  
REVOKE ACCESS ON <группа-доверитель> FROM {<группа-получатель>|ALL};
```

### Модификация пользователя

Создание пользователя с категорией Connect по умолчанию:

```
CREATE USER <имя пользователя> IDENTIFIED BY <пароль>;
```

Создание пользователя с указанием категории доступа:

```
GRANT <категория> TO <имя пользователя> IDENTIFIED BY <пароль>;
```

Удаление пользователя:

```
DROP USER <имя пользователя>;
```

Изменение категории пользователя:

```
GRANT <категория> TO <имя пользователя> IDENTIFIED BY <пароль>;
```

Изменение пароля пользователя:

```
ALTER USER <имя пользователя> IDENTIFIED BY <новый пароль>;
```

Назначение группы доступа пользователю:

```
CREATE USER <имя пользователя> IDENTIFIED BY <пароль> GROUP <имя группы>;  
ALTER USER <имя пользователя> GROUP <имя группы>;
```

Назначение уровня доступа пользователю:

```
CREATE USER <имя пользователя> IDENTIFIED BY <пароль>  
LEVEL(<RAL>, <WAL>);  
ALTER USER <имя пользователя> LEVEL(<RAL>, <WAL>);
```

Назначение привилегий пользователю:

```
GRANT <привилегия> ON <имя объекта> TO <имя пользователя>;  
REVOKE <привилегия> ON <имя объекта> FROM <имя пользователя>;
```

### Роли

Создание:

```
CREATE ROLE <имя роли>;
```

Удаление:

```
DROP ROLE <имя роли>;
```

Назначение/отмена привилегии:

```
GRANT <привилегия> ON <имя объекта> TO <имя роли>;  
REVOKE <привилегия> ON <имя объекта> FROM <имя роли>;
```

Назначение/отмена:

```
GRANT ROLE <имя роли> TO <имя пользователя>;  
REVOKE ROLE <имя роли> FROM <имя пользователя>;
```

### Уровни доступа

Определение уровня доступа для таблиц:

```
CREATE TABLE <имя таблицы>(<имя столбца> <тип столбца>, [...])  
LEVEL(<RAL>,<WAL>);  
ALTER TABLE <имя таблицы> SET LEVEL(<RAL>,<WAL>);
```

Определение уровня доступа для столбца:

```
CREATE TABLE <имя таблицы>(<имя столбца> <тип столбца>  
LEVEL(<RAL>,<WAL>) [...]);  
ALTER TABLE <имя таблицы> SET COLUMN <имя столбца>  
LEVEL(<RAL>,<WAL>);
```

Использование групп и уровней в SQL- выражении:

```
INSERT INTO <имя таблицы>[#<группа>]#[<RAL>]#[<WAL>]]  
(<имя столбца>[#<группа>]#[<RAL>]#[<WAL>]]) VALUES (<значение>);  
UPDATE <имя таблицы>[#<группа>]#[<RAL>]#[<WAL>]]  
SET <имя столбца>[#<группа>]#[<RAL>]#[<WAL>]]=<значение> [...];
```

Чтение меток доступа:

```
SELECT SECURITY({*|<имя столбца>},{'R' |'W' |'G'}) FROM <имя таблицы>;
```

### Сетевая станция

Создание:

```
CREATE STATION <имя станции>  
PROTOCOL <сетевой протокол>  
{ ADDRESS <адрес станции>}  
[LEVEL (<RAL>,<WAL>)]  
[<рабочее время>;
```

<имя станции> ::= <идентификатор>  
 <сетевой протокол> ::= <символьный литерал>  
 <адрес станции> ::= <символьный литерал>  
 <RAL> ::= <идентификатор>  
 <WAL> ::= <идентификатор>  
 <рабочее время> ::=  
   { **ENABLE** | **DISABLE** } **LOGIN**  
   { **ALWAYS** | <график по времени> | <график по дням> }  
 <график по времени> ::=  
   **FROM** <время начала работы>  
   **TO** <время окончания работы> [ **FOR** <дни работы> ]  
 <график по дням> ::=  
   { **SINCE** <дата начала> } { **UNTIL** <дата окончания> }  
 <время начала работы> ::= 'HH:MM'  
 <время окончания работы> ::= 'HH:MM'  
 <дни работы> ::= <день недели> { [, <день недели> ] ... }  
 <день недели> ::= 'MON'|'TUE'|'WED'|'THU'|'FRI'|'SAT'|'SUN'  
 <дата начала> ::= 'DD.MM.YYYY'  
 <дата окончания> ::= 'DD.MM.YYYY'

Изменение параметров:

**ALTER STATION** <имя станции> [ **SET** <новое имя станции> ]  
 [ **LEVEL** (<RAL>, <WAL>) ] [ <рабочее время> ];

Удаление:

**DROP STATION** <имя станции>;

Регулирование доступа:

**GRANT ACCESS ON UNLISTED STATION TO** { <имя группы> | **ALL** };  
**REVOKE ACCESS ON UNLISTED STATION FROM** { <имя группы> | **ALL** };  
**GRANT ACCESS ON STATION** <имя станции> **TO** { <имя группы> | **ALL** };  
**REVOKE ACCESS ON STATION** <имя станции> **FROM** { <имя группы> | **ALL** };

## Устройства

Создание:

**CREATE DEVICE** <имя устройства> **DIRECTORY** <каталог>  
 [ **COMMENT** <комментарий> ] [ **LEVEL** (<RAL-устройства>, <WAL-устройства>) ];

Изменение параметров:

**ALTER DEVICE** <имя устройства> [ **DIRECTORY** <каталог> ]  
 [ **LEVEL** (<RAL-устройства>, <WAL-устройства>) ];

Удаление:

**DROP DEVICE** <имя устройства>;

Регулирование доступа:

**GRANT ACCESS ON UNLISTED DEVICE TO** { <имя группы> | **ALL** };  
**REVOKE ACCESS ON UNLISTED DEVICE FROM** { <имя группы> | **ALL** };  
**GRANT ACCESS ON DEVICE** <имя устройства> **TO** { <имя группы> | **ALL** };  
**REVOKE ACCESS ON DEVICE** <имя устройства> **FROM** { <имя группы> | **ALL** };

### Мониторинг КСЗ

Запуск протоколирования:

```
AUDIT START;
```

Останов протоколирования:

```
AUDIT STOP;
```

Управление протоколированием:

```
AUDIT {ENABLE|DISABLE|CLEAR} [<имя события> [ON [<объект БД>]]  
[FOR <имя пользователя>] [BY {SESSION|STATEMENT|ACCESS}]  
[WHEN [NOT] SUCCESS];
```

Параметры протоколирования:

```
AUDIT {SET | CANCEL} <параметры протоколирования>;  
<параметры протоколирования>::=  
RECORDS LIMIT <количество записей>|DAYS LIMIT <срок хранения>;
```

Протоколирование пользовательского сообщения

```
AUDIT MESSAGE <сообщение>;
```

Начать комментирование протоколируемых событий

```
AUDIT SET MESSAGE <комментарий>;
```

Отменить комментирование протоколируемых событий

```
AUDIT CANCEL MESSAGE;
```

# Указатель команд SQL-запросов

ALTER DEVICE, 32  
ALTER GROUP, 16  
ALTER LEVEL, 20  
ALTER STATION, 29  
AUDIT CANCEL, 56  
AUDIT CLEAR, 49  
AUDIT DISABLE, 49  
AUDIT ENABLE, 49  
AUDIT SET, 56  
AUDIT START, 10  
AUDIT STOP, 10  
CREATE DEVICE, 32  
CREATE GROUP, 16  
CREATE LEVEL, 20  
CREATE ROLE, 14  
CREATE STATION, 28  
CREATE USER, 14  
DROP DEVICE, 32  
DROP ROLE, 14  
DROP STATION, 29  
GRANT, 14  
GRANT ACCESS ON, 17  
GRANT ACCESS ON DEVICE, 33  
GRANT ACCESS ON STATION, 30  
GRANT ROLE, 14  
GROUP, 17  
LEVEL, 21  
REVOKE ACCESS ON, 17  
REVOKE ACCESS ON DEVICE, 33  
REVOKE ACCESS ON STATION, 30  
REVOKE ROLE, 14





