Задание

1. Изучить материал лекции, законспектировать.

2. Фотоотчет прислать на электронную почту

С уважением, Хвастова Светлана Ивановна

!!! Если возникнут вопросы обращаться по телефону 0721389311. Электронная почта: xvsviv@rambler.ru

Лекция на тему «Создание, обновление, удаление представлений. Представления и защита данных в SQL. Представление и отмена привилегий доступа к данным**»**

Наряду с привилегиями, ограничивающими доступ к таблицам, ключевую роль в защите данных играют также представления. Создавая представление и давая пользователю разрешение на доступ к нему, а не к исходной таблице, можно тем самым ограничить доступ пользователя, позволив ему обращаться только к данным столбцам и строкам. Таким образом, представления позволяют осуществлять четкий контроль над тем, какие данные доступны тому или иному пользователю. Предположим, например, что вы решили установить в базе данных следующее правило защиты данных:

Персонал финансового отдела имеет право извлекать из таблицы СОТРУДНИКИ идентификаторы и имена служащих, а также идентификаторы офисов; информация об объеме продаж и личных планах должна быть недоступна.

Это правило можно реализовать, создав представление

```
CREATE VIEW ВЕРТИКАЛЬНАЯ_ЗАЩИТА AS
SELECT ИДЕНТИФИКАТОР_СЛ, ИМЯ, ФАМИЛИЯ
FROM СОТРУДНИКИ
```

и назначить привилегию SELECT для этого представления пользователю с идентификатором КАДРЫ. В данном примере для ограничения доступа к отдельным столбцам используется *вертикальное представление.*

С помощью *горизонтальных представлений* также можно эффективно реализовывать правила защиты данных. Возьмем, к примеру, такое правило: начальники отделов должны иметь доступ ко всем данным, касающимся служащих только их отделов.

Для реализации этого правила следует создать представления, в которых будут содержаться данные таблицы СОТРУДНИКИ отдельно по каждому отделу, а затем каждому начальнику отдела дать разрешение на доступ к соответствующему представлению.

Конечно, представления могут быть гораздо сложнее, чем простое подмножество строк и столбцов одной таблицы, как в приведенных примерах. Создавая представление посредством запроса с группировкой, пользователю можно предоставить доступ к итоговым данным, а не к подробным данным исходной таблицы. В представлении можно объединять данные из двух или более таблиц, разрешая пользователю доступ именно к тем данным, которые ему необходимы, и запрещая доступ ко всем остальным. Использование представлений для защиты данных в SQL ограничивается двумя важными факторами:

- *Ограничения на обновление данных.* Если представление доступно только для выборки, то для него можно установить привилегию SELECT, но привилегии INSERT, DELETE и UPDATE не имеют для него смысла. Когда пользователь должен обновлять данные, видимые в доступном только для выборки представлении, ему следует дать разрешение на обновление исходных таблиц и он должен иметь возможность выполнять инструкции INSERT, DELETE и UPDATE по отношению к этим таблицам.
- *Производительность*. Так как СУБД преобразует каждое обращение к представлению в соответствующие обращения к исходным

таблицам, при использовании представлений может значительно увеличиться трудоемкость операций, выполняемых в базе данных. Поэтому представления нельзя неосмотрительно применять для ограничения доступа к данным — это неизбежно влечет за собой снижение обшей производительности СУБД.

Привилегии на столбцы

Стандарт SQL1 позволяет выдавать привилегию UPDATE на отдельные столбцы таблицы или представления, а стандарт SQL2 дает возможность предоставлять таким же образом привилегии INSERT и REFERENCES. Список столбцов располагается после ключевого слова UPDATE, INSERT или REFERENCES и заключается в круглые скобки. Вот, например, инструкция GRANT, разрешающая сотрудникам строевого отдела обновлять в таблице СОТРУДНИКИ только столбцы ВОИНСКОЕ_ЗВАНИЕ и СЕМЕЙНОЕ_ПОЛОЖЕНИЕ:

```
GRANT UPDATE(ВОИНСКОЕ_ЗВАНИЕ, СЕМЕЙНОЕ_ПОЛОЖЕНИЕ)
    ON СОТРУДНИКИ
    TO СТР_ОТДЕЛ
```

Примечание. Если список столбцов опущен, то привилегия действительна для всех столбцов таблицы или представления.

Согласно стандарту ANSI/ISO, не разрешается указывать список столбцов для привилегии SELECT; она должна распространяться на все столбцы таблицы или представления. Но на практике это не является серьезным ограничением. Чтобы обеспечить доступ не ко всем столбцам, на основе таблицы создается представление, включающее только необходимые столбцы, а затем выдается привилегия SELECT на это представление (как описывалось ранее). Однако представления, созданные исключительно для целей безопасности, могут заметно усложнить структуру простой базы

данных. По этой причине в некоторых СУБД разрешается указывать список столбцов для привилегий SELECT.

1.1.1 Предоставление и отмена привилегий на защищаемые объекты.

Когда пользователь создает базу данных, он автоматически получает право на администрирование этой базы. Для того, чтобы другие пользователи смогли иметь какие-то права на данную базу, эти права им должны быть явно предоставлены. Для этого используется оператор:

GRANT <тип права на БД> TO <имя пользователя>

Примечание: При управлении правами на уровне базы данных имя базы не указывается, подразумевается текущая.

Право на администрирование базы данных дает возможность и отбирать права у пользователей. Отбор права выполняется оператором:

REVOKE <тип права на БД> FROM <имя пользователя>

Примечание: Администратор базы данных не может лишить права на администрирование самого себя. Но он может лишиться этого права, если это выполнит другой пользователь имеющий право на администрирование.

Синтаксис оператора передачи права на таблицу:

GRANT <тип права на таблицу> ON <имя таблицы> TO <имя пользователя>

Примечание: В тех случаях, когда при передаче права на таблицу необходимо конкретизировать поля (права SELECT, UPDATE, REFERENCES), имена полей надо указать в скобках после названия права.

Пользователь, получивший право с помощью оператора GRANT…ON…TO…, не может передать это право другому пользователю. Для возможности передачи права другим пользователь должен получить право на его передачу. Это право указывается в операторе *GRANT* с помощью ключевых слов *WITH GRANT OPTION*:

GRANT <тип права на таблицу>(<имя поля>,…) ON <имя таблицы>
TO <имя пользователя> WITH GRANT OPTION

Отобрать право у пользователя можно лишь в том случае, если вы передали ему это право, или являетесь владельцем таблицы. Если вы отбираете право у пользователя, который получил его с правом на передачу. То автоматически этого права лишаются и те, кому пользователь это право передал.

Отбирание права осуществляется оператором:

REVOKE <тип права на таблицу> ON <имя таблицы> FROM <имя пользователя>

В реальной ИС может быть очень много пользователей. Следить за тем, чтобы каждому пользователю были даны только нужные ему права, достаточно сложно. Но, как правило, число категорий пользователей не так велико. SQL дает возможность управлять доступом не путем явного указания каждого конкретного пользователя, а путем создания роли, приписывания того или иного пользователя к конкретной роли и управления правами и привилегиями на уровне ролей.

Механизм ролей.

Роль – это группа пользователей с определенными правами.

Роль создается оператором:

CREATE ROLE <имя>

Имя роли должно быть не длиннее восьми символов и не должно совпадать с именем пользователя. Для указания пользователей, входящих в ту или иную роль, применяется оператор:

GRANT <имя роли> TO <имя пользователя1>, <имя пользователя2>,…

Для приписывания роли или отбирания у роли тех или иных прав и привилегий используются варианты операторов *GRANT* и *REVOKE*. рассмотренные выше.

Определять новые роли может только пользователь, являющийся администратором данной базы. Удалить роль, модифицировать список пользователей, входящих в ту или иную роль, а также приписать и отобрать у роли некоторые привилегии, может только администратор БД или пользователь, которому данная роль была передана с опцией WITH GRANT OPTION.

Использование механизма ролей позволяет упростить ведение списка пользователей, сделать администрирование системы более простым.