**Тема Определение прав доступа пользователей к данным**

1. **Пользователи и привилегии. Стандартные привилегии**

SQL обычно применяется в многопользовательских средах, которые требуют разграничения прав пользователей с точки зрения доступа к данным и прав на выполнение с ними тех или иных манипуляций. Для этих целей в SQL реализованы средства, позволяющие устанавливать и контролировать привилегии пользователей баз данных.

Каждый пользователь в среде SQL имеет специальное имя (идентификатор), с помощью которого осуществляется идентификация пользователя с целью установки и определения его прав с точки зрения доступа к данным. Каждая посланная к СУБД команда SQL-запроса ассоциируется СУБД с идентификатором доступа к данным конкретного пользователя.

Пользователь определяется с помощью следующей команды:

CREATE USER <username> IDENTIFIED BY <password>

Пользователь становится известен БД, но не может выполнять никаких операций.

Удаление пользователя производится командой:

DROUP USER<username>

Назначаемые привилегии –это разрешение на выполнение указанным пользователем данной команды над определенным объектом БД.

Имеется несколько типов привилегий, соответствующих нескольким типам операций.

Привилегии даются и отменяются двумя командами SQL:

GRANT –установка привилегий;

REVOKE-отмена привилегий.

Привилегии над объектом означают, что пользователь имеет право (привилегию) на выполнение данной команды только на определенном объекте в БД.

Привилегии объекта связаны одновременно и с пользователями, и с таблицами БД. То есть привилегия дается определенному пользователю в определенной таблице.

Это может быть, как базовая таблица, так и представление (т.е. таблица, полученная в результате запроса).

Пользователь, создавший таблицу любого вида, является **владельцем** этой таблицы.

То есть имеет все привилегии, относящиеся к таблице, может передавать привилегии на работу с этой таблицей другим пользователям.

Могут быть назначены следующие привилегии:

SELECT, INSERT, UPDATE, DELETE и другие.

Могут быть нестандартные привилегии объекта (INDEX, SYNONIM, ALTER, EXECUTE).

Назначение пользователям этих привилегий осуществляется с помощью команды GRANT.

1. **Команда GRANT, REVOKE**

Пользователь, являющийся владельцем таблицы STUDENT, может передать другому пользователю (IVANOV) привилегию SELECT с помощью следующей команды:

GRANT SELECT ON STUDENT TO IVANOV;

Без наличия других привилегий IVANOV может только выбирать (SELECT) значения, но не может выполнять любые действия, которые бы воздействовали на значения в таблице STUDENT, включая использование таблицы в качестве родительской таблицы внешнего ключа.

Для команды GRANT сначала проверяется допустимость этой команды для давшего ее пользователя. IVANOV дать этой команды не может.

Он также не может предоставить право SELECT другому пользователю, так как таблица принадлежит не ему. Команда:

GRANT INSERT ON EXAM\_MARKS TO IVANOV;

позволяет вводить в таблицу новые строки.

Можно передавать несколько привилегий нескольким пользователям:

GRANT SELECT, INSERT ON SUBJECT

TO IVANOV, PETROV;

Однако, в ANSI **невозможно предоставить привилегии для нескольких таблиц** одной командой GRANT.

Чтобы предоставить все привилегии в таблице используют команду ALL PRIVILEGES или ALL.

Например,

GRANT ALL PRIVILEGES ON STUDENT TO IVANOV;

Или:

GRANT ALL ON STUDENT TO IVANOV;

Передает пользователю Иванов все привилегии в таблице Студент.

Аргумент PUBLIC используется для передачи указанных в команде привилегий всем остальным пользователям. Чаще применяется для привилегии SELECT в базовых таблицах или представлениях, которые необходимо сделать доступными пользователям:

GRANT SELECT ON EXAM\_MARKS TO PUBLIC;

Позволяет любому пользователю получить информацию из таблицы EXAM\_MARKS.

Предоставление **всех** привилегий в таблице **всем** пользователям обычно является нежелательным. Так как это может вызвать определенные проблемы обеспечения безопасности и защиты данных. Поэтому для ограничения доступа к таблице лучше всего предоставить привилегии, отличные от SELECT, только индивидуальным пользователям.

Отмена привилегии осуществляется с помощью команды REVOKE, синтаксис аналогичный GRANT.

1. **Другие типы привилегий**

Теперь следует рассмотреть следующие вопросы:

-кто имеет право создавать таблицы;

-изменять, удалять, ограничивать таблицы;

-должны ли права создания базовых таблиц отличаться от прав создания представлений;

-должен ли существовать суперпользователь, т.е. пользователь, отвечающий за поддержку БД, имеющий наибольшие (полные привилегии), которые не предоставляются обычному пользователю.

Привилегии, которые не определяются в терминах специальных объектов данных, называются *привилегиями системы или правами БД.*

Эти привилегии включают в себя право создавать объекты данных, отличающиеся от базовых таблиц (обычно создаваемых несколькими пользователями) и представлений (создаваемых большинством пользователей).

Привилегии системы должны дополнять, а не заменять привилегии объекта, которые стандарт требует от создателей представлений.

В любой системе всегда имеются некоторые типы суперпользователей. То есть пользователей, которые имеют большинство или все привилегии и могут передать их (администратор БД).

Общий подход определяет три базовые привилегии системы:

CONNECT-подключить;

RESOURCE-ресурс;

DBA-администратор БД.

CONNECT-право зарегистрироваться и права создать представления и синонимы;

RESOURCE-создать базовые таблицы;

DBA-привилегия администратора, может иметь одного или более пользователей.

В большинстве реализаций SQL нового пользователя создает DBA, т.е. администратор БД, автоматически предоставляющий новому пользователю привилегию CONNECT.

В этом случае добавляется IDENTIFIED BY ‘PASSWORD’.

Например,

GRANT CONNECT TO PETROV IDENTIFIED BY ‘PASSWORD’;