Раздел 4. Основы языка SQL

Тема 4.2

Язык определения данных (Data Definition Language)

Вопросы лекции:

1. Общая характеристика основных объектов БД.
2. Язык определения данных (DDL).
3. Создание таблиц и их элементов.
4. Изменение и удаление таблиц БД.

Группы операторов языка SQL-2003.

- Операторы определения данных – **Data Definishion Language** (DDL).
- Оператор запросов (выборки) данных - **Data Query Language** (DQL).
- Операторы манипулирования данными – **Data Manipulation Language** (DML).
- Операторы управления транзакциями.
- Операторы администрирования СУБД и данными.

Основные объекты БД:

1. База данных (DATABASE) – контейнер, содержащий все остальные объекты.
2. Схема (SCHEMA) – часть БД, в пределах которой все имена создаваемых объектов должны быть уникальны. **В разных схемах одной и той же БД могут быть одинаковые имена.** Схемы поддерживаются не всеми СУБД.
3. Таблица (TABLE) – основной объект БД.
4. Индекс (INDEX) – вспомогательный объект для ускорения поиска (замедляет операции модификации).
5. Представление (VIEW) – именованный запрос на выборку, хранящийся в БД (виртуальная таблица).
6. Хранимая процедура или функция (STORAGE PROCEDURE, FUNCTION) – именованные конструкции из операторов SQL, хранящиеся в БД.
7. Триггер (TRIGGER) – особый вид ХП, срабатывающий автоматически при наступлении заданных событий

Дополнительные объекты БД:

7. Пользователь и роль (USER, ROLE) – пользователи и их права (разрешения) на выполнение действий в БД.
8. Последовательность (SEQUENCE) – в некоторых СУБД (Oracle) служат для генерации уникальных значений первичных ключей таблиц.
9. Связь, снимок, синоним (LINK, SNAPSHOT, SYNONIM) – используются для организации распределенных БД.

Снимок (объект SNAPSHOT) – таблица или представление, которое посылается на удаленный сервер и периодически обновляется в автоматическом режиме.

Снимок, в отличие от представления, это реальная физическая таблица, хранящаяся на удаленном сервере, которая позволяет избежать многочисленных запросов пользователей удаленного сервера к данным, хранящемся на основном сервере. Важно – снимок не может обновляться очень часто, поэтому в какие-то моменты его данные могут отличаться от актуальных.

Раздел 4. Основы языка SQL

Тема 4.2

Язык определения данных (Data Definition Language)

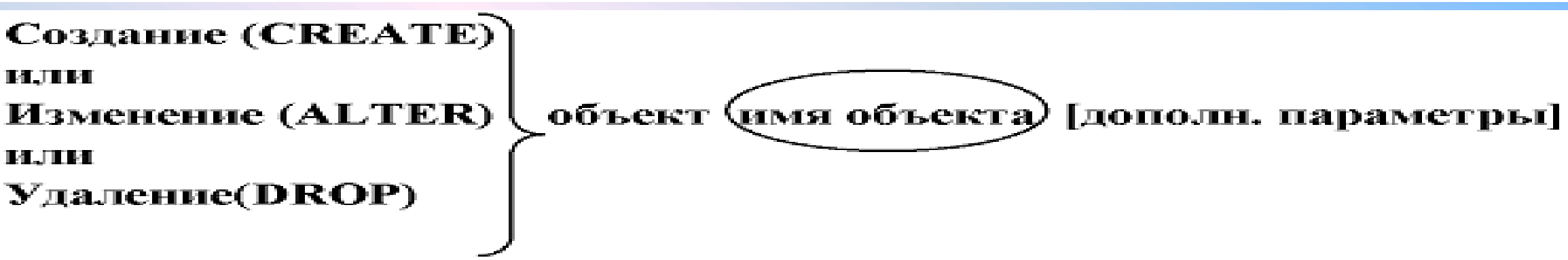
Вопросы лекции:

1. Общая характеристика основных объектов БД.
2. Язык определения данных (DDL).
3. Создание таблиц и их элементов.
4. Изменение и удаление таблиц БД.

Язык DDL служит для **создания, удаления и модификации** всех **объектов,** входящих в состав базы данных.

Он содержит три команды, которые могут быть применены к различным **объектам базы данных**.

В общем виде команду DDL можно определить так:



Примеры:

DROP TABLE t - удаление таблицы с именем t, здесь объектом является TABLE, его имя t.

Данная команда не требует никаких дополнительных параметров

CREATE TABLE tt (n NUMBER, x VARCHAR(50)) – создание таблицы с именем tt, в качестве дополнительных параметров задаются определения столбцов, в данном прмере их два: столбец n имеет цифровой тип, а столбец x - текстовый тип, причем длина текста не превышает 50 символов.

Примеры операторов определения данных (**DDL**)

CREATE TABLE	Создать таблицу	Создает новую таблицу в БД
DROP TABLE	Удалить таблицу	Удаляет таблицу из БД
ALTER TABLE	Изменить таблицу	Изменяет структуру существующей таблицы или ограничения целостности, задаваемые для данной таблицы
CREATE VIEW	Создать представление	Создает виртуальную таблицу, соответствующую некоторому SQL-запросу
ALTER VIEW	Изменить представление	Изменяет ранее созданное представление
DROP VIEW	Удалить представление	Удаляет ранее созданное представление
CREATE INDEX	Создать индекс	Создает индекс для некоторой таблицы для обеспечения быстрого доступа по атрибутам, входящим в индекс
DROP INDEX	Удалить индекс	Удаляет ранее созданный индекс

Раздел 4. Основы языка SQL

Тема 4.2

Язык определения данных (Data Definition Language)

Вопросы лекции:

1. Общая характеристика основных объектов БД.
2. Язык определения данных (DDL).
3. Создание таблиц и их элементов.
4. Изменение и удаление таблиц БД.

Формат команды для создания новой таблицы:

CREATE TABLE имя_таблицы
(список_определений_столбцов [, список_ограничений_столбцов])

Элементы списка всегда **разделяются запятыми**.

Определение каждого столбца имеет вид:

Имя_столбца тип_столбца [DEFAULT значение_по_умолчанию] [ограничения на столбец]

Обязательными элементами описания столбца являются **имя столбца и тип данных** в нем.

Имя столбца должно быть **уникальным в пределах таблицы**.

Чтобы создать таблицу с помощью оператора CREATE TABLE, нужно указать следующие данные:

- • имя новой таблицы; оно вводится после ключевого слова CREATE TABLE;
- • имена и определения столбцов таблицы, разделенные запятыми;
- • в некоторых СУБД также требуется, чтобы было указано место размещения таблицы.

```
CREATE TABLE Products
(
prod_id CHAR(10) NOT NULL,
vend_id CHAR(10) NOT NULL,
prod_name CHAR(254) NOT NULL,
prod_price DECIMAL(8,2) NOT NULL,
prod_desc VARCHAR(1000) NULL
 );
```

В некоторых СУБД могут быть различия в типах данных!

Также может потребоваться указание имени БД – USES XXX

Язык SQL при создании таблиц позволяет определять значения по умолчанию, которые будут использованы в том случае, если при добавлении строки какое-то ее значение не указано.

Значения по умолчанию определяются с помощью ключевого слова DEFAULT в определениях столбца оператора CREATE TABLE.

```
CREATE TABLE OrderItems
(
order_num INTEGER NOT NULL,
order_item INTEGER NOT NULL,
prod_id CHAR(10) NOT NULL,
quantity INTEGER NOT NULL DEFAULT 1,
item_price DECIMAL (8 ,2) NOT NULL
);
```

Ограничения (constraints) в таблицах

Ограничения позволяют задавать дополнительные условия проверки вводимых данных, проверяемые СУБД автоматически в момент их ввода.

Ограничения могут быть заданы **для** отдельных **столбцов** или **таблицы** в целом.

Любое ограничение может быть поименовано (рекомендуется), или **имя присваивается СУБД автоматически**.

Для явного именования к описанию ограничения следует добавить слева фразу :

CONSTRAINT Имя_ограничения Описание ограничения.

Ограничения (constraints) в таблицах

Ограничения столбца:

NOT NULL/NULL – допустимы ли пустые (неопределенные) значения в столбце, по умолчанию используется значение NULL (т.е. допустимы), что в большинстве случаев не соответствует бизнес-правилам предметной области, поскольку пропуски какой-либо нужной информации обычно недопустимы. Данное ограничение не именуется.

PRIMARY KEY – ограничение первичного ключа, при этом автоматически накладывается ограничение NOT NULL. При задании значений первичного ключа любое значение проверяется на уникальность и при обнаружении дубликата операция прерывается. В таблице может быть только один столбец с ограничением PRIMARY KEY.

UNIQUE – ограничение уникальности (альтернативный ключ), ограничение NOT NULL также накладывается автоматически. Фактически, UNIQUE ничем не отличается от PRIMARY KEY, однако количество столбцов с UNIQUE не ограничено.

CHECK – проверка логических выражений при изменении данных в столбце, например, CHECK (имя_столбца BETWEEN 1 AND 100).

Ограничения (constraints) в таблицах

Ограничения столбца:

REFERENCES имя_главной_таблицы — внешний ключ, который задается для столбца подчиненной (детальной) таблицы. Для значений внешнего ключа автоматически выполняется проверка на существование равного значения первичного ключа главной таблицы. При определении внешнего ключа могут быть дополнительно определены правила обеспечения ссылочной целостности. Например, правилами для удаления являются:

- ON DELETE RESTRICT – запретить удаление строки главной таблицы, если в подчиненной есть хотя бы одна строка, которая на нее ссылается;
- ON DELETE CASCADE – вместе со строкой главной таблицы автоматически удалить все ссылающиеся на нее строки подчиненной таблицы;
- ON DELETE SET NULL – при удалении строки главной таблицы установить во внешнем ключе ссылающихся строк значение NULL (это правило может быть определено только в том случае, если на внешний ключ не наложено ограничение NOT NULL);
- ON DELETE SET DEFAULT – при удалении строки главной таблицы установить во внешнем ключе значение по умолчанию (это правило может быть определено только в том случае, если при определении внешнего ключа задано DEFAULT значение_по_умолчанию).

По умолчанию принят запрет удаления строк при наличии ссылок на них (ON DELETE RESTRICT).

Например:

```
CREATE TABLE t
(c1 NUMBER(8) DEFAULT 0 NOT NULL CONSTRAINT un UNIQUE,
 c2 VARCHAR(100) NOT NULL
)
```

Создается таблица с именем t, содержащая два столбца: числовой столбец c1, обязательный для заполнения и принимающий значение 0 по умолчанию, все значения этого столбца уникальны (ограничение уникальности имеет имя un) и текстовый столбец c2, обязательный для заполнения текстом, его размер не более 100 символов.

Ограничения (constraints) в таблицах

Ограничения таблицы

Данные ограничения используются в том случае, если они затрагивают сразу несколько столбцов:

PRIMARY KEY (список столбцов) — составной первичный ключ;

UNIQUE (список столбцов) — составной альтернативный ключ;

FOREIGN KEY имя_внешнего_ключа (список столбцов) REFERENCES имя_главной_таблицы [правила поддержки ссылочной целостности];

CHECK (логическое выражение, затрагивающее сразу несколько столбцов).

Например:

```
CREATE TABLE t1
( c1 NUMBER(3) NOT NULL,
  c2 DATE NOT NULL,
  c3 NUMBER(3) NOT NULL,
  CONSTRAINT pk_t1 PRIMARY KEY(c1,c2),
  CONSTRAINT ck_t1 CHECK(c1+c3<=200)
)
```

Создается таблица с именем t1, содержащая три столбца, обязательных для заполнения. Составной ключ таблицы включает столбцы c1 и c2 (обратим внимание, что каждый из столбцов таблицы не обладает свойствами уникальности, поэтому PRIMARY KEY не может быть ограничением одного столбца). Ограничение CHECK также затрагивает сразу два столбца, поэтому оформлено как ограничение таблицы.

Раздел 4. Основы языка SQL

Тема 4.2

Язык определения данных (Data Definition Language)

Вопросы лекции:

1. Общая характеристика основных объектов БД.
2. Язык определения данных (DDL).
3. Создание таблиц и их элементов.
4. Изменение и удаление таблиц БД.

Изменение структуры таблиц и их удаление

Для того чтобы обновить (изменить) определения таблицы, следует воспользоваться оператором ALTER TABLE.

Хотя все СУБД поддерживают этот оператор, то, что они при этом позволяют вам делать, в значительной степени зависит от реализации СУБД.

Добавление столбцов в таблицу — единственная операция, поддерживаемая всеми СУБД.

Чтобы изменить таблицу посредством оператора **ALTER TABLE**, нужно ввести следующую информацию.

- • **Имя таблицы**, подлежащей изменению, после ключевых слов ALTER TABLE.
  (Таблица с таким именем должна существовать, иначе будет выдано сообщение об ошибке.)
- • **Список изменений**, которые должны быть сделаны.

Операции изменения: добавление, изменение или удаление столбцов, введение ограничений или ключей.

ALTER TABLE Stud **ADD** [COLUMN] vend_phone CHAR(20);

ALTER TABLE Stud **DROP** COLUMN vend_phone

Удаление таблиц и изменение их структуры

Команда удаления таблицы имеет вид:

DROP имя_таблицы

Нельзя удалить таблицу, если существуют внешние ключи, ссылающиеся на эту таблицу.

Вместе с таблицей **удаляются и все** созданные для нее **индексы и триггеры**.

Удаление отдельного **столбца** из таблицы производится не напрямую, а через параметры команды **ALTER TABLE**

```
DROP COLUMN имя_столбца
DROP [CONSTRAINT] ограничение
```

Например:

```
ALTER TABLE t DROP PRIMARY KEY(n)
```

или

```
ALTER TABLE t DROP CONSTRAINT pk_t
ALTER TABLE t DROP COLUMN n
```

Сложные изменения структуры таблицы обычно выполняются вручную и включают следующие шаги:

- • Создание новой таблицы с новым расположением столбцов.
- • Использование оператора INSERT SELECT (Добавление данных, для копирования данных из старой таблицы в новую. При необходимости используются функции преобразования и вычисляемые поля).
- • Проверка того факта, что новая таблица содержит нужные данные.
- • Переименование старой таблицы (или удаление ее).
- • Присвоение новой таблице имени, которое ранее принадлежало старой таблице.
- • Восстановление триггеров, хранимых процедур, индексов и внешних ключей, если это необходимо.