## 15. Использование команд DDL для создания и управления таблицами, ограничениями, представлениями, последовательностями, индексами, синонимами, типы данных.

## Создание таблицы

Чтобы создать таблицу в базе данных, администратор базы данных должен иметь под рукой определенную информацию – имя таблицы, имя столбца, типы данных столбца и размеры столбца. Вся эта информация может быть изменена позже с помощью команд DDL.

## CREATE TABLE заявление

CREATE TABLE – это оператор DDL, который используется для создания таблиц в базе данных. Таблица создается после выполнения сценария CREATE TABLE и готова хранить данные в дальнейшем. Пользователь должен иметь системную привилегию CREATE TABLE для создания таблица в своей собственной схеме. Но чтобы создать таблицу в схеме любого пользователя, пользователь должен иметь схему CREATE ANY TABLE.

Вот синтаксис основного оператора CREATE TABLE. Может быть много дополнительных предложений, чтобы явно предоставить спецификации хранилища или значения сегмента.

CREATE TABLE [schema.]table

( { column datatype [DEFAULT expr] [column\_constraint] ...

| table\_constraint}

[, { column datatype [DEFAULT expr] [column\_constraint] ...

| table\_constraint} ]...)

[AS subquery]

В приведенном выше синтаксисе DEFAULT указывает значение по умолчанию, которое можно использовать во время оператора INSERT, если столбец игнорируется. Он не может содержать ссылки на другие столбцы таблицы или псевдостолбцы (CURRVAL, NEXTVAL, LEVEL и ROWNUM), за исключением SYSDATE и USER или констант даты, которые не указаны полностью.

Ограничения – это правила, определенные по выбору на уровне столбца или на уровне таблицы (рассматриваются далее в этой главе). Эти правила проверяются во время любого действия с данными (вставка, обновление) в таблице и выдают ошибку, чтобы отменить действие при его нарушении.

Например, инструкция CREATE TABLE ниже создает таблицу EMP\_TEST. Обратите внимание на характеристики столбца, тип данных и точность.

CREATE TABLE SCOTT.EMP\_TEST

(EMPID NUMBER,

ENAME VARCHAR2(100),

DEPARTMENT\_ID NUMBER,

SALARY NUMBER,

JOB\_ID VARCHAR2(3),

HIREDATE DATE,

COMM NUMBER);

Пользователь может ссылаться на таблицы из схемы другого пользователя, добавляя к имени таблицы префикс имени пользователя или схемы. Например, пользователь GUEST желает запросить имя и зарплату сотрудника из таблицы EMP\_TEST, которая принадлежит SCOTT. Он может выдать следующий запрос –

SELECT ENAME, SALARY,

FROM GUEST.EMP\_TEST;

Столбец может содержать значение по умолчанию во время создания таблицы. Это помогает ограничить значения NULL, попадающие в столбец. Значение по умолчанию может быть выведено из литерала, выражения или функции SQL, которая должна возвращать в столбец совместимый тип данных. В приведенном ниже операторе CREATE TABLE обратите внимание, что столбец LOCATION\_ID имеет значение по умолчанию 100.

CREATE TABLE SCOTT.DEPARTMENT

(DEPARTMENT\_ID NUMBER,

DNAME VARCHAR2 (100),

LOCATION\_ID NUMBER DEFAULT 100);

## Типы данных

Типы данных используются для указания основного поведения столбца в таблице. В более широком смысле поведение столбца может принадлежать как числу, так и символу или семейству дат. Существует несколько других подтипов, принадлежащих этим семействам.

## Тип данных номера

Тип данных NUMBER включает в себя числовые значения как целых чисел, так и чисел с фиксированной запятой. В ранних версиях Oracle определялись разные типы данных для каждого из этих различных типов чисел, но теперь тип данных NUMBER служит всем этим целям. Выбирайте тип данных NUMBER, когда столбец должен хранить числовые данные, которые могут использоваться в математических вычислениях. Иногда тип данных NUMBER используется для хранения идентификационных номеров, в которых эти числа генерируются СУБД в виде последовательных чисел.

NUMBER (p, s), где p – точность до 38 цифр, а s – шкала (количество цифр справа от десятичной точки). Шкала может находиться в диапазоне от -84 до 127.

NUMBER (p) – это число с фиксированной точкой со шкалой от нуля и точностью до p.

FLOAT [(p)], где p – двоичная точность, которая может варьироваться от 1 до 126. Если p не указано, значением по умолчанию является двоичное 126.

## Тип данных даты

Для каждого типа данных DATE в базе данных хранятся век, год, месяц, день, час, минута, секунда. Каждая система баз данных имеет формат даты по умолчанию, который определяется параметром инициализации NLS\_DATE\_FORMAT. Этот параметр обычно имеет значение DD-MON-YY. Если вы не укажете время, по умолчанию используется время 12:00:00.

## Тип символьных данных

Oracle поддерживает три предопределенных символьных типа данных, включая CHAR, VARCHAR, VARCHAR2 и LONG.VARCHAR и VARCHAR2 фактически являются синонимами, и Oracle рекомендует использовать VARCHAR2 вместо VARCHAR. Используйте тип данных CHAR, когда столбец будет хранить символьные значения фиксированной длины. Например, номер социального страхования (SSN) в Соединенных Штатах присваивается каждому гражданину и всегда имеет размер 9 символов (хотя SSN строго состоит из цифр, цифры обрабатываются как символы) и будет указан как CHAR (9). Используйте тип данных VARCHAR2 для хранения буквенно-цифровых данных переменной длины. Например, имя клиента или адрес могут значительно различаться в зависимости от количества символов, которые должны быть сохранены. Максимальный размер столбца VARCHAR2 составляет 4000 символов.

## Тип данных LOB

Oracle предоставляет несколько различных типов данных LOB, включая CLOB (большой символьный объект) и BLOB (большой двоичный объект). Столбцы этих типов данных могут хранить неструктурированные данные, включая текстовые, графические, видео и пространственные данные. Тип данных CLOB может хранить до восьми терабайт символьных данных с использованием набора символов базы данных CHAR. Тип данных BLOB используется для хранения неструктурированных двоичных больших объектов, таких как объекты, связанные с данными изображения и видео, где данные представляют собой просто поток «битовых» значений. Тип данных BLOB может хранить до восемь терабайт двоичных данных. Тип данных NCLOB может хранить большие символьные объекты в многобайтовых национальных символах размером от 8 ТБ до 128 ТБ. Значение типа данных BFILE работает как локатор файлов или указатель на файл в файловой системе сервера. Максимальный поддерживаемый размер файла составляет от 8 ТБ до 128 ТБ.

**Ограничения**

Ограничения – это набор правил, определенных в таблицах Oracle для обеспечения целостности данных. Эти правила применяются для каждого столбца или набора столбцов. Когда таблица участвует в действии данных, эти правила проверяются и выдают исключение при нарушении. Доступные типы ограничений: NOT NULL, первичный ключ, уникальный, проверка и внешний ключ.

Приведенный ниже синтаксис может использоваться для наложения ограничений на уровне столбцов.

**Синтаксис:**

column [data type] [CONSTRAINT constraint\_name] constraint\_type

Все ограничения, кроме NOT NULL, также могут быть определены на уровне таблицы. Составные ограничения могут быть указаны только на уровне таблицы.

**NOT NULL Ограничение**

Ограничение NOT NULL означает, что строка данных должна иметь значение для столбца, указанного как NOT NULL. Если столбец указан как NOT NULL, СУБД Oracle не позволит хранить строки в таблице сотрудников, которые нарушают это ограничение. может быть определено только на уровне столбца, а не на уровне таблицы.

**Синтаксис:**

COLUMN [data type] [NOT NULL]

**Уникальное ограничение**

Иногда необходимо обеспечить уникальность для значения столбца, которое не является столбцом первичного ключа. Ограничение UNIQUE может использоваться для применения этого правила, и Oracle будет отклонять любые строки, которые нарушают уникальное ограничение. Ограничение Уникальность гарантирует, что значения столбца различны без каких-либо дубликатов.

**Синтаксис:**

**Уровень столбца:**

COLUMN [data type] [CONSTRAINT <name>] [UNIQUE]

**Уровень таблицы:** CONSTRAINT [имя ограничения] UNIQUE (имя столбца)

Примечание: Oracle внутренне создает уникальный индекс для предотвращения дублирования в значениях столбца. Индексы будут обсуждаться позже в PL / SQL.

CREATE TABLE TEST

( ... ,

NAME VARCHAR2(20)

CONSTRAINT TEST\_NAME\_UK UNIQUE,

... );

В случае составного уникального ключа он должен быть определен на уровне таблицы, как показано ниже.

CREATE TABLE TEST

( ... ,

NAME VARCHAR2(20),

STD VARCHAR2(20) ,

CONSTRAINT TEST\_NAME\_UK UNIQUE (NAME, STD)

);

**Оператор для изменения таблицы**

## Оператор ALTER TABLE

Администратор базы данных может вносить изменения в структуру таблицы или определения столбцов после того, как таблица создана в базе данных. Для выполнения таких действий используется команда DDL ALTER TABLE. Команда Alter предоставляет несколько утилит, исключающих объекты схемы. Используется оператор ALTER TABLE. добавить, удалить, переименовать и изменить столбец в таблице.

Приведенный ниже оператор ALTER TABLE переименовывает таблицу EMP в EMP\_NEW.

ALTER TABLE EMP RENAME TO EMP\_NEW;

Приведенный ниже оператор ALTER TABLE добавляет новый столбец TESTCOL в таблицу EMP\_NEW.

ALTER TABLE EMP\_NEW ADD (TESTCOL VARCHAR2 (100))

Приведенный ниже оператор ALTER TABLE переименовывает столбец TESTCOL в TESTNEW.

ALTER TABLE EMP\_NEW RENAME COLUMN TESTCOL TO TESTNEW

Приведенный ниже оператор ALTER TABLE удаляет столбец TESTNEW из таблицы EMP\_NEW

ALTER TABLE EMP\_NEW DROP COLUMN TESTNEW;

Приведенный ниже оператор ALTER TABLE добавляет первичный ключ в столбец EMPLOYEE\_ID.

ALTER TABLE EMP\_NEW ADD PRIMARY KEY (EMPLOYEE\_ID)

Приведенный ниже оператор ALTER TABLE удаляет первичный ключ.

ALTER TABLE EMP\_NEW DROP PRIMARY KEY;

Приведенный ниже оператор ALTER TABLE переключает режим таблицы только для чтения.

ALTER TABLE EMP\_NEW READ ONLY;

## DROP TABLE заявление

Оператор DROP TABLE используется для удаления таблицы из базы данных. Удаленная таблица и ее данные больше не доступны для выбора. Удаленную таблицу можно восстановить с помощью утилиты FLASHBACK, если она доступна в recyclebin. Удаление таблицы удаляет индекс и связанные с ним триггеры.

### Синтаксис:

DROP TABLE [TABLE NAME] [PURGE]

Приведенный ниже оператор удалит таблицу и поместит ее в корзину.

DROP TABLE emp\_new;

Приведенный ниже оператор удалит таблицу и также очистит ее от корзины.