

**Продолжаем учиться**

# Использование команд DDL для создания и управления таблицами

# Рассматриваемые вопросы

- Главные объекты базы данных
- Просмотр структуры таблицы
- Типы данных, которые могут использоваться в определениях столбцов
- Создание простой таблицы
- Изменение определений таблиц
- Создание ограничений вместе с таблицей
- Описание использования объектов схемы

# Объекты базы данных

Объект	Описание
Таблица	Основная единица хранения; состоит из строк и столбцов
Представление	Логически представляет подмножества данных из одной или нескольких таблиц
Последовательность	Генератор числовых значений
Индекс	Увеличивает производительность некоторых запросов
Синоним	Дает альтернативные имена объектам

# Правила присвоения имен

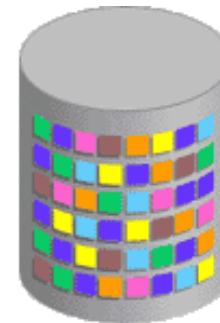
- Имена таблиц и столбцов :
  - Должны начинаться с буквы
  - Могут быть длиной от 1 до 30 символов
  - Должны содержать только символы A–Z, a–z, 0–9, \_, \$ и #
  - Не должны совпадать с именем другого объекта, принадлежащего этому же самому пользователю
  - Не должны совпадать со словом, зарезервированным сервером Oracle

# Команда CREATE TABLE

- Необходимо иметь :
  - привилегию CREATE TABLE
  - Область хранения

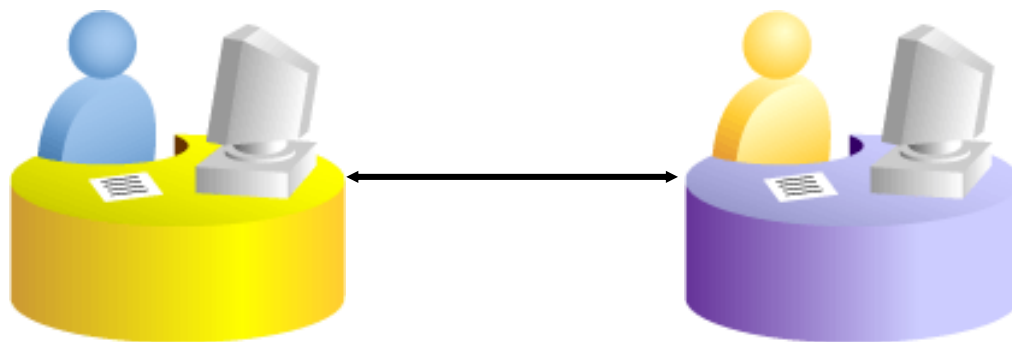
```
CREATE TABLE [схема.] таблица  
    (столбец тип_данных [DEFAULT выражение] [, ...]) ;
```

- Вы задаете:
  - Имя таблицы
  - Имя столбца, тип данных столбца и размер столбца



# Ссылки на таблицы других пользователей

- Таблицы, принадлежащие другим пользователям, не входят в схему пользователя.
- В качестве префикса в имени таблицы следует указать имя владельца.



Пользователь USERA

```
SELECT *  
FROM userB.employees;
```

Пользователь USERB

```
SELECT *  
FROM userA.employees;
```

# Опция DEFAULT

- Задаёт значение по умолчанию, если при вставке данных значение не указано явно.

```
... hire_date DATE DEFAULT SYSDATE, ...
```

- В качестве значения допускается литерал, выражение или функция SQL.
- Не может использоваться имя другого столбца или псевдостолбец.
- Тип данных, используемый по умолчанию, должен совпадать с типом данных столбца.

```
CREATE TABLE hire_dates  
    (id          NUMBER(8),  
      hire_date DATE DEFAULT SYSDATE);
```

Table created.



# Создание таблиц

- Создание таблицы

```
CREATE TABLE dept
    (deptno      NUMBER(2) ,
     dname       VARCHAR2(14) ,
     loc         VARCHAR2(13) ,
     create_date DATE DEFAULT SYSDATE) ;
```

```
CREATE TABLE succeeded.
```

# Типы данных

<code>VARCHAR2 (size)</code>	Символьные данные переменной длины
<code>CHAR (size)</code>	Символьные данные постоянной длины
<code>NUMBER (p, s)</code>	Числовые данные переменной длины
<code>DATE</code>	Значения даты и времени
<code>LONG</code>	Символьные данные переменной длины до 2 гигабайтов
<code>CLOB</code>	Символьные данные до 4 гигабайтов
<code>RAW and LONG RAW</code>	Необработанные (“сырые”) двоичные данные
<code>BLOB</code>	Двоичные данные до 4 гигабайтов
<code>BFILE</code>	Двоичные данные, которые хранятся во внешнем файле; длина до 4 гигабайтов
<code>ROWID</code>	Уникальный адрес строки в таблице в 64-ой символьной системе кодировки

# Типы данных для хранения даты и времени

Тип данных	Описание
TIMESTAMP	Дата до долей секунды
INTERVAL YEAR TO MONTH	Хранит интервал времени в годах и месяцах
INTERVAL DAY TO SECOND	Хранит интервал времени в днях, часах, минутах и секундах



# Ограничения

- Ограничения обеспечивают выполнение правил на уровне таблицы
- Ограничения предотвращают удаление таблицы при наличии подчиненных данных в других таблицах
- В Oracle допускаются следующие виды ограничений:
  - NOT NULL
  - NULL
  - UNIQUE
  - PRIMARY KEY
  - FOREIGN KEY
  - CHECK

# Указания по ограничениям

- Присваивайте имена ограничениям сами; в противном случае сервер Oracle присвоит имя в формате SYS\_Cn.
- Создавайте ограничения:
  - при создании таблицы;
  - после создания таблицы.
- Устанавливайте ограничения на уровне столбца или таблицы.
- Просматривайте ограничения в словаре данных.

# Определение ограничений

- Синтаксис:

```
CREATE TABLE [схема.] таблица  
    (столбец тип_данных [DEFAULT выражение]  
    [ограничение_на_уровне_столбца],  
    ...  
    [ограничение_на_уровне_таблицы] [, ...]) ;
```

- Ограничение на уровне столбца:

```
столбец [CONSTRAINT имя_ограничения] тип_ограничения,
```

- Ограничение на уровне таблицы:

```
столбец, ...  
[CONSTRAINT имя_ограничения] тип_ограничения  
(столбец, ...),
```

# Определение ограничений

- Ограничение на уровне столбца:

```
CREATE TABLE employees (  
    employee_id  NUMBER(6)  
    CONSTRAINT emp_emp_id_pk PRIMARY KEY,  
    first_name   VARCHAR2(20),  
    ...);
```

1

- Ограничение на уровне таблицы:

```
CREATE TABLE employees (  
    employee_id  NUMBER(6),  
    first_name   VARCHAR2(20),  
    ...  
    job_id       VARCHAR2(10) NOT NULL,  
    CONSTRAINT emp_emp_id_pk  
        PRIMARY KEY (EMPLOYEE_ID) );
```

2

# Ограничение NOT NULL

- Предотвращает появление неопределенных значений в столбце

EMPLOYEE_ID	FIRST_NAME	LAST_NAME	SALARY	COMMISSION_PCT	DEPARTMENT_ID	EMAIL	PHONE_NUMBER	HIRE_DATE
100	Steven	King	24000	(null)	90	SKING	515.123.4567	17-JUN-87
101	Neena	Kochhar	17000	(null)	90	NKOCHHAR	515.123.4568	21-SEP-89
102	Lex	De Haan	17000	(null)	90	LDEHAAN	515.123.4569	13-JAN-93
103	Alexander	Hunold	9000	(null)	60	AHUNOLD	590.423.4567	03-JAN-90
104	Bruce	Ernst	6000	(null)	60	BERNST	590.423.4568	21-MAY-91
107	Diana	Lorentz	4200	(null)	60	DLORENTZ	590.423.5567	07-FEB-99
124	Kevin	Mourgos	5800	(null)	50	KMOURGOS	650.123.5234	16-NOV-99
141	Trenna	Rajs	3500	(null)	50	TRAJS	650.121.8009	17-OCT-95
142	Curtis	Davies	3100	(null)	50	CDAVIES	650.121.2994	29-JAN-97
143	Randall	Matos	2600	(null)	50	RMATOS	650.121.2874	15-MAR-98
144	Peter	Vargas	2500	(null)	50	PVARGAS	650.121.2004	09-JUL-98
149	Eleni	Zlotkey	10500	0.2	80	EZLOTKEY	011.44.1344.429018	29-JAN-00
174	Ellen	Abel	11000	0.3	80	EABEL	011.44.1644.429267	11-MAY-96
176	Jonathon	Taylor	8600	0.2	80	JTAYLOR	011.44.1644.429265	24-MAR-98
178	Kimberely	Grant	7000	0.15	(null)	KGRANT	011.44.1644.429263	24-MAY-99
200	Jennifer	Whalen	4400	(null)	10	JWHALEN	515.123.4444	17-SEP-87
201	Michael	Hartstein	13000	(null)	20	MHARTSTE	515.123.5555	17-FEB-96
202	Pat	Fay	6000	(null)	20	PFAY	603.123.6666	17-AUG-97
205	Shelley	Higgins	12000	(null)	110	SHIGGINS	515.123.8080	07-JUN-94
206	William	Gietz	8300	(null)	110	WGIEZT	515.123.8181	07-JUN-94

↑  
Ограничение NOT NULL  
(ни одна строка не может  
содержать неопределенное  
значение в этом столбце)

↑  
Отсутствие ограничения  
NOT NULL (любая строка  
может содержать  
неопределенное значение в  
этом столбце)



# Ограничение UNIQUE

EMPLOYEES

Ограничение UNIQUE

	EMPLOYEE_ID	LAST_NAME	EMAIL
1	100	King	SKING
2	101	Kochhar	NKOCHHAR
3	102	De Haan	LDEHAAN
4	103	Hunold	AHUNOLD
5	104	Ernst	BERNST
6	107	Lorentz	DLORENTZ

...



INSERT INTO

208	SMITH	JSMITH
-----	-------	--------

Разрешается

209	SMITH	JSMITH
-----	-------	--------


Не разрешается  
(уже  
существует)





# Ограничение UNIQUE

- Может быть задано на уровне столбца или таблицы

```
CREATE TABLE employees (  
    employee_id      NUMBER(6),  
    last_name        VARCHAR2(25) NOT NULL,  
    email            VARCHAR2(25),  
    salary           NUMBER(8,2),  
    commission_pct   NUMBER(2,2),  
    hire_date        DATE NOT NULL,  
    ...  
    CONSTRAINT emp_email_uk UNIQUE(email));
```

# Ограничение PRIMARY KEY

DEPARTMENTS  ГЛАВНЫЙ КЛЮЧ

	 DEPARTMENT_ID	 DEPARTMENT_NAME	 MANAGER_ID	 LOCATION_ID
1	10	Administration	200	1700
2	20	Marketing	201	1800
3	50	Shipping	124	1500
4	60	IT	103	1400
5	80	Sales	149	2500
6	90	Executive	100	1700
7	110	Accounting	205	1700
8	190	Contracting	(null)	1700

Не разрешается  
(неопределенное  
значение)

 INSERT INTO





(null)	Public Accounting	124	2500
50	Finance	124	1500

Не разрешается  
(50 уже существует)

# Ограничение FOREIGN KEY




## DEPARTMENTS

ГЛАВНЫЙ КЛЮЧ

	 DEPARTMENT_ID	 DEPARTMENT_NAME	 MANAGER_ID	 LOCATION_ID
1	10	Administration	200	1700
2	20	Marketing	201	1800
3	50	Shipping	124	1500
4	60	IT	103	1400
5	80	Sales	149	2500

...

## EMPLOYEES

	 EMPLOYEE_ID	 LAST_NAME	 DEPARTMENT_ID
1	200	Whalen	10
2	201	Hartstein	20
3	202	Fay	20
4	205	Higgins	110
5	206	Gietz	110

ВНЕШНИЙ КЛЮЧ

...



INSERT INTO

200	Ford	9
200	Ford	60

Не разрешается  
(9 не существует)

Разрешается

# Ограничение FOREIGN KEY

- Может быть задано на уровне таблицы или столбца

```
CREATE TABLE employees(  
    employee_id      NUMBER(6),  
    last_name        VARCHAR2(25) NOT NULL,  
    email            VARCHAR2(25),  
    salary            NUMBER(8,2),  
    commission_pct   NUMBER(2,2),  
    hire_date        DATE NOT NULL,  
    ...  
    department_id    NUMBER(4),  
    CONSTRAINT emp_dept_fk FOREIGN KEY (department_id)  
        REFERENCES departments(department_id),  
    CONSTRAINT emp_email_uk UNIQUE(email));
```

# Ключевые слова ограничения FOREIGN KEY

- FOREIGN KEY

Определяет на уровне таблицы столбец в подчиненной таблице, используемый в качестве внешнего ключа

- REFERENCES

Определяет родительскую таблицу и столбец в ней

- ON DELETE CASCADE

Разрешает удаление в родительской таблице с одновременным удалением зависимых строк в подчиненной таблице

- ON DELETE SET NULL

При удалении строк в родительской таблице преобразует зависимое значение внешнего ключа в неопределенное

# Ограничение CHECK

- Задаёт условие, которому должна удовлетворять каждая строка
- Не допускаются:
  - ссылки на псевдостолбцы CURRVAL, NEXTVAL, LEVEL и ROWNUM ;
  - вызовы функций SYSDATE, UID, USER и USERENV ;
  - запросы, ссылающиеся на другие значения в других строках.

```
..., salary  NUMBER(2)  
    CONSTRAINT emp_salary_min  
        CHECK (salary > 0),...
```

# CREATE TABLE: пример

```
CREATE TABLE employees
( employee_id      NUMBER(6)
  CONSTRAINT emp_employee_id PRIMARY KEY
, first_name       VARCHAR2(20)
, last_name        VARCHAR2(25)
  CONSTRAINT emp_last_name_nn NOT NULL
, email            VARCHAR2(25)
  CONSTRAINT emp_email_nn    NOT NULL
  CONSTRAINT emp_email_uk    UNIQUE
, phone_number     VARCHAR2(20)
, hire_date        DATE
  CONSTRAINT emp_hire_date_nn NOT NULL
, job_id           VARCHAR2(10)
  CONSTRAINT emp_job_nn      NOT NULL
, salary           NUMBER(8,2)
  CONSTRAINT emp_salary_ck   CHECK (salary>0)
, commission_pct   NUMBER(2,2)
, manager_id       NUMBER(6)
  CONSTRAINT emp_manager_fk REFERENCES
    employees (employee_id)
, department_id    NUMBER(4)
  CONSTRAINT emp_dept_fk    REFERENCES
    departments (department_id));
```



# Нарушение ограничений

```
UPDATE employees  
SET    department_id = 55  
WHERE  department_id = 110;
```

Error starting at line 1 in command:

```
UPDATE employees  
SET    department_id = 55  
WHERE  department_id = 110
```

Error report:

```
SQL Error: ORA-02291: integrity constraint (ORA1.EMP_DEPT_FK) violated - parent key not found  
02291. 00000 - "integrity constraint (%s.%s) violated - parent key not found"  
*Cause:      A foreign key value has no matching primary key value.
```

- Отдел номер 55 не существует.

# Нарушение ограничений

- Нельзя удалить строку, содержащую первичный ключ, на который есть ссылка из внешнего ключа другой таблицы.

```
DELETE FROM departments  
WHERE department_id = 60;
```

Error starting at line 1 in command:

```
DELETE FROM departments
```

```
WHERE department_id = 60
```

Error report:

SQL Error: ORA-02292: integrity constraint (ORA1.JHIST\_DEPT\_FK) violated - child record found  
02292. 00000 - "integrity constraint (%s.%s) violated - child record found"

\*Cause: attempted to delete a parent key value that had a foreign  
dependency.

\*Action: delete dependencies first then parent or disable constraint.

# Создание таблицы с использованием подзапроса

- Создание таблицы и вставка строк путем совместного использования команды CREATE TABLE и опции AS *подзапрос*.

```
CREATE TABLE table  
            [ (столбец, столбец...) ]  
AS подзапрос;
```

- Количество заданных столбцов должно совпадать с количеством столбцов в подзапросе.
- Столбцы могут быть определены с именами и значениями, используемыми по умолчанию.

# Создание таблицы с использованием подзапроса

```
CREATE TABLE dept80
AS
SELECT  employee_id, last_name,
        salary*12 ANNSAL,
        hire_date
FROM    employees
WHERE   department id = 80;
```

```
CREATE TABLE succeeded.
```

```
DESCRIBE dept80
```

Name	Null	Type
EMPLOYEE_ID		NUMBER(6)
LAST_NAME	NOT NULL	VARCHAR2(25)
ANNSAL		NUMBER
HIRE_DATE	NOT NULL	DATE

# Команда ALTER TABLE

- Команда ALTER TABLE используется для следующих операций:
  - Добавление столбца
  - Изменение существующего столбца
  - Задание значения по умолчанию для нового столбца
  - Удаление столбца
  - Изменение статуса таблицы read-only

# Read-Only таблицы

- Можно использовать ALTER TABLE для:
  - Перевода таблицы в режим «только чтение», который запрещает выполнение команд DML или DDL по отношению к ней
  - Перевода таблицы в режим «чтение/запись»

```
ALTER TABLE employees READ ONLY;  
  
-- perform table maintenance and then  
-- return table back to read/write mode  
  
ALTER TABLE employees READ WRITE;
```

# Удаление таблицы

- Перемещение таблицы в корзину (recycle bin)
- Удаление таблицы и всех ее данных при использовании опции PURGE
- Перевод в состояние «invalid» зависимых объектов и удаление объектных привилегий на удаляемую таблицу

```
DROP TABLE dept80 succeeded.
```

```
DROP TABLE dept80;
```