10

Использование инструкций DDL для создания таблиц и управления ими

Цели

Изучив материал этого занятия, вы освоите следующие темы:

- Классификация основных объектов базы данных
- Просмотр структуры таблиц
- Доступные типы данных столбцов
- Создание простой таблицы
- Определение ограничений при создании таблиц
- Описание принципов работы объектов схемы

План занятия

- Объекты базы данных
  - правила присвоения имен
- Инструкция `CREATE TABLE`:
  - доступ к таблицам другого пользователя
  - параметр `DEFAULT`
- Типы данных
- Обзор ограничений: ограничения по наличию данных (`NOT NULL`), первичного ключа (`PRIMARY KEY`), внешнего ключа (`FOREIGN KEY`) и проверки (`CHECK`)
- Создание таблицы с помощью подзапроса
- Инструкция `ALTER TABLE`
  - таблицы только для чтения
- Инструкция `DROP TABLE`

Объекты базы данных

Объект	Описание
Таблица	Основная единица хранения, состоящая из строк
Представление	Логически представляет подмножества данных из одной или нескольких таблиц
Последователь	Создает численные значения
Индекс	Повышает эффективность некоторых запросов
Синоним	Определяет альтернативное имя объекта

Правила присвоения имен

Имена таблиц и столбцов должны подчиняться следующим правилам:

- начинаться с буквы
- иметь длину 1–30 символов
- содержать только символы A–Z, a–z, 0–9, _, $ и #
- не дублировать имя другого объекта, принадлежащего тому же пользователю
- не являться зарезервированным словом сервера Oracle

План занятия

- Объекты базы данных
  - правила присвоения имен
- Инструкция `CREATE TABLE`:
  - доступ к таблицам другого пользователя
  - параметр `DEFAULT`
- Типы данных
- Обзор ограничений: ограничения по наличию данных (`NOT NULL`), первичного ключа (`PRIMARY KEY`), внешнего ключа (`FOREIGN KEY`) и проверки (`CHECK`)
- Создание таблицы с помощью подзапроса
- Инструкция `ALTER TABLE`
  - таблицы только для чтения
- Инструкция `DROP TABLE`

Инструкция CREATE TABLE

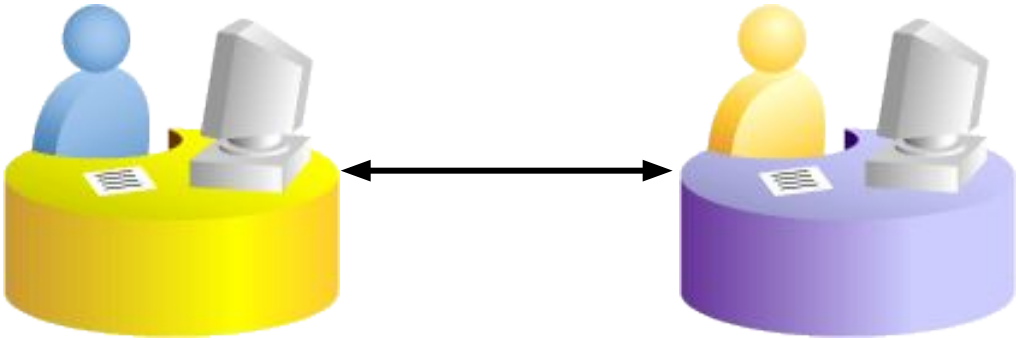
- Для этого необходимо:
  - привилегия на создание таблиц (CREATE TABLE)
  - область хранения

```
CREATE TABLE [schema.]table
          (column datatype [DEFAULT expr][, ...]);
```

- Требуется определить:
  - имя таблицы
  - имя, тип данных и размер столбца

Ссылка на таблицы другого пользователя

- Таблицы других пользователей не находятся в схеме данного пользователя.
- Необходимо указывать в качестве префикса в названиях этих таблиц имя владельца.



USERA

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

USERB

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

Параметр DEFAULT

- Укажите при вставке стандартное значение столбца.

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

- Допустимые значения: литералы, выражения или функции SQL.
- Использование имен других столбцов или псевдостолбцов не допускается.
- Стандартный тип данных должен совпадать с типом данных столбца.

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

```
CREATE TABLE succeeded.
```

Создание таблиц

- Создайте таблицу:

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

```
CREATE TABLE succeeded.
```

- Подтвердите создание таблицы:

```
DESCRIBE dept
```

```
DESCRIBE dept
Name                          Null     Type
----------------------------- -------- ------------
DEPTNO                                 NUMBER(2)
DNAME                                  VARCHAR2(14)
LOC                                    VARCHAR2(13)
CREATE_DATE                            DATE
```

План занятия

- Объекты базы данных
  - правила присвоения имен
- Инструкция `CREATE TABLE`:
  - доступ к таблицам другого пользователя
  - параметр `DEFAULT`
- Типы данных
- Обзор ограничений: ограничения по наличию данных (`NOT NULL`), первичного ключа (`PRIMARY KEY`), внешнего ключа (`FOREIGN KEY`) и проверки (`CHECK`)
- Создание таблицы с помощью подзапроса
- Инструкция `ALTER TABLE`
  - таблицы только для чтения
- Инструкция `DROP TABLE`

Типы данных

Тип данных	Описание
VARCHAR2(*size*)	Символьные данные переменной длины
CHAR(*size*)	Символьные данные фиксированной длины
NUMBER(*p*,*s*)	Численные данные переменной длины
DATE	Значения даты и времени
LONG	Символьные данные переменной длины (до 2 Гб)
CLOB	Символьные данные (до 4 Гб)
RAW и LONG RAW	Необработанные двоичные данные
BLOB	Двоичные данные (до 4 Гб)
BFILE	Двоичные данные, сохраненные во внешнем файле (до 4 Гб)
ROWID	Система номеров base-64, представляющая уникальный адрес строки в таблице

Типы данных даты/времени

Можно использовать несколько типов данных даты/времени:

Тип данных	Описание
TIMESTAMP	Дата с дробными секундами
INTERVAL YEAR TO MONTH	Хранится в виде интервала лет и месяцев
INTERVAL DAY TO SECOND	Хранится в виде интервала дней, часов, минут и секунд

План занятия

- Объекты базы данных
  - правила присвоения имен
- Инструкция `CREATE TABLE`:
  - доступ к таблицам другого пользователя
  - параметр `DEFAULT`
- Типы данных
- Обзор ограничений: ограничения по наличию данных (`NOT NULL`), первичного ключа (`PRIMARY KEY`), внешнего ключа (`FOREIGN KEY`) и проверки (`CHECK`)
- Создание таблицы с помощью подзапроса
- Инструкция `ALTER TABLE`
  - таблицы только для чтения
- Инструкция `DROP TABLE`

Включение ограничений

- Ограничения применяют правила на уровне таблицы.
- Ограничения предотвращают удаление таблицы, если для нее существуют зависимости.
- Допускаются следующие типы ограничений:
  - NOT NULL (не пусто)
  - UNIQUE (уникальность)
  - PRIMARY KEY (первичный ключ)
  - FOREIGN KEY (внешний ключ)
  - CHECK (проверка)

Указания по ограничениям

- Имя ограничения может быть определено пользователем или создано сервером Oracle с использованием формата `SYS_C`*n*.
- Время создания ограничений:
  - при создании таблицы
  - после создания таблицы
- Ограничение определяется на уровне столбца или таблицы.
- Просмотр ограничения возможен в словаре данных.

Определение ограничений

- Синтаксис:

```
CREATE TABLE [schema.]table
      (column datatype [DEFAULT expr]
      [column_constraint],
      ...
      [table_constraint][,...]);
```

- Синтаксис ограничения на уровне столбца:

```
column [CONSTRAINT constraint_name]
constraint_type,
```

- Синтаксис ограничения на уровне таблицы:

```
column,...
  [CONSTRAINT constraint_name] constraint_type
  (column, ...),
```

Определение ограничений

- Пример ограничения на уровне столбца:

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

1

- Пример ограничения 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 (не пусто)

Обеспечивает запрет пустых значений для столбца:

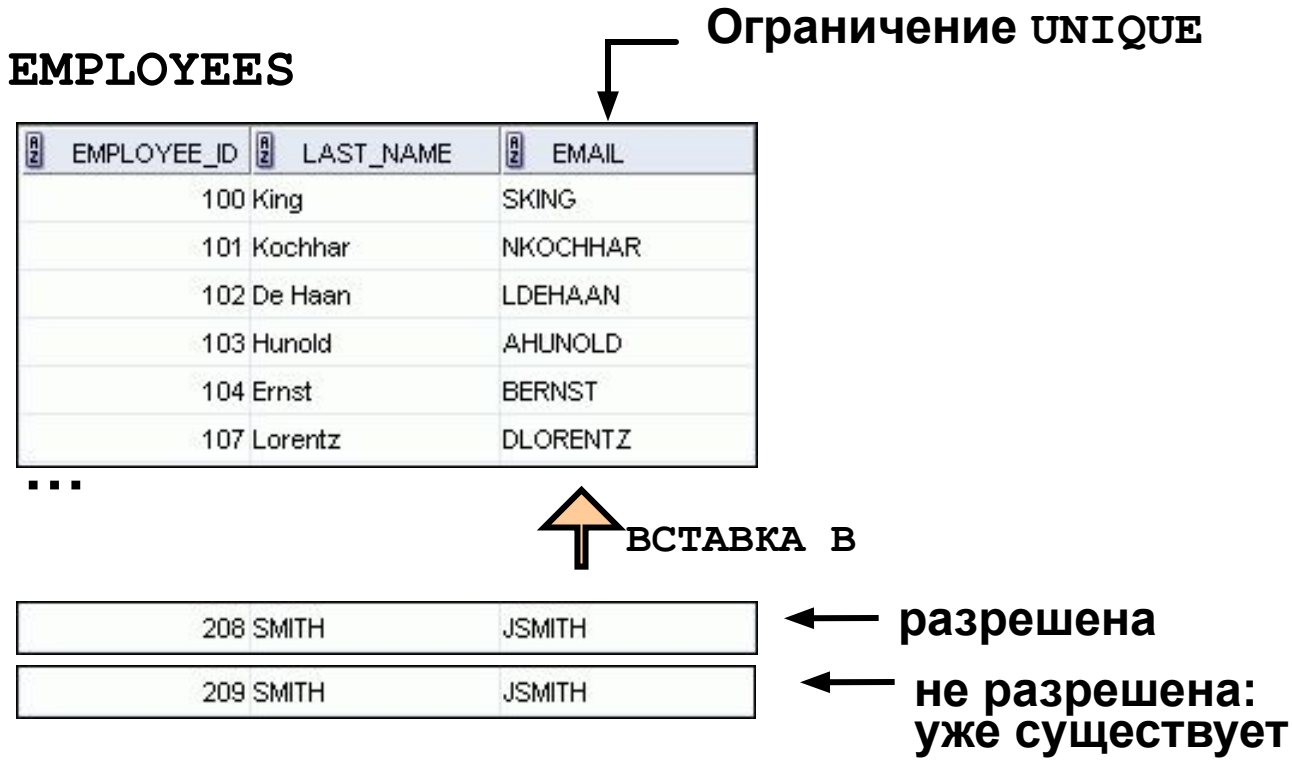
EMPLOYEE_ID	FIRST_NAME	LAST_NAME	EMAIL	HIRE_DATE	JOB_ID	COMMISSION_PCT
100	Steven	King	SKING	17-JUN-87	AD_PRES	(null)
101	Neena	Kochhar	NKOCHHAR	21-SEP-89	AD_VP	(null)
102	Lex	De Haan	LDEHAAN	13-JAN-93	AD_VP	(null)
103	Alexander	Hunold	AHUNOLD	03-JAN-90	IT_PROG	(null)
104	Bruce	Ernst	BERNST	21-MAY-91	IT_PROG	(null)
107	Diana	Lorentz	DLORENTZ	07-FEB-99	IT_PROG	(null)
124	Kevin	Mourgos	KMOURGOS	16-NOV-99	ST_MAN	(null)
141	Trenna	Rajs	TRAJS	17-OCT-95	ST_CLERK	(null)
142	Curtis	Davies	CDAVIES	29-JAN-97	ST_CLERK	(null)
143	Randall	Matos	RMATOS	15-MAR-98	ST_CLERK	(null)
144	Peter	Vargas	PVARGAS	09-JUL-98	ST_CLERK	(null)
149	Eleni	Zlotkey	EZLOTKEY	29-JAN-00	SA_MAN	0.2
174	Ellen	Abel	EABEL	11-MAY-96	SA_REP	0.3
...						

Ограничение NOT NULL (не пусто) (первичный ключ применяет ограничение NOT NULL (не пусто).)

ограничение NOT NULL (не пусто)

Отсутствие ограничения NOT NULL (не пусто) (любая строка в этом столбце может содержать пустое значение.)

Ограничение UNIQUE



Ограничение UNIQUE

Определяется на уровне таблицы или столбца:

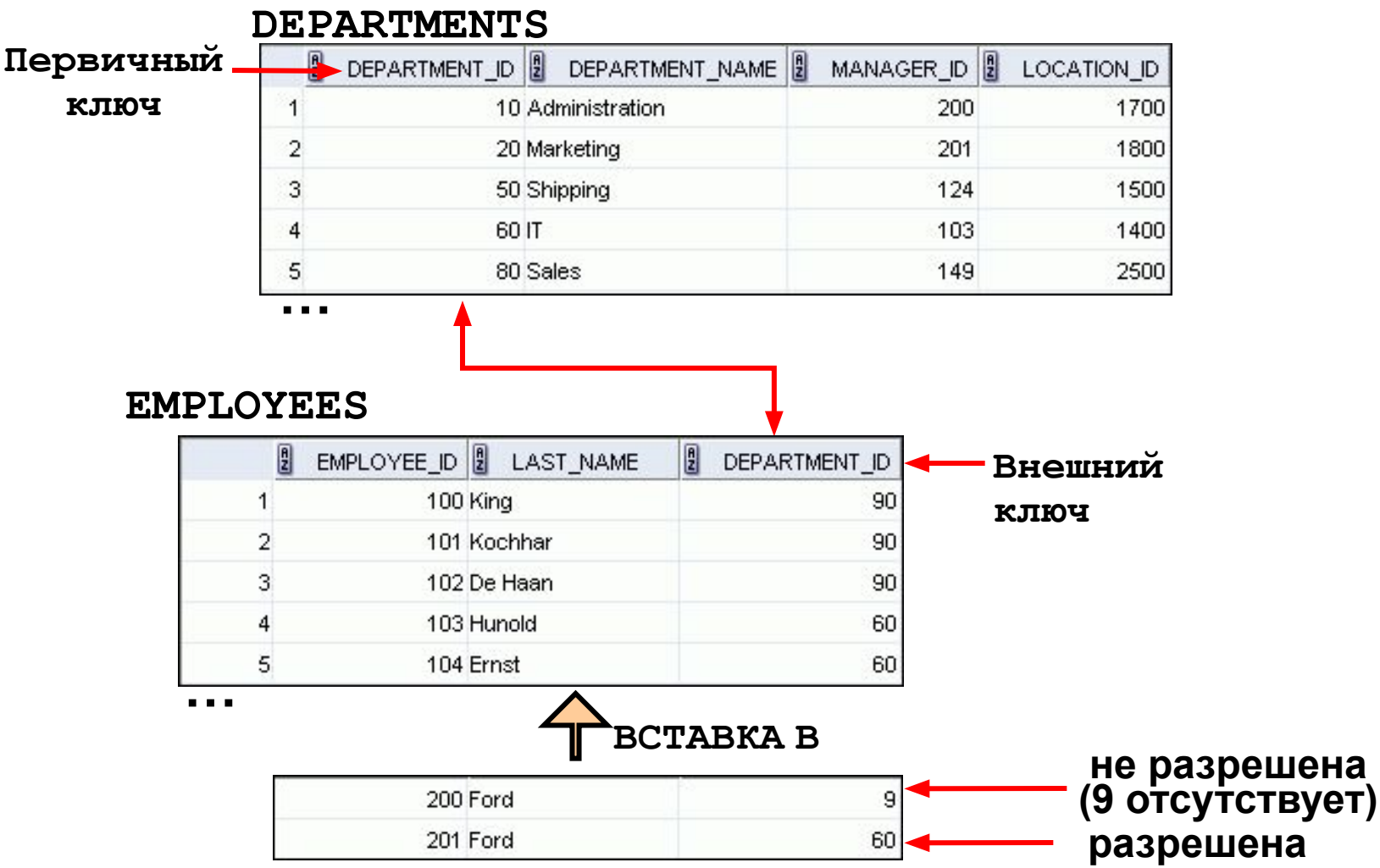
```
CREATE TABLE employees(
    employee_id      NUMBER(6),
    last_name        VARCHAR2(25)
NOT NULL,
    email            VARCHAR2(25),
    salary           NUMBER(8,2),
    commission_pct   NUMBER(2,2),
    hire_date        DATE NOT NULL,
...
    CONSTRAINT emp_email_uk
UNIQUE(email));
```

Ограничение PRIMARY KEY (первичный 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

	Public Accounting	124	2500
50	Finance	124	1500

Ограничение FOREIGN KEY (внешний ключ)



Ограничение FOREIGN KEY (внешний ключ)

Определяется 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));
```

Ограничение FOREIGN KEY: ключевые слова

- FOREIGN KEY: определяет столбец в дочерней таблице на уровне ограничения таблицы
- REFERENCES: определяет таблицу и столбец в родительской таблице
- ON DELETE CASCADE: удаляет зависимые строки в дочерней таблице при удалении строки в родительской таблице
- ON DELETE SET NULL: преобразует зависимые значения внешнего ключа в пустые

Ограничение CHECK

- Определяет условие, которому должна соответствовать каждая строка
- Следующие значения запрещены:
  - 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),...
```

Пример инструкции CREATE TABLE

```
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 (ORA16.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.
*Action:   Delete the foreign key or add a matching primary key.
```

Отдел 55 не существует.

Нарушение ограничений

Нельзя удалить строку, которая содержит первичный ключ, используемый в качестве внешнего ключа в другой таб

```
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 (ORA16.EMP_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.
```

План занятия

- Объекты базы данных
  - правила присвоения имен
- Инструкция `CREATE TABLE`:
  - доступ к таблицам другого пользователя
  - параметр `DEFAULT`
- Типы данных
- Обзор ограничений: ограничения по наличию данных (`NOT NULL`), первичного ключа (`PRIMARY KEY`), внешнего ключа (`FOREIGN KEY`) и проверки (`CHECK`)
- Создание таблицы с помощью подзапроса
- Инструкция `ALTER TABLE`
  - таблицы только для чтения
- Инструкция `DROP TABLE`

Создание таблицы с помощью подзапроса

- Создайте таблицу, объединив инструкцию `CREATE TABLE` и параметр `AS` *`subquery`*.

```
CREATE TABLE table
      [(column, column...)]
AS subquery;
```

- Согласуйте число указанных столбцов с числом столбцов в подзапросе.
- Определите имена столбцов и стандартные значения.

Создание таблицы с помощью 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
```

План занятия

- Объекты базы данных
  - правила присвоения имен
- Инструкция `CREATE TABLE`:
  - доступ к таблицам другого пользователя
  - параметр `DEFAULT`
- Типы данных
- Обзор ограничений: ограничения по наличию данных (`NOT NULL`), первичного ключа (`PRIMARY KEY`), внешнего ключа (`FOREIGN KEY`) и проверки (`CHECK`)
- Создание таблицы с помощью подзапроса
- Инструкция `ALTER TABLE`
  - таблицы только для чтения
- Инструкция `DROP TABLE`

Инструкция ALTER TABLE

Инструкция `ALTER TABLE` позволяет выполнять следующие действия:

- Добавление нового столбца
- Изменение существующего определения столбца
- Определение стандартного значения для нового столбца
- Удаление столбца
- Переименование столбца
- Присвоение столбцу статуса «только чтение»

Таблицы только для чтения

Синтаксис `ALTER TABLE` позволяет перевести таблицу в режим «только чтение»:

- Запрещает изменения DDL или DML при обслуживании таблицы
- Снова переключает таблицу в режим «чтение/запись»

```
ALTER TABLE employees READ ONLY;

-- выполняет обслуживание таблицы и затем
-- возвращает таблицу в режим «чтение/запись»

ALTER TABLE employees READ WRITE;
```

План занятия

- Объекты базы данных
  - правила присвоения имен
- Инструкция `CREATE TABLE`:
  - доступ к таблицам другого пользователя
  - параметр `DEFAULT`
- Типы данных
- Обзор ограничений: ограничения по наличию данных (`NOT NULL`), первичного ключа (`PRIMARY KEY`), внешнего ключа (`FOREIGN KEY`) и проверки (`CHECK`)
- Создание таблицы с помощью подзапроса
- Инструкция `ALTER TABLE`
  - таблицы только для чтения
- Инструкция `DROP TABLE`

Удаление таблицы

- Перемещение таблицы в корзину
- Полное удаление таблицы и всех ее данных, если указано предложение `PURGE`
- Перевод всех зависимых объектов в статус недействительных и удаление полномочий объектов для таблицы

```
DROP TABLE dept80;
```

```
DROP TABLE dept80 succeeded.
```

Заключение

На этом занятии вы изучили использование инструкции `CREATE TABLE` для создания таблицы и включения ограничений:

- Классификация основных объектов базы данных
- Просмотр структуры таблиц.
- Доступные типы данных столбцов
- Создание простой таблицы
- Определение ограничений при создании таблиц
- Описание принципов работы объектов схемы

Упражнение 10: обзор

Это упражнение охватывает следующие темы:

- Создание новых таблиц
- Создание новой таблицы с помощью синтаксиса `CREATE TABLE AS`
- Проверка существования таблицы
- Присвоение таблице статуса только для чтения
- Удаление таблиц