

Практика 6-2

Задачи

1. Откройте Oracle Application Express и выполните вход
2. Нажмите на значок **Help** и ознакомьтесь со следующими разделами и темами:
 - 1) Application Express SQL Workshop
 - a) Управление объектами базы данных в обозревателе объектов
 - b) Использование команд SQL
 - c) Использование сценариев SQL

Oracle Application Express SQL Workshop Guide

- ⊕ Preface
- ⊕ Changes in This Release
- ⊕ Getting Started with SQL Workshop
- ⊕ Managing Database Objects with Object Browser
- ⊕ Using SQL Commands
- ⊕ Using SQL Scripts
- ⊕ Using Oracle Application Express Utilities
- ⊕ Using RESTful Services

Практика 6-3

Задачи

1. Создайте инструкции DDL для формирования таблиц указанной выше академической базы данных. Там, где это необходимо, включите ограничения NOT NULL. (Остальные ограничения будут добавлены позже)
2. Запустите/выполните эти команды в **Oracle Application Express**.

```
CREATE TABLE academic_sessions (
    id VARCHAR2(10) NOT NULL,
    name VARCHAR2(30) NOT NULL
);

CREATE TABLE courses (
    id VARCHAR2(10) NOT NULL,
    name VARCHAR2(30) NOT NULL
);

CREATE TABLE departments_for_ab (
    id VARCHAR2(10) NOT NULL,
```

```
name VARCHAR2(30) NOT NULL,
head VARCHAR2(30) NOT NULL
);

CREATE TABLE exams (
    id VARCHAR2(10) NOT NULL,
    str_date DATE
);

CREATE TABLE faculties (
    id      VARCHAR2(10) NOT NULL,
    frst_name  VARCHAR2(30) NOT NULL,
    lst_name   VARCHAR2(30) NOT NULL,
    email     VARCHAR2(30) NOT NULL,
    sly      VARCHAR2(30),
    isu      VARCHAR2(30),

    hourly_rat VARCHAR2(30)
);

CREATE TABLE faculty_course_details (
    cnt_hours VARCHAR2(30) NOT NULL
);

CREATE TABLE faculty_login_details (
    login_dte_tme DATE NOT NULL
);

CREATE TABLE onlines (
    lgn_id VARCHAR2(30) NOT NULL,
    pwd   VARCHAR2(30) NOT NULL
);

CREATE TABLE pnt_informations (
    id  VARCHAR2(10) NOT NULL,
    pnt_1_frst_name VARCHAR2(30) NOT NULL,
    pnt_1_lst_name  VARCHAR2(30) NOT NULL,
    pnt_2_frst_name VARCHAR2(30) NOT NULL,
    pnt_2_lst_name  VARCHAR2(30) NOT NULL
);

CREATE TABLE seateds (
    bldn      VARCHAR2(30) NOT NULL,
    room      VARCHAR2(30) NOT NULL,
    date_time DATE NOT NULL
);

CREATE TABLE student_attendances (
    number_of_wrk_days  NUMBER(10) NOT NULL,
    number_of_days_off  NUMBER(10) NOT NULL,
    eligibility_for_xm VARCHAR2(30)
);

CREATE TABLE student_course_details (
```

```

grade NUMBER(10) NOT NULL
);

CREATE TABLE students (
    id VARCHAR2(10) NOT NULL,
    frst_name VARCHAR2(30) NOT NULL,
    lst_name VARCHAR2(30) NOT NULL,
    rgs_yea DATE NOT NULL,
    email VARCHAR2(30) NOT NULL
);

CREATE TABLE xam_results (
    grade NUMBER(10) NOT NULL
);

CREATE TABLE xam_types (
    typ VARCHAR2(30) NOT NULL,
    name VARCHAR2(30) NOT NULL,
    dsc VARCHAR2(30)
);

```

Все таблицы были созданы, но скрин, к сожалению не получилось сделать. Если же сделать повторный запуск, будет написано, что все таблицы уже созданы

Number ↑	Elapsed	Statement	Feedback	Rows
1	0.01	CREATE TABLE academic_sessions (id VARCHAR2(10) NOT N	ORA-00955: name is already used by an existing object	-
2	0.01	CREATE TABLE courses (id VARCHAR2(10) NOT NULL, na	ORA-00955: name is already used by an existing object	-
3	0.02	CREATE TABLE departments_for_ab (id VARCHAR2(10) NOT	Table created.	0
4	0.00	CREATE TABLE exams (id VARCHAR2(10) NOT NULL, str_	ORA-00955: name is already used by an existing object	-
5	0.01	CREATE TABLE faculties (id VARCHAR2(10) NOT NU	ORA-00955: name is already used by an existing object	-
6	0.00	CREATE TABLE faculty_course_details (cnt_hours VARCHAR2	ORA-00955: name is already used by an existing object	-
7	0.01	CREATE TABLE faculty_login_details (login_dte_tmie DATE	ORA-00955: name is already used by an existing object	-
8	0.00	CREATE TABLE onlines (lgn_id VARCHAR2(30) NOT NULL,	ORA-00955: name is already used by an existing object	-
9	0.01	CREATE TABLE pnt_informations (id VARCHAR2(10) NOT	ORA-00955: name is already used by an existing object	-
10	0.00	CREATE TABLE seateds (bldn VARCHAR2(30) NOT NUL	ORA-00955: name is already used by an existing object	-
11	0.01	CREATE TABLE student_attendances (number_of_wrk_days	ORA-00955: name is already used by an existing object	-
12	0.01	CREATE TABLE student_course_details (grade NUMBER(10)	ORA-00955: name is already used by an existing object	-
13	0.01	CREATE TABLE students (id VARCHAR2(10) NOT NULL,	ORA-00955: name is already used by an existing object	-
14	0.00	CREATE TABLE xam_results (grade NUMBER(10) NOT NULL)	ORA-00955: name is already used by an existing object	-
15	0.01	CREATE TABLE xam_types (typ VARCHAR2(30) NOT NULL,	ORA-00955: name is already used by an existing object	-

Так же случился конфликт, так как таблица department уже была создана. Пришлось таблицу в этом проекте переименовать...

Упражнение 2

Предположения

Ограничения первичного и внешнего ключа основываются на ERD-диаграмме из предыдущего упражнения, а в основе уникальных ограничений лежит следующее:

Перечисленные ниже поля должны иметь уникальные значения:

- Course Name в AD_COURSES
- Department Name в AD_DEPARTMENTS
- Student Email в AD_STUDENTS
- Faculty Email в AD_FACULTY
- Session Name в AD_ACADEMIC_SESSIONS

Задачи

1. Измените таблицы в академической базе данных, чтобы определить первичный ключ, внешний ключ и уникальные ограничения.
2. Измените таблицу AD_FACULTY_LOGIN_DETAILS и укажите значение по умолчанию для столбца LOGIN_DATE_TIME в SYSDATE.
3. Присвойте таблице AD_PARENT_INFORMATION статус «только для чтения».

ПРИМЕЧАНИЕ. Инструкции INSERT / ALTER TABLE можно выполнить в Oracle Application Express одним из следующих способов.

Метод 1.

- a. Откройте Oracle Application Express, поочередно вставьте команды на экране SQL Commands и выполните эти команды.

Метод 2.

- a. Откройте Oracle Application Express и примените метод загрузки сценария, который ранее использовался с командами DDL.

```
ALTER TABLE courses ADD(  
    dpt_id          VARCHAR2(10) NOT NULL,  
    onlines_lgn_id   VARCHAR2(10) ,  
    std_bld         VARCHAR2(10) ,  
    std_room        VARCHAR2(10) ,  
    date_time       DATE ,  
    academic_sessions_id VARCHAR2(10) NOT NULL);
```

```
ALTER TABLE courses  
ADD CONSTRAINT arc_1 CHECK ( ( ( onlines_lgn_id IS NOT NULL )  
                                AND ( std_bld IS NULL )  
                                AND ( std_room IS NULL )  
                                AND ( date_time IS NULL ) )  
                            OR ( ( std_bld IS NOT NULL )  
                                AND ( std_room IS NOT NULL )  
                                AND ( date_time IS NOT NULL )  
                                AND ( onlines_lgn_id IS NULL ) ) );
```

```
ALTER TABLE exams ADD xam_types_typ VARCHAR2(10);  
ALTER TABLE faculties ADD dpt_id VARCHAR2(10);  
ALTER TABLE faculty_course_details ADD  
    (crs_id VARCHAR2(10) NOT NULL,  
     fclt_id VARCHAR2(10) NOT NULL);  
ALTER TABLE faculty_login_details ADD fclt_id VARCHAR2(10) NOT NULL;  
ALTER TABLE student_attendances ADD  
    (stu_id VARCHAR2(10) NOT NULL,  
     academic_sessions_id VARCHAR2(10) NOT NULL);  
ALTER TABLE student_course_details ADD  
    (stu_id VARCHAR2(10) NOT NULL,  
     crs_id VARCHAR2(10) NOT NULL);
```

```

ALTER TABLE students ADD pnt_inf_id VARCHAR2(10);
ALTER TABLE xam_results ADD
    (stu_id VARCHAR2(10) NOT NULL,
    crs_id VARCHAR2(10) NOT NULL,
    xam_id VARCHAR2(10) NOT NULL);

ALTER TABLE courses ADD CONSTRAINT unique_name_crs UNIQUE (name);
ALTER TABLE departments_for_ab ADD CONSTRAINT unique_name_dprt UNIQUE (name);
ALTER TABLE students ADD CONSTRAINT unique_email_std UNIQUE (email);
ALTER TABLE faculties ADD CONSTRAINT unique_email_fclt UNIQUE (email);
ALTER TABLE academic_sessions ADD CONSTRAINT unique_name_session UNIQUE (name);

ALTER TABLE academic_sessions ADD CONSTRAINT academic_sessions_pk PRIMARY KEY ( id );
ALTER TABLE courses ADD CONSTRAINT crs_pk PRIMARY KEY ( id );
ALTER TABLE exams ADD CONSTRAINT xam_pk PRIMARY KEY ( id );
ALTER TABLE faculties ADD CONSTRAINT fclt_pk PRIMARY KEY ( id );
ALTER TABLE faculty_course_details ADD CONSTRAINT fclt_crs_details_pk PRIMARY KEY ( crs_id,
    fclt_id );
ALTER TABLE faculty_login_details ADD CONSTRAINT fclt_login_details_pk PRIMARY KEY ( fclt_id );
ALTER TABLE onlines ADD CONSTRAINT onlines_pk PRIMARY KEY ( lgn_id );
ALTER TABLE seateds
    ADD CONSTRAINT std_pk PRIMARY KEY ( bldn,
        room,
        date_time);
ALTER TABLE student_attendances ADD CONSTRAINT stu_attendances_pk PRIMARY KEY ( stu_id );
ALTER TABLE student_course_details ADD CONSTRAINT stu_crs_details_pk PRIMARY KEY ( stu_id,
    crs_id );
ALTER TABLE students ADD CONSTRAINT stu_pk PRIMARY KEY ( id );
ALTER TABLE xam_results
    ADD CONSTRAINT xam_results_pk PRIMARY KEY ( crs_id,
        stu_id,
        xam_id );
ALTER TABLE xam_types ADD CONSTRAINT xam_types_pk PRIMARY KEY ( typ );

ALTER TABLE courses
    ADD CONSTRAINT crs_academic_sessions_fk FOREIGN KEY ( academic_sessions_id )
        REFERENCES academic_sessions ( id );

ALTER TABLE courses
    ADD CONSTRAINT crs_onlines_fk FOREIGN KEY ( onlines_lgn_id )
        REFERENCES onlines ( lgn_id );

ALTER TABLE courses
    ADD CONSTRAINT crs_std_fk FOREIGN KEY ( std_bld,
        std_room,
        date_time )
        REFERENCES seateds ( bldn,
            room,
            date_time );

```

```
        date_time );  
  
ALTER TABLE faculty_course_details  
    ADD CONSTRAINT fclt_crs_details_crs_fk FOREIGN KEY ( crs_id )  
        REFERENCES courses ( id );  
  
ALTER TABLE faculty_course_details  
    ADD CONSTRAINT fclt_crs_details_fclt_fk FOREIGN KEY ( fclt_id )  
        REFERENCES faculties ( id );  
  
ALTER TABLE faculty_login_details  
    ADD CONSTRAINT fclt_login_details_fclt_fk FOREIGN KEY ( fclt_id )  
        REFERENCES faculties ( id );  
  
ALTER TABLE student_attendances  
    ADD CONSTRAINT stu_attend_acadm_sessions_fk FOREIGN KEY ( academic_sessions_id )  
        REFERENCES academic_sessions ( id );  
  
ALTER TABLE student_attendances  
    ADD CONSTRAINT stu_attendances_stu_fk FOREIGN KEY ( stu_id )  
        REFERENCES students ( id );  
  
ALTER TABLE student_course_details  
    ADD CONSTRAINT stu_crs_details_crs_fk FOREIGN KEY ( crs_id )  
        REFERENCES courses ( id );  
  
ALTER TABLE student_course_details  
    ADD CONSTRAINT stu_crs_details_stu_fk FOREIGN KEY ( stu_id )  
        REFERENCES students ( id );  
  
ALTER TABLE xam_results  
    ADD CONSTRAINT xam_results_crs_fk FOREIGN KEY ( crs_id )  
        REFERENCES courses ( id );  
  
ALTER TABLE xam_results  
    ADD CONSTRAINT xam_results_stu_fk FOREIGN KEY ( stu_id )  
        REFERENCES students ( id );  
  
ALTER TABLE xam_results  
    ADD CONSTRAINT xam_results_xam_fk FOREIGN KEY ( xam_id )  
        REFERENCES exams ( id );  
  
ALTER TABLE exams  
    ADD CONSTRAINT xam_xam_types_fk FOREIGN KEY ( xam_types_typ )  
        REFERENCES xam_types ( typ );  
  
ALTER TABLE PNT_INFORMATIONS READ ONLY;  
  
ALTER TABLE FACULTY_LOGIN_DETAILS MODIFY LOGIN_DTE_TME SYSDATE;
```

Все ограничения созданы.

Упражнение 3. Создание составных первичных, внешних и уникальных ключей

Обзор

Цель этого упражнения — создать следующие элементы:

- Составной первичный ключ
- Составной внешний ключ
- Составной уникальный ключ

**** Примечание. Эти таблицы не входят в академическую базу данных**

1. Создайте таблицу DEPT с указанной структурой:

Column	Data Type	Description
dept_id	number(8)	Department ID
dept_name	varchar2(30)	Department Name
loc_id	number(4)	Location ID

Первичный ключ для этой таблицы должен представлять собой составной ключ, включающий в себя dept_id и loc_id.

```
CREATE TABLE DEPT(  
dept_id NUMBER(8),  
dept_name VARCHAR2(30),  
loc_id NUMBER(4),  
CONSTRAINT dept_pk PRIMARY KEY (dept_id, loc_id)  
);
```

Column Name	Data Type	Nullable	Default	Primary Key
DEPT_ID	NUMBER(8,0)	No	-	1
LOC_ID	NUMBER(4,0)	No	-	2
DEPT_NAME	VARCHAR2(30)	Yes	-	-

[Download](#)

2. Создайте таблицы SUPPLIERS и PRODUCTS с указанной структурой:

ТАБЛИЦА SUPPLIERS

Column	Data Type	Description
sup_id	number(15)	Supplier ID — часть составного первичного ключа
sup_name	varchar2(30)	Supplier Name — часть составного первичного ключа
contact_name	number(4)	Контактное имя агента

Первичный ключ для этой таблицы состоит из sup_id и sup_name.

ТАБЛИЦА PRODUCTS

Column	Data Type	Description
product_id	number(10)	Product ID — первичный ключ
sup_id	number(15)	Supplier ID, не содержащий значение NULL
sup_name	varchar2(30)	Supplier Name, не содержащий значение NULL

Первичный ключ для этой таблицы — product_id. Внешний ключ для этой таблицы состоит из sup_id и sup_name.

```

CREATE TABLE SUPPLIERS(
sup_id NUMBER(15),
sup_name VARCHAR2(30),
contact_name NUMBER(4),
CONSTRAINT suppl_pk PRIMARY KEY (sup_id , sup_name)
);

```

Column Name	Data Type	Nullable	Default	Primary Key
SUP_ID	NUMBER(15,0)	No	-	1
SUP_NAME	VARCHAR2(30)	No	-	2
CONTACT_NAME	NUMBER(4,0)	Yes	-	-

```

CREATE TABLE PRODUCTS(
product_id NUMBER(10),
sup_id NUMBER(15) NOT NULL,
sup_name VARCHAR2(30) NOT NULL,
CONSTRAINT prod_pk PRIMARY KEY (product_id ),
CONSTRAINT prod_sup_fk FOREIGN KEY ( sup_id, sup_name )
    REFERENCES SUPPLIERS(sup_id, sup_name)
);

```

Column Name	Data Type	Nullable	Default	Primary Key
PRODUCT_ID	NUMBER(10,0)	No	-	1
SUP_ID	NUMBER(15,0)	No	-	-
SUP_NAME	VARCHAR2(30)	No	-	-

SYS_C0014521098	Check	"SUP_ID" IS NOT NULL	-	-
SYS_C0014521099	Check	"SUP_NAME" IS NOT NULL	-	-
PROD_PK	Primary	-	-	PRODUCT_ID
PROD_SUP_FK	Foreign	-	PROD_SUP_FK (RU_A404_SQL_S04.SUPPLIERS)	SUP_ID, SUP_NAME

3. Создайте таблицу DEPT_SAMPLE с указанной структурой:

Column	Data Type	Description
dept_id	number(8)	Department ID
dept_name	varchar2(30)	Department Name
loc_id	number(4)	Location ID

УНИКАЛЬНЫЙ ключ для этой таблицы включает в себя dept_id и dept_name.

```
CREATE TABLE DEPT_SAMPLE(
dept_id NUMBER(8),
dept_name VARCHAR2(30),
loc_id NUMBER(4),
CONSTRAINT dept_sample_uk UNIQUE(dept_id, dept_name)
);
```

Проект 6-3

Задание 1

Часть 1. Чтение информации из сценария Цель этого упражнения — объединить знания о языке DDL на примере файла «obl Sports.ddl». Откройте файл «obl Sports.ddl» в текстовом редакторе.

1. Сколько таблиц создано с помощью инструкции CREATE TABLE?

10

2. Сколько столбцов создано для таблицы истории цен?

6

3. Какая инструкция используется для применения ограничения, указывающего, что столбец категории в таблице элементов должен иметь значение?

NOT NULL

4. Каково имя ограничения внешнего ключа между таблицами клиентов и адресов клиентов?

ALTER TABLE address ADD CONSTRAINT add_ctm_fk FOREIGN KEY (ctm_store_card) REFERENCES customer (store_card);

customer_address_customer_fk

5. Какое минимальное и максимальное значение могут храниться в столбце размера комиссии для таблицы продавцов-консультантов?

-99 to 99

6. Какое минимальное и максимальное значение могут храниться в столбце цен для таблицы истории цен?

-99999.99 to 99999.99

7. Какие 3 столбца образуют первичный ключ для таблицы истории цен?

itm_number, start_date, start_time.

Задание 2

Изменение столбца

1. Выполните команду DESCRIBE для таблицы заказов, чтобы просмотреть ее структуру.
2. **Задача.** Добавьте ограничение по умолчанию, указывающее, что требуется использовать сегодняшнюю дату для присвоения значения в столбце odr_date таблицы заказов, если дата не указана.
3. Выполните команду DESCRIBE еще раз, чтобы проверить, что она выполняется успешно.

1.

Table	Column	Data Type	Length	Precision	Scale	Primary Key
ORDERS	ID	VARCHAR2	9	-	-	1
	ODR_DATE	DATE	7	-	-	-
	ODR_TIME	DATE	7	-	-	-
	NUMBER_OF_UNITS	NUMBER	-	2	0	-
	CTR_NUMBER	VARCHAR2	6	-	-	-

2.

ALTER TABLE orders MODIFY odr_date DATE DEFAULT SYSDATE;

3.

Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default	Comm
ORDERS	ID	VARCHAR2	9	-	-	1	-	-	-
	ODR_DATE	DATE	7	-	-	-	-	SYSDATE	-
	ODR_TIME	DATE	7	-	-	-	-	-	-
	NUMBER_OF_UNITS	NUMBER	-	2	0	-	-	-	-
	CTR_NUMBER	VARCHAR2	6	-	-	-	-	-	-

Добавление ограничения проверки значения

1. Выполните команду DESCRIBE для таблицы клиентов, чтобы просмотреть ее структуру.
2. **Задача.** Добавьте ограничение проверки значения, которое не позволит текущему балансу клиентов становиться ниже нуля.
3. Выполните команду DESCRIBE еще раз, чтобы проверить, что она выполняется успешно.
4. Ограничение проверки значения не показано в результатах команды describe.
 - a. Перейдите в обозреватель объектов.
 - b. Выберите таблицу клиентов.
 - c. Откройте вкладку CONSTRAINTS.
 - d. Там отобразится ограничение.

1.

Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default	Comment
CUSTOMERS	CTR_NUMBER	VARCHAR2	6	-	-	1	-	-	-
	EMAIL	VARCHAR2	50	-	-	-	-	-	-
	FIRST_NAME	VARCHAR2	20	-	-	-	-	-	-
	LAST_NAME	VARCHAR2	30	-	-	-	-	-	-
	PHONE_NUMBER	VARCHAR2	11	-	-	-	-	-	-
	CURRENT_BALANCE	NUMBER	-	6	2	-	-	-	-
	SRE_ID	VARCHAR2	4	-	-	-	✓	-	-
	TEM_ID	VARCHAR2	4	-	-	-	✓	-	-
	LOYALTY_CARD_NUMBER	VARCHAR2	6	-	-	-	✓	-	-

2.

ALTER TABLE customers ADD CONSTRAINT cstm_cbl_min CHECK (current_balance > 0);

3.

Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default	Comment
CUSTOMERS	CTR_NUMBER	VARCHAR2	6	-	-	1	-	-	-
	EMAIL	VARCHAR2	50	-	-	-	-	-	-
	FIRST_NAME	VARCHAR2	20	-	-	-	-	-	-
	LAST_NAME	VARCHAR2	30	-	-	-	-	-	-
	PHONE_NUMBER	VARCHAR2	11	-	-	-	-	-	-
	CURRENT_BALANCE	NUMBER	-	6	2	-	-	-	-
	SRE_ID	VARCHAR2	4	-	-	-	✓	-	-
	TEM_ID	VARCHAR2	4	-	-	-	✓	-	-
	LOYALTY_CARD_NUMBER	VARCHAR2	6	-	-	-	✓	-	-

4.

Constraint	Type	Search Condition	Related Constraint			Columns	Delete Rule	Status	Last Change
CSTM_CBL_MIN	Check	current_balance > 0	-			-	-	ENABLED	04/08/2020 01:41:33 AM
SYS_C0014367907	Check	"CTR_NUMBER" IS NOT NULL	-			-	-	ENABLED	03/23/2020 07:20:23 AM

Добавление столбца

Клиент решил, что их компании требуется отдельный столбец для номера мобильного телефона клиентов. В этом дополнительном столбце требуется хранить 11 цифр.

- Выполните команду DESCRIBE для таблицы клиентов, чтобы просмотреть ее структуру.
- Задача.** Добавьте столбец, который удовлетворит требованиям клиента.
- Выполните команду DESCRIBE для таблицы клиентов, чтобы просмотреть ее структуру.

1.

Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default	Comment
CUSTOMERS	CTR_NUMBER	VARCHAR2	6	-	-	1	-	-	-
	EMAIL	VARCHAR2	50	-	-	-	-	-	-
	FIRST_NAME	VARCHAR2	20	-	-	-	-	-	-
	LAST_NAME	VARCHAR2	30	-	-	-	-	-	-
	PHONE_NUMBER	VARCHAR2	11	-	-	-	-	-	-
	CURRENT_BALANCE	NUMBER	-	6	2	-	-	-	-
	SRE_ID	VARCHAR2	4	-	-	-	✓	-	-
	TEM_ID	VARCHAR2	4	-	-	-	✓	-	-
	LOYALTY_CARD_NUMBER	VARCHAR2	6	-	-	-	✓	-	-

2.

ALTER TABLE customers ADD mobile_number NUMBER(11);

3.

Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default	Comment
CUSTOMERS	CTR_NUMBER	VARCHAR2	6	-	-	1	-	-	-
	EMAIL	VARCHAR2	50	-	-	-	-	-	-
	FIRST_NAME	VARCHAR2	20	-	-	-	-	-	-
	LAST_NAME	VARCHAR2	30	-	-	-	-	-	-
	PHONE_NUMBER	VARCHAR2	11	-	-	-	-	-	-
	CURRENT_BALANCE	NUMBER	-	6	2	-	-	-	-
	SRE_ID	VARCHAR2	4	-	-	-	✓	-	-
	TEM_ID	VARCHAR2	4	-	-	-	✓	-	-
	LOYALTY_CARD_NUMBER	VARCHAR2	6	-	-	-	✓	-	-
	MOBILE_NUMBER	NUMBER	-	11	0	-	✓	-	-

Удаление столбца

Клиент решил, что их компании не требуется отдельный столбец для номера мобильного телефона, потому что большинство клиентов указывают лишь один номер контактного лица, который уже обслуживается имеющимся столбцом phone_number.

1. Выполните команду DESCRIBE для таблицы клиентов, чтобы просмотреть ее структуру.
2. **Задача.** Удалите столбец, который создан для хранения номера мобильного телефона.
3. Выполните команду DESCRIBE для таблицы клиентов, чтобы просмотреть ее структуру.

1.

Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default	Comment
CUSTOMERS	CTR_NUMBER	VARCHAR2	6	-	-	1	-	-	-
	EMAIL	VARCHAR2	50	-	-	-	-	-	-
	FIRST_NAME	VARCHAR2	20	-	-	-	-	-	-
	LAST_NAME	VARCHAR2	30	-	-	-	-	-	-
	PHONE_NUMBER	VARCHAR2	11	-	-	-	-	-	-
	CURRENT_BALANCE	NUMBER	-	6	2	-	-	-	-
	SRE_ID	VARCHAR2	4	-	-	-	✓	-	-
	TEM_ID	VARCHAR2	4	-	-	-	✓	-	-
	LOYALTY_CARD_NUMBER	VARCHAR2	6	-	-	-	✓	-	-
	MOBILE_NUMBER	NUMBER	-	11	0	-	✓	-	-

2.

ALTER TABLE customers DROP (mobile_number);

3.

Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default	Comment
CUSTOMERS	CTR_NUMBER	VARCHAR2	6	-	-	1	-	-	-
	EMAIL	VARCHAR2	50	-	-	-	-	-	-
	FIRST_NAME	VARCHAR2	20	-	-	-	-	-	-
	LAST_NAME	VARCHAR2	30	-	-	-	-	-	-
	PHONE_NUMBER	VARCHAR2	11	-	-	-	-	-	-
	CURRENT_BALANCE	NUMBER	-	6	2	-	-	-	-
	SRE_ID	VARCHAR2	4	-	-	-	✓	-	-
	TEM_ID	VARCHAR2	4	-	-	-	✓	-	-
	LOYALTY_CARD_NUMBER	VARCHAR2	6	-	-	-	✓	-	-