

## Лабораторная работа №1

### Создание базы данных средствами СУБД Oracle

Существует три способа создания базы данных Oracle:

1. используя мастер создания базы данных Oracle Database Configuration Assistant;
2. при помощи специальных процедур создания баз данных;
3. с помощью SQL-команды CREATE DATABASE.

Для создания базы данных воспользуемся первым способом. Выберем программу Database Configuration Assistant, используя меню Пуск-Программы- Configuration and Migration Tools (Рис.1).

Рис. 1. Запуск мастера Database Configuration Assista



nt

При запуске мастера Database Configuration Assistant щелкните на кнопке Next (Далее), затем выберите Create a database(Создать базу данных) и General Purpose (Общего назначения). В текстовом поле Global Database

Name (Глобальное имя базы данных) введите имя, например STUDENT. Далее выберите Dedicated Server Mode (Режим выделенного сервера) и Typical (Обычная). Теперь щелкните по кнопке Finish (Готово), а затем OK, и Oracle создаст базу данных с заданными по умолчанию размерами и местоположением файлов. После создания базы данных Oracle предложит ввести пароли для учетных записей SYS и SYSTEM. Выберите подходящие пароли и далее щелкните по кнопке Exit (Выход). Управление базой данных Oracle происходит двумя способами: из командной оболочки SQL\*Plus (меню Пуск- Программы - Oracle-Application Development) или используя для этого программу Enterprise Manager Console (меню Пуск - Программы -Oracle -Enterprise Manager).

## **Работа с программой Enterprise Manager Console**

### **Создание таблиц базы данных**

Базами данных можно управлять с помощью программы Enterprise Manager Console. Данная программа предоставляет графический интерфейс для выполнения операций создания, редактирования таблиц, табличных областей, представлений, индексов и т.д. Запустив программу, вы увидите список имеющихся баз данных на вашем компьютере (Рис.2).

Практически все данные в базе данных хранятся в таблицах. Чтобы увидеть таблицы, созданные в базе данных, щелкните на узле Schema и на ту учетную запись, таблицы которой вы хотите просмотреть, например SYSTEM. Для изменения и просмотра свойств таблицы, щелкните по ее имени. В правой панели появится список полей с указанием типов данных, ограничений и т.д. (Рис.3) Данные свойства можно изменять. Сохранить изменения можно кнопкой Apply (применить). Если необходимо отказаться от изменений, то для этого можно воспользоваться кнопкой Revert (Отменить).

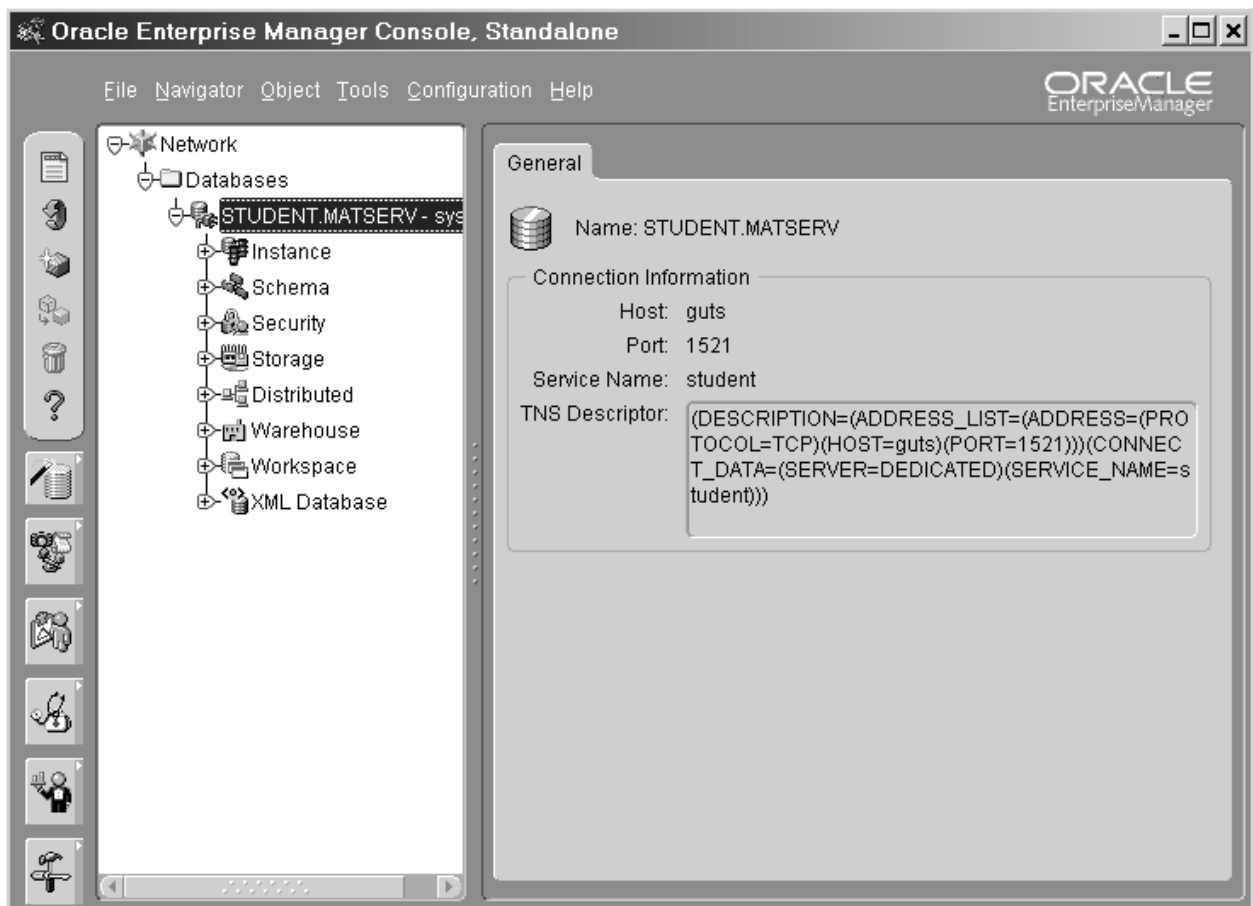


Рис. 2. Открытие базы данных STUDENT

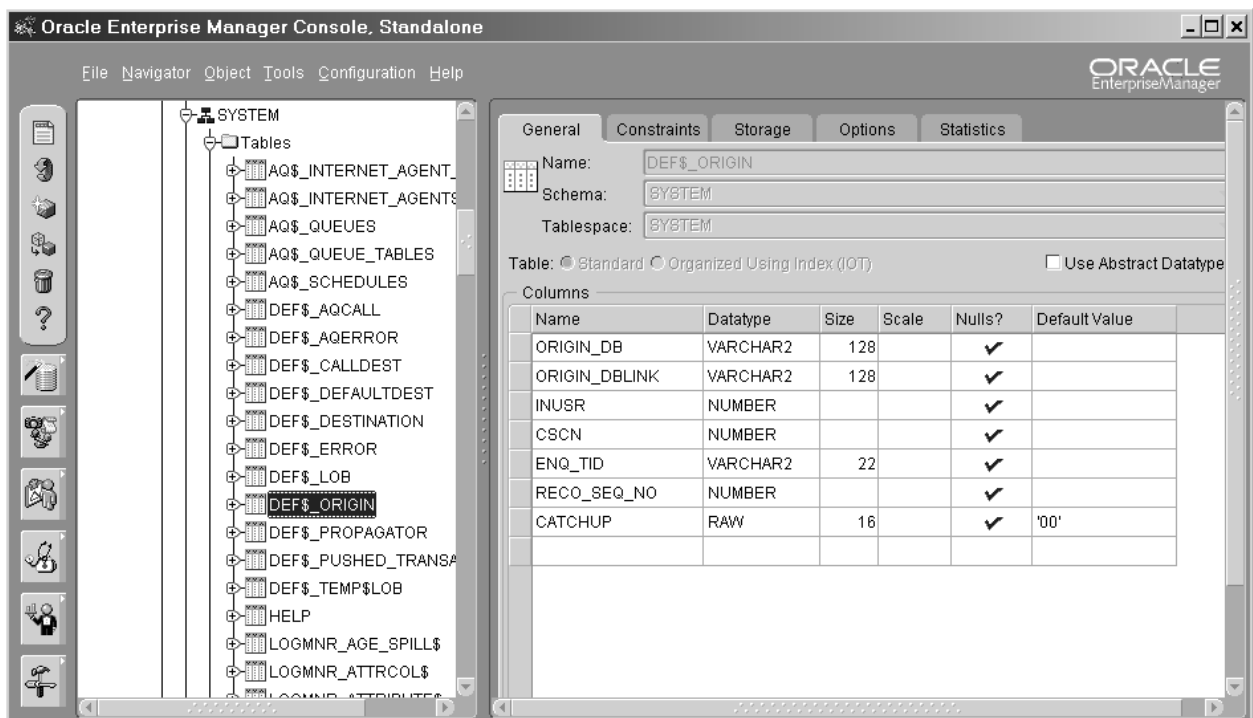


Рис. 3. Просмотр столбцов таблицы DBFS ORIGIN

СУБД Oracle позволяет работать с множеством типов данных. Ниже содержится список типов данных с которыми работает Oracle (таб.№1):

Таблица№1. Описание типов данных

Тип	Описание типа
CHAR(n)	Символьное поле длиной до 2000 символов
DATE	Поле даты со значением от 1 января 4712 года до н.э.до 31 декабря 4712 года н.э.
NUMBER(n, m)	Числовое поле длиной до 38 позиций
VARCHAR2(n)	Символьное поле переменной длины длиной до 4000 символов
LONG	Символьное поле переменной длины до 2 гигабайт
GLOB	Однобайтовое символьное поле длиной до 4 гигабайт
NLOB	Двухбайтовое символьное поле длиной до 4 гигабайт
RAW (и LONG RAW)	Неструктурированное двоичное поле длиной до 2000 байт
BLOB	Двоичное поле длиной до 4 гигабайт
BFILE	Хранимое во внешнем файле двоичное поле длиной до 4 гигабайт

### Пример.

Создадим таблицу STUDENT\_DATA под учетной записью IVANOV, в табличном пространстве SPISOK следующего содержания (таб.№2):

Таблица №2. Данные для таблицы STUDENT\_DATA

название поля	тип данных	размер поля	описание поля
NOM_STUDENTA	числовой	10	первичный ключ, поле уникальное
SURNAME	символьный	20	
ADRESS	символьный	30	
DATE BORN	дата		
GROUP	символьный	10	



## Создание табличного пространства SPISOK

Вся информация базы данных Oracle располагается в табличных пространствах. База данных может состоять как из одного, так и из нескольких табличных пространств. Табличное пространство - это один или несколько файлов данных. А файлы данных могут храниться на нескольких физических дисках. Один и тот же файл не может принадлежать одновременно нескольким табличным пространствам. При установке Oracle по умолчанию создается табличное пространство SYSTEM. В табличном пространстве размещаются любые объекты базы данных: таблицы, индексы, представления и т.д.

Для создания табличного пространства выберите в меню Object>Create. В открывшемся списке выберите элемент Tablespace и нажмите кнопку Create (Рис.4).

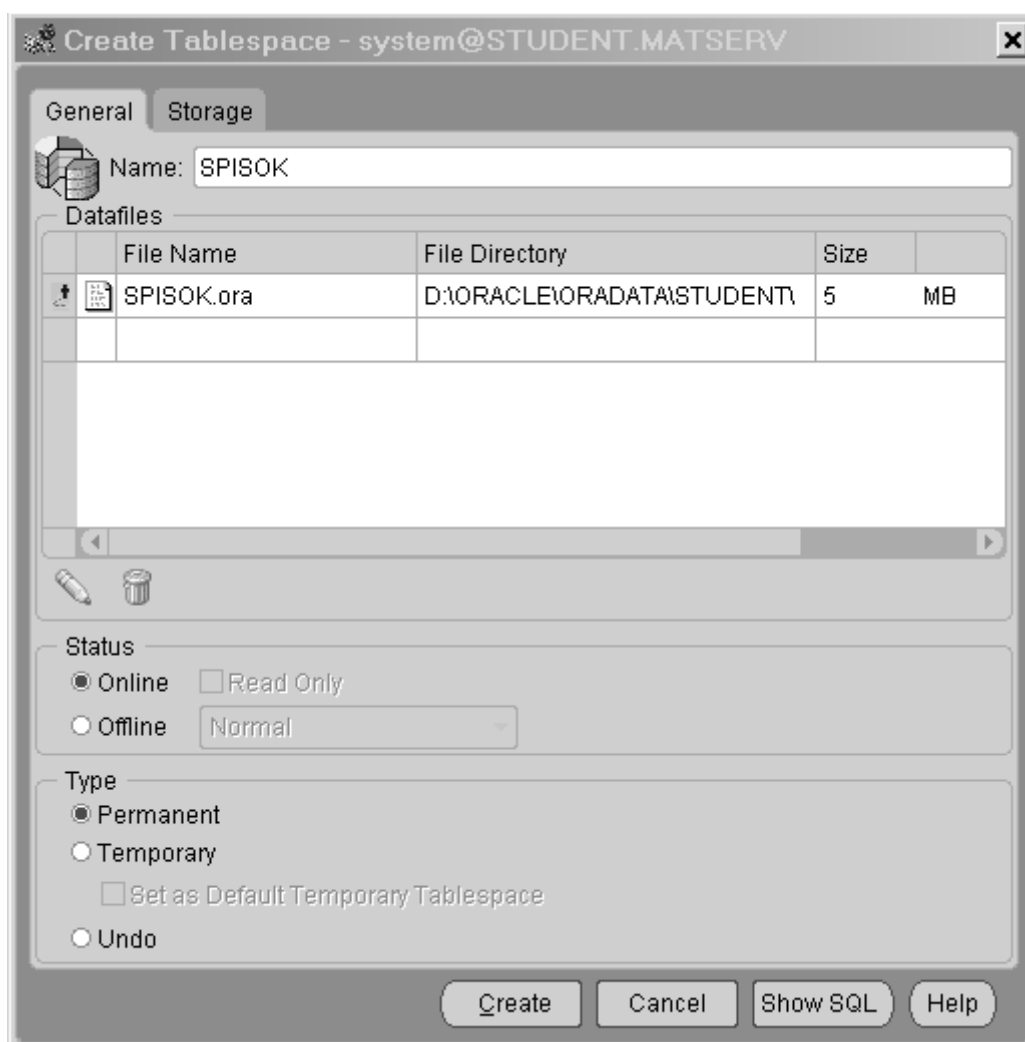


Рис. 4. Создание табличного пространства SPISOK

В окне в поле Name впишите название табличного пространства SPISOK, нажмите кнопку Create. По умолчанию создастся табличное пространство в файле SPISOK.ora и размером в 5 Мб.

### **Создание таблицы STUDENT\_DATA**

1. Процесс создания новой таблицы начинается с выбора команды Create... меню Object.
2. В списке выберите элемент Table, далее нажмите кнопку Create;
3. В окне Table Wizard, step 1 of 13: Introduction в поле What do you want the Name of the new table to be? Вписать название создаваемой таблицы STUDENT\_DATA, в поле Which Schema do you want the table to be part of? выберите название схемы IVANOV, в поле Which Tablespace do you want to create the table in? выберите название табличной области SPISOK, после нажмите кнопку Следующий;
4. В окне Table Wizard, step 2 of 13: Columns Definition в поле Column Name: (имя поля) впишите SURNAME, в поле Column Datatype: (тип поля) выберите VARCHAR2 и в поле (размер поля) укажите 20, нажмите кнопку Add(Добавить): в окне Columns defined зафиксированное введенное поле, остальные поля вводятся аналогично, далее нажмите кнопку Следующий;
5. В окне Table Wizard, step 3 of 13: Primary Key Definition установите переключатель на Yes, I want to create a primary key (Создать первичный ключ). Щелкните мышкой в колонке Order напротив записи NOM\_STUDENTA, далее нажмите кнопку Следующий;
6. В окне Table Wizard, step 4 of 13: Null and Unique Constraints установите переключатели на No, it cannot be Null (пустые записи недопустимы) и на Yes, it must be Unique (запись уникальна) для поля NOM\_STUDENTA, далее нажмите кнопку Следующий;
7. В окне Table Wizard, step 5 of 13: Foreign Constraint установите переключатель на No, the column is not a Foreign Key (поле не является внешним ключом), далее нажмите кнопку Следующий;
8. В окне Table Wizard, step 6 of 13: Check Constraints, если на поле накладываются какие-либо ограничения, то установите переключатель на Yes, the column has a Check Condition, введите имя ограничения в поле Constraint Name и в поле What is the Check Condition for this column? впишите ограничение на поле по примеру, расположенному ниже в этом окне, если ограничения отсутствуют, то выберите No, the column have a Check Condition, далее нажмите кнопку Следующий;
8. В окне Table Wizard, step 13 of 13: Summary просмотрите код создания таблицы на языке SQL, нажмите кнопку Готово.



## Создание индексов в таблице

Индексы необходимы для сортировки и быстрого поиска информации в базе данных, они также обеспечивают уникальность столбцов. Индексы — это главный инструмент повышения производительности любой базы данных.

1. Выберите команду Create... меню Object;
2. В открывшемся списке выберите элемент Index и нажмите кнопку Create;
3. В окне Create Index - IVANOV@STUDENT вписать название создаваемого индекса, например NOM\_SURNAME, выберите название схемы IVANOV, название табличной области SPISOK и таблицу в которой создается индекс, т.е. STUDENT\_DATA. В нижней части окна выберите поля (NOM\_STUDENTA, SURNAME) в нужной последовательности, после нажмите кнопку Create.

Всю информацию о созданном индексе можно просмотреть, щелкнув правой кнопкой мыши по индексу и выбрав команду View /Edit Details...

## Редактирование таблиц базы данных

С помощью Enterprise Manager Console можно также просматривать и редактировать содержимое таблиц. Для этого необходимо щелкнуть правой кнопкой мыши по таблице и выбрать команду View /Edit Contents...(Рис.5).

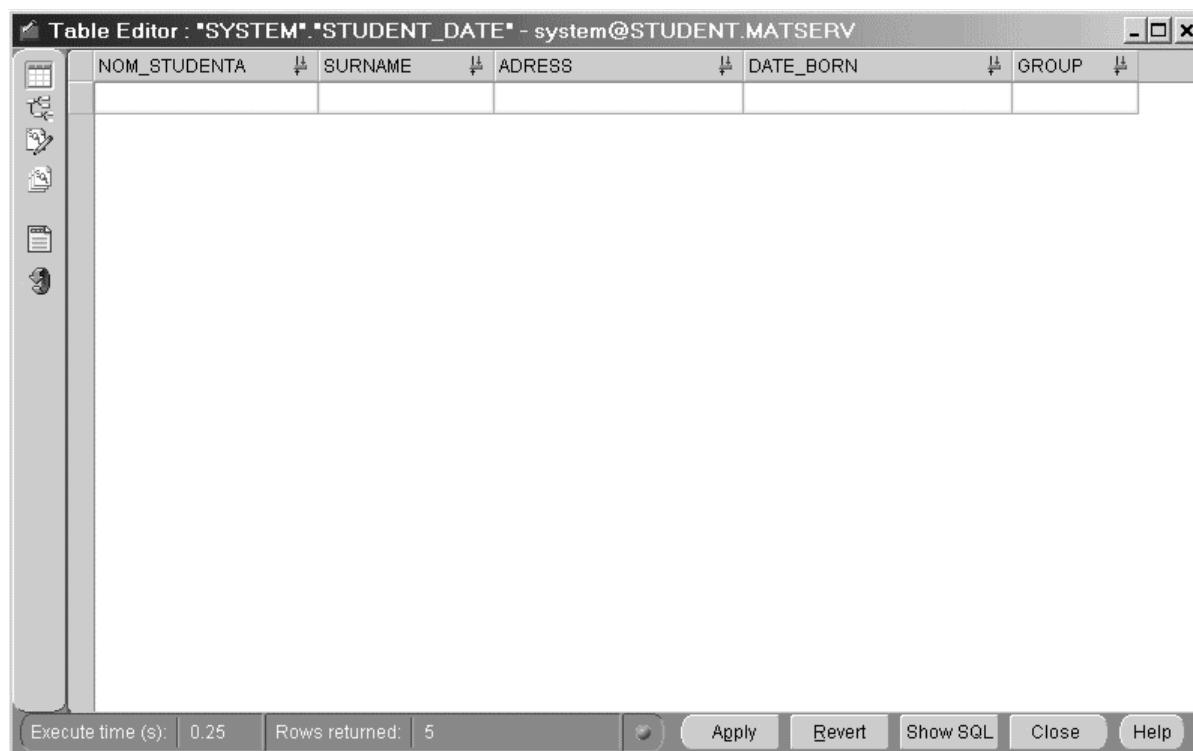


Рис. 5. Редактирование таблицы STUDENT\_DATA

## Замечание

При введении значений в поле типа DATE, необходимо использовать стандартный формат даты ORACLE. Стандартный формат даты для ввода и вывода имеет вид DD-MON-YY 12:00:00 AM, например: 13-ноя-1992 12:00:00 AM.

## **РАБОТА С ПРОГРАММОЙ SQL\*PLUS**

### **Создание таблиц базы данных с помощью программы SQL\*Plus**

Для **создания таблицы** в Oracle используется следующий оператор:

```
CREATE TABLE <имя_схемы.имя_таблицы>
(<имя_столбца> <тип_столбца> [NOT NULL]
[UNIQUE],...,
[ CONSTRAINT <имя_ограничения>
PRIMARY KEY(<имя_столбца>)],
[CONSTRAINT <имя_ограничения>
FOREIGN KEY(<имя_столбца>)
REFERENCES <имя_схемы.имя_мастер_таблицы>
(<имя_столбца >)],
[CONSTRAINT <имя_ограничения>
CHECK (ограничения на столбец)])
[TABLESPACE (имя_табличной_области)].
```

- NOT NULL - в этом случае элементы столбца всегда должны иметь определенное значение, т.е. не пусто;
- UNIQUE - значение каждого элемента столбца должно быть уникальным;
- CONSTRAINT <имя\_ограничения>

PRIMARY KEY (<имя\_столбца>)- столбец является первичным ключом;

• CONSTRAINT <имя\_ограничения>  
FOREIGN KEY (<имя\_столбца>)  
REFERENCES <имя\_схемы.имя\_мастер\_таблицы>

(<имя\_столбца >) - определяет, что данный столбец является внешним ключом и указывает на ключ какой мастер \_ таблицы он ссылается;

- CONSTRAINT <имя\_ограничения>

CHECK (ограничения на столбец) - определяет какие значения может принимать данный столбец,

например

CHECK(NOM\_STUD between 1111 and 2222) -

поле NOM\_STUD может принимать только значения от 1111 до 2222,

CHECK(GROUP in ('CB - 402','CB - 101')) -

поле GROUP может принимать только значения CB -402 или CB - 101.

Имя ограничения выбирается произвольно, хотя пользователь может его не указывать, тогда СУБД создаст имя ограничения самостоятельно. Oracle позволяет создавать композитные ключи, т.е. ключ состоящий из двух и более атрибутов, например

CONSTRAINT <имя\_ограничения>

PRIMARY KEY (<имя\_столбца>,<имя\_столбца>,...).

**Удаление таблицы:** DROP TABLE <имя\_таблицы>

При удалении таблицы удаляются все ее данные, индексы и привилегии, которые были даны пользователям для работы с ней. Остальные объекты, связанные с данной таблицей не удаляются, но перестают действовать.

Для **добавления столбцов**, используется команда:

ALTER TABLE <имя\_схемы.имя\_таблицы> ADD

(<имя\_столбца> <тип\_столбца> [NOT NULL]

[UNIQUE],...,

[ CONSTRAINT <имя\_ограничения>

PRIMARY KEY(<имя\_столбца>)],

[ CONSTRAINT <имя\_ограничения>

FOREIGN KEY(<имя\_столбца>)

REFERENCES <имя\_схемы.имя\_мастер\_таблицы>

(<имя\_столбца >)],

[CONSTRAINT <имя\_ограничения>

CHECK (ограничения на столбец))].

Чтобы **удалить столбцы** воспользуемся следующей командой:

ALTER TABLE <имя\_схемы.имя\_таблицы> DROP

(<имя\_столбца>) [CASCADE CONSTRAINTS].

CASCADE CONSTRAINTS - автоматически удаляются все ограничения, связанные с этим столбцом.

#### Создание **индексов**.

Для создания индекса, воспользуемся командой языка SQL:

```
CREATE INDEX «имя_схемы.имя_индекса»  
ON <имя_схемы.имя_таблицы> (<имя_столбца>,...)  
TABLESPACE (имя_табличной_области).
```

При создании уникального индекса необходимо вставить слово UNIQUE перед INDEX, т.е.

```
CREATE UNIQUE INDEX «имя_схемы.имя_индекса»  
ON <имя_схемы.имя_таблицы> (<имя_столбца>,...)  
TABLESPACE (имя_табличной_области).
```

Уникальный индекс гарантирует, что не будет вставлена запись с одним и тем же значением по столбцам составляющим индекс.

**Пример.** Создадим следующую таблицу(таб. №3)под учетной записью IVANOV, в табличном пространстве SPISOK.

Таблица №3. Данные для таблицы **KURS\_RABOTA**

название поля	тип данных	размер поля	описание поля
NOM_STUDENTA	числовой	10	внешний ключ к таблице STUDENT_DATA, поле не уникальное
NAME_KURS_RAB	символьный	40	

## Создание таблицы KURS\_RABOTA.

1. Запустите программу SQL\*Plus, подключившись к базе данных STUDENT;
2. Введите команду создания таблицы:

```
SQL> CREATE TABLE IVANOV.KURS_RABOTA  
(NOM_STUDENTA NUMBER(10) NOT NULL,  
NAME_KURS_RAB VARCHAR2(40) NOT NULL,  
CONSTRAINT NOM_STUD  
FOREIGN KEY(NOM_STUDENTA) REFERENCES  
IVANOV.STUDENT_DATA(NOM_STUDENTA))  
TABLESPACE SPISOK;
```

3. Нажмите клавишу ввода (Рис.6).

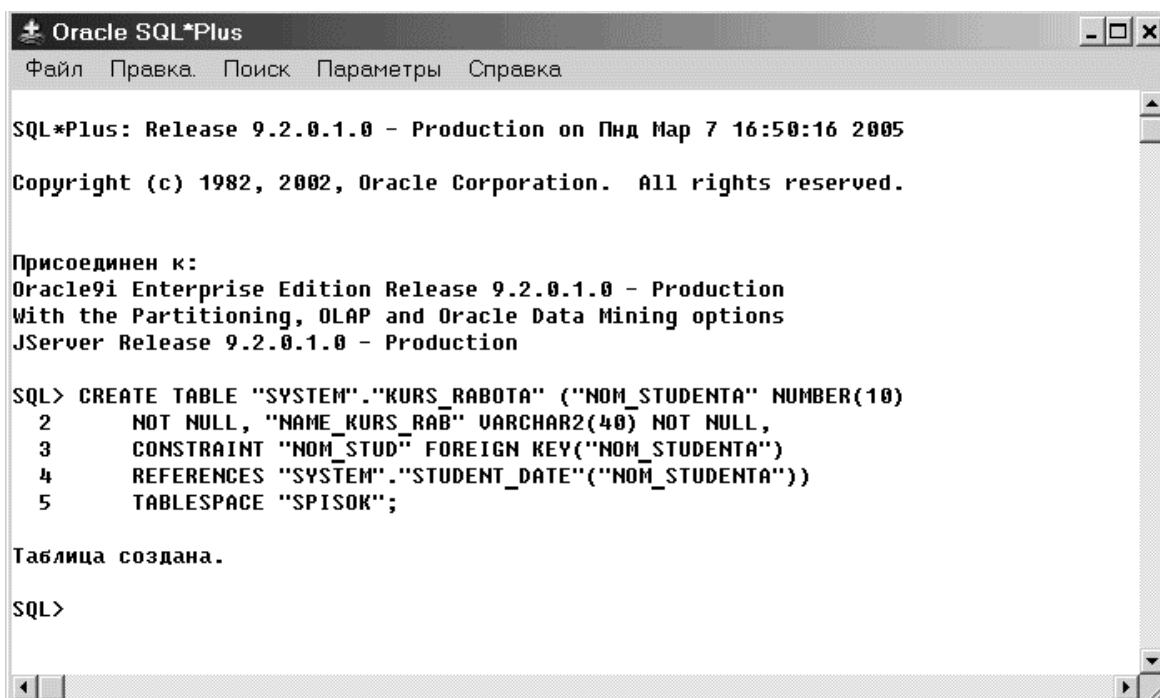


Рис. 6. Создание таблицы KURS\_RABOTA в программе SQL\*Plus

Проверьте структуру созданной таблицы задав команду: DESCRIBE IVANOV.KURS\_RABOTA.

## Задание к лабораторной работе №1

Спроектируйте базу данных, состоящую из нескольких таблиц таким образом, чтобы были использованы связи один к одному, один ко многим. При создании таблиц воспользуйтесь программами Enterprise Manager Console и SQL\*Plus.

#### **Вариант №1**

Спроектируйте базу данных, которая используется для автоматизации технологического процесса в крупной фирме, осуществляющей оптовую торговлю промышленными товарами.

#### **Вариант №2**

Спроектируйте базу данных по складу продовольственных товаров. В базе данных должны храниться данные о поставщиках, заказчиках и движении товара между складом и магазинами.

#### **Вариант №3**

Спроектируйте базу данных Предприятия Омска, в которой содержится информация о руководителе предприятия, о поставке и заказе товаров, какие и с кем были заключены контракты.

#### **Вариант №4**

Создать базу данных, позволяющую вести учет движения товаров на складе, где отобразите взаимодействие между различными отделами, осуществляющими работу склада.

#### **Вариант №5**

Спроектировать базу данных Сотрудники образовательного учреждения. В базе данных хранятся данные о самих преподавателях, о том где и какие предметы они ведут и их нагрузке.

#### **Вариант №6**

Спроектируйте базу данных, содержащую информацию о читателях и каталоге книг библиотеки. В базе данных также должны храниться сведения о том какие книги у читателя на руках и их дате возврата.

#### **Вариант №7**

Создать базу данных работы аптеки, которая принимает заказы лекарств по телефону и осуществляет доставку лекарств заказчикам.

#### **Вариант №8**

Спроектировать базу данных работы фотосалона, в котором принимаются заказы на печать уже готовых фотоснимков.

#### **Вариант №9**

Спроектировать базу данных опытного хозяйства по выращиванию цветов. В базе данных должны содержаться такие данные о цветах, как условия выращивания, время посадки, количество посаженных цветов и т.д.

#### **Вариант №10**

Создать базу данных работы кассы железнодорожного вокзала. В базе данных содержится информация о количестве свободных мест по различным направлениям, кем и когда куплены билеты.

**Вариант №11**

Спроектировать работу телефонной службы. В базе данных должна находиться информация об абонентах сети, тарифах и услугах службы.

**Вариант №12**

Спроектировать базу данных банка. В базе данных должны храниться данные о вкладчиках, счетах и банковских операциях.

**Вариант №13**

Создать базу данных Продажа товаров в супермаркете, в которой отражаются данные о продуктах, реализуемых данным супермаркетом и о поставщиках этих продуктов.