**Лабораторная работа №1**

***Тема: «Изучение возможностей СУБД Oracle по созданию и управлению базами данных»***

**Oracle Database 11g Express Edition** (**Oracle Database XE**) - это СУБД начального уровня, основанная на программном коде **СУБД Oracle Database 11g Release 2**. Данная СУБД бесплатна для разработки, развертывания и продажи, быстро скачивается и проста в администрировании. **Oracle Database XE** может быть установлена на любой компьютер с любым количеством процессоров (одна база данных на машину), но с ограничением в 4Гб пользовательских данных, использует не более 1Гб оперативной памяти и только один из имеющихся процессоров.

**Oracle Database XE** имеет следующие преимущества:

* абсолютно свободная СУБД;
* можно тестировать, разрабатывать и распространять с нулевыми инвестициями в ПО и без риска;
* легко мигрировать на промышленные редакции;
* не требуется переписывание приложений при переходе на промышленные редакции.

Для работы с **Oracle Database XE** используется интуитивно-понятный web-интерфейс, позволяющий:

* администрировать базу данных;
* создавать таблицы, представления и другие объекты базы данных;
* импортировать, экспортировать и просматривать табличные данные;
* выполнять запросы и запускать SQL-скрипты;
* генерировать отчеты.

В поставку **Oracle Database XE** также включен **Oracle Application Express release 2.1** - графическая среда разработки для создания web-приложений, основанных на базе данных.

В поставку **Oracle Database XE** также входят следующие консольные приложения:

* SQL-консоль (**SQL\*Plus**), для выполнения команд SQL и PL/SQL, и запуска скриптов;
* **SQL\*Loader**, для загрузки данных в базу данных;
* **Data Pump** и инструменты импорта/экспорта данных.

Отправной точкой web-интерфейса для работы с **Oracle Database XE** является домашняя страница базы данных. Крупные пиктограммы домашней страницы базы данных имеют следующее функциональное назначение:

**Administration** - управление пользовательскими учетными записями, распределением оперативной памяти; управление дисковым пространством, сетевыми соединениями, мониторинг активности базы данных, а также просмотр конфигурационной информации.

**Object Browser** - просмотр, создание, изменение и удаление объектов базы данных, а также навигация по ним. Чтобы редактировать и компилировать процедуры, функции и триггеры, пользуйтесь PL/SQL-редактором.

**SQL** - доступ к следующим инструментам SQL:

**SQL Commands** - запуск команд SQL и анонимных блоков PL/SQL. Запуск скрпитов и сохраненных запросов.

**SQL Scripts** - создание, редактирование, просмотр, запуск и удаление скриптовых файлов. Загрузка и выгрузка скриптов для локальной файловой системы.

**Query Builder** - позволяет с небольшим опытом или совсем без знания языка SQL строить SQL-запросы, используя графический пользовательский интерфейс. Просмотр отформатированных результатов запроса и сохранение запросов.

**Utilities** - загрузка и выгрузка данных, генерирование DDL, просмотр отчетов по объектам, а также восстановление удаленных объектов базы данных.

Для работы с СУБД **Oracle Database XE** также можно воспользоваться программой-интерпретатором **SQL\*Plus**. Утилита **SQL\*Plus** позволяет выполнять команды **SQL** и блоки **PL/SQL**, а также решать ряд других задач. С помощью **SQL\*Plus** можно:

- вводить, редактировать, запоминать, загружать и выполнять команды **SQL** и блоки **PL/SQL**;

- форматировать, создавать, сохранять, печатать и публиковать в web результаты выполнения запросов (отчеты);

- получать описание (имена и типы столбцов) любой таблицы и представления;

- обращаться к удаленным базам данных и копировать из них данные;

- посылать и принимать сообщения от конечных пользователей;

- администрировать базу данных.

Для запуска утилиты **SQL\*Plus** используется команда операционной системы, обычно - **sqlplus**. Запуск выполняется из командного интерпретатора или окна командной строки. Для доступа к данным необходимо сначала подключиться к базе данных. При запуске **SQL\*Plus** обычно происходит подключение к стандартной базе данных с указанным именем пользователя и паролем. После запуска и успешной регистрации можно в любой момент подключиться к схеме другого пользователя или к другой базе данных.

Каждая база Oracle с момента своего создания содержит две схемы (сле­дует отметить, что термины «учетная запись», «схема» и «пользователь» обозначают в Oracle одно и то же) — SYS и SYSTEM. Схема SYS содер­жит все системные объекты — внутренние таблицы базы данных, пакеты, процедуры. Крометого, пользователь SYS является владельцем словаря данных. Словарь данных Oracle — это совокупность таблиц и представле­ний, позволяющих получить любую информацию о структуре базы дан­ных, о ее настройках и состоянии при помощи стандартных SQL-запросов. Учетная запись SYS является также учетной записью администратора ба­зы данных с неограниченными полномочиями. Учетная запись SYSTEM предоставляет доступ ко всем объектам базы и наделена ролью DBA.

Для соединения с базой данных используется команда **CONNECT**. Команда **CONNECT** имеет следующий синтаксис:

**CONNECT** [<пользователь и база данных>] [AS <роль пользователя>]

**Роль пользователя** – это именованная совокупность полномочий. Использование ролей существенно облегчает управление полномочиями. И полномочия, и роли предоставляются пользователям оператором **GRANT**, а отбираются, соответственно - **REVOKE**.

После создания базы данных в ней создается несколько стандартных ролей, например, *CONNECT* - содержит только одну привилегию CREATE SESSION, позволяющую создавать соединение с базой, *DBA* - полный набор привилегий, необходимых администратору базы, *RESOURCE* - базовый набор привилегий, необходимых разработчику и т.д.

**Примеры:**

**CONNECT system/11**

**--**подключаемся пользователем **system** с паролем **11**

**CONNECT system/11 as sysdba**

**--** подключаемся пользователем **system** с паролем **11** с ролью **SYSDBA**

**CONNECT system/11@myOra11**

**--** подключаемся пользователем **system** с паролем **11** к базе данных **myOra11**

Для выхода из **SQL\*Plus** необходимо ввести команду **EXIT**. Будет выдана информация о версии сервера, к которому была подключена утилита, и о версии **SQL\*Plus**. Затем работа программы завершается и либо закрывается окно, либо вы получаете приглашение командной строки операционной системы.

**Oracle Database XE** запускается автоматически сразу после инсталляции и после каждой системной перезагрузки. Таким образом, нет необходимости запускать базу данных, если предварительно ее не остановили. Вы можете запустить/остановить базу данных непосредственно с рабочего стола (**Programs**-> **Oracle Database 10g Express Edition**->**Start/Stop Database**) или при помощи командной строки **SQL (SQL\*Plus)**. Для запуска/остановки базы данных с помощью **SQL\*Plus** используются команды **STARTUP/SHUTDOWN IMMEDIATE.**

Для создания табличного пространства необходимо использовать команду **CREATE TABLESPACE**:

**CREATE TABLESPACE <**имя\_табл\_пространства>

**DATAFILE [**спецификация\_файла**]**

**[**дополнительные\_конструкции**]**

**Примеры:**

**CREATE** **TABLESPACE** **ts1**

**DATAFILE** **'/mydata/myfileTS1.dat' SIZE 100M** --создание табличного пространства **ts1** размером **100 Мбайт**

**CREATE TABLESPACE ts1**

**DATAFILE '/mydata/myfileTS1.dat' SIZE 100M**

**REUSE AUTOEXTEND ON NEXT 10M MAXSIZE 200M**

--создание табличного пространства **ts1** размером **100 Мбайт** и возможностью расширения до **200 Мбайт**

Для создания нового пользователя существует команда **CREATE USER:**

**CREATE USER** пользователь **IDENTIFIED BY** пароль

**[DEFAULT TABLESPACE** tablespace

**QUOTA целое число [K][M] ON** tablespace

**TEMPORARY TABLESPACE** tablespace

**QUOTA целое число [K][M] ON tablespace]**

**Примеры:**

**CREATE USER Student1 IDENTIFIED BY 123**

-- создание пользователя с именем “**Student1**” и паролем “**123**”

**CREATE USER Student1 IDENTIFIED BY 123**

**DEFAULT TABLESPACE ts1**

-- создание пользователя с именем “**Student1**” и паролем “**123**”, для объектов которого табличным пространством по умолчанию назначается табличное пространство **ts1.**

*GRANT R1 TO Stud;*  
- Предоставит полномочия роли *R1* пользователю Stud.

***Задание на лабораторную работу.***

1. Запустите утилиту **SQL\*Plus** и подключитесь к базе данных под пользователем **SYS** или **SYSTEM.**
2. Создайте нового табличное пространство с именем «**TS\_фамилия\_номер\_группы\_год**» размером не **более** **200 Мбайт**.
3. Создайте нового пользователя с именем   
   «**NNN\_номер\_группы**» (**NNN** – ваши инициалы). Установите в качестве табличного пространства по умолчанию табличное пространство, созданное на предыдущем шаге.
4. Назначьте новому пользователю роли **CONNECT** и **RESOURCES**.
5. Выполните соединение с БД, указав имя пользователя и пароль, созданные на предыдущем шаге.
6. Отключите базу данных и запустите ее вновь. Сделайте это с помощью утилиты **SQL\*Plus** и с помощью команд **Start/Stop Database** (для пользователя с правами admin в Windows).
7. Ознакомьтесь с web-интерфейсом для работы с **Oracle Database XE**. Поменяйте с помощью него пароль для вашего пользователя.

**Лабораторная работа №2**

***Тема: «Создание таблиц»***

Таблицы представляют собой механизм сохранения информации в базе данных **Oracle**. Структура таблицы включает перечень имен полей (столбцов) таблицы. Столбец характеризуется типом данных и длиной. Для столбцов типа **NUMBER** можно задать дополнительные характеристики точности и масштаба. Точность определяет число значащих цифр. Масштаб показывает место десятичной точки. Основные типы полей приведены ниже:

|  |  |
| --- | --- |
| **CHAR** | Символьное поле фиксированной длины до 2000 байт; |
| **NCHAR** | Поле фиксированной длины для набора символов, состоящих из нескольких байт. Максимальный размер - 2000 символов или 2000 байт в зависимости от набора символов; |
| **VARCHAR2** | Символьное поле переменной длины до 4000 символов; |
| **NVARCHAR2** | Поле переменной длины для набора символов, состоящих из нескольких байт. Максимальный размер - 4000 символов или 4000 байт в зависимости от набора символов; |
| **DATE** | 7-байтовое поле фиксированной длины, используемое для хранения дат; |
| **NUMBER** | Числовой столбец переменной длины; |
| **LONG** | Поле переменной длины, до 2 Гбайт; |
| **RAW** | Поле переменной длины (до 2 000 байт), используемое для хранения двоичных данных; |
| **LONG RAW** | Поле переменной длины (до 2 Гбайт), используемое для хранения двоичных данных; |
| **BLOB** | Двоичный большой объект длиной до 4 Гбайт; |
| **CLOB** | Символьный большой объект длиной до 4 Гбайт; |
| **NCLOB** | Тип данных CLOB для набора символов, состоящих из нескольких байт; длина до 4 Гбайт; |
| **BFILE** | Внешний двоичный файл; размер ограничивается операционной системой; |
| **ROWID** | Двоичные данные, представляющие идентификатор RowID. Все RowID занимают 6 байт для нормальных индексов в таблицах, не разбитых на разделы; локальных индексов в таблицах, разбитых на разделы; и указателей строк, используемых для строк с указателями или с перегруппировкой. RowID занимает 10 байт только для глобальных индексов в таблицах, разбитых на разделы; |
| **UROWID** | Двоичные данные, используемые для адресации данных; длина до 4000 байт. Могут поддерживать как логические, так и физические значения RowID, а также внешние таблицы, доступ к которым осуществляется через шлюз. |

Таблицы, принадлежащие **SYS**, называются таблицами словаря данных. Они содержат системный каталог, с помощью которого система управляет своей работой. Таблицы можно создавать с помощью web-интерфейса **Oracle Database XE** или вручную, пользуясь средствами **SQL\*Plus**. Их можно связывать друг с другом. База данных реализует эти отношения с помощью ограничений ссылочной целостности. На столбцы таблицы можно накладывать ограничения; при этом каждая ее строка должна удовлетворять указанному в описании ограничению.

Создание нового отношения (таблицы) выполняется с помощью команды **DDL (Data Definition Language)** **CREATE TABLE.** Упрощенный синтаксис этой команды выглядит следующим образом:

**CREATE TABLE** **<*имя таблицы***>

({<***имя поля***> <***тип данных***> [(<***размер***>)]

[<***ограничения целостности поля***> …]},..

[,<***ограничения целостности таблицы***>,…])

[**TABLESPACE** <***имя табличного пространства***>]

[**STORAGE** <***параметры хранения таблицы***>];

Здесь:

<***имя поля***> - имя поля (столбца) таблицы;

<***тип данных***> - один из выше перечисленных типов данных;

<***размер***> - размер поля в символах;

<***ограничения целостности поля***> - предполагает использование следующих ограничений:

**primary key** – первичный ключ (обязательный и уникальный);

**unique** – уникальное значение поля в пределах таблицы;

**[not] null** – [не] возможность не указывать значение поля;

**check(<***условие***>)** – проверка условия для поля (полей);

**default**<*выражение*> - задание значения поля по умолчанию;

**references**<*имя таблицы*>[<*имя столбца*>] - внешний ключ;

<***ограничения целостности таблицы***> - те же ограничения, что и для поля и дополнительно:

**foreign key** [<*список полей*>, …] **references** <*имя таблицы*> [(<*список полей*>)] - внешний ключ

Примеры:

**CREATE TABLE** **tab1**

**(**

**id NUMBER(6) PRIMARY KEY,**

**name CHAR(6)**

**); --** создание таблицы с именем «**tab1**» и двумя столбцами, один из которых («**id**») числовой и является первичным ключом, другой («**name**») – символьный (максимальная длина 6 символов).

**CREATE TABLE tab2**

**(**

**id NUMBER(6) PRIMARY KEY,**

**class NUMBER(3),**

**fdata DATE,**

**group CHAR(6),**

**FOREIGN KEY (class, fdata) REFERENCES exam(class, fdata)); --** создание таблицы с именем «**tab2**» и четырьмя столбцами, два из которых («**id**» и «**class**») числовые, один («**fdate**») – типа даты, один («**group**») – символьный. Первичным ключом назначено поле «**id**». Также введен составной внешний ключ, состоящий из двух полей «**class**» и «**fdata**», для связи с таблицей «**exam».**

**CREATE TABLE** **tab3**

**(**

**id NUMBER(6) PRIMARY KEY CHECK(id>1000),**

**name CHAR(6) DEFAULT “Petrov”**

**); --** создание таблицы с именем «**tab3**» и двумя столбцами, один из которых («**id**») числовой и является первичным ключом, другой («**name**») – символьный (максимальная длина 6 символов). Также на поля наложены ограничения: в поле «**id»** можно ввести только значения больше **1000**; для поля «**name»** определено значение, вводимое по умолчанию «**Petrov**».

***Задание на лабораторную работу.***

1. Создайте первую таблицу в соответствии со своим вариантом (варианты заданий представлены в таблице 1.)
2. В соответствие с заданием выберите первичный ключ для вашей таблицы и задайте все указанные ограничения
3. Создайте вторую таблицу в соответствии со своим вариантом.
4. У второй таблицы выберите внешний ключ для её связи с первой таблицей.
5. Создайте нового пользователя с именем   
   «NNN1\_номер\_группы» в своем табличном пространстве. Разрешите ему работать с таблицами (просматривать, вставлять, обновлять и удалять записи).

Таблица 1. Варианты задания для выполнения лабораторной работы №2.

|  |  |
| --- | --- |
| Вариант | Создаваемые таблицы |
| 1. | 1. **Первая таблица STUDENTS**– таблица для хранения данных о студентах.  **Поля**:  **номер зачетной книжки** (**Num**), тип данных:  NUMBER(6, 0), первичный ключ, допустимые значения >100000;  **ФИО (Fname)**, тип данных: VARCHAR(100), обязательное поле;  **год поступления (Year)**, тип данных: NUMBER(4, 0), обязательное поле  **дата рождения (Bday)**, тип данных: DATE  **вид обучения (Plata)**, тип данных: NUMBER(1, 0), допустимые значения: 0, 1;  значение по умолчанию: 0 (т.е. нет платы).  **средний бал при поступлении (Mb)**, тип данных: NUMBER(3, 2)  **стоимость обучения в платной группе (Money)** , тип данных: NUMBER(5, 0), значение по умолчанию: 0 (т.е. нет платы).  **адрес (Address)**, тип данных: VARCHAR(200), обязательное поле  2. Вторая таблица **SESSION\_HISTORY –** таблица для хранения результатов сессий.  **Поля**:  **номер зачетной книжки** (**Num**), тип данных:  NUMBER(6, 0), внешний ключ для связи с таблицей **Stud**  **номер сессии (SessNum)**, тип данных: NUMBER(1), обязательное поле;  **оценка по предмету 1 (Sub1)**, тип данных:  NUMBER(1, 0), допустимые значений: 1, 2, 3, 4, 5, обязательное поле;  **оценка по предмету2 (Sub2)**, тип данных:  NUMBER(1, 0), допустимые значений: 1, 2, 3, 4, 5, обязательное поле;  **оценка по предмету 2 (Sub3)**, тип данных:  NUMBER(1, 0), допустимые значений: 1, 2, 3, 4, 5, обязательное поле; |
| 2. | 1. **Первая таблица EMPLOYEES**– таблица для хранения данных о сотрудниках фирмы.  **Поля**:  **Табельный номер** (**Num**), тип данных:  NUMBER(4, 0), первичный ключ;  **ФИО (Fname)**, тип данных: VARCHAR(100), обязательное поле;  **дата рождения (Bday)**, тип данных: DATE  **пол (Gender)**, тип данных: CHAR(1), допустимые значения: f, m;  значение по умолчанию: m  **должность (Job),** тип данных: VARCHAR(30), обязательное поле;  **количество ставок (WageRate)**,  тип данных: NUMBER(2, 1), допустимые значения: от 0.1 до 1.5, значение по умолчанию: 1;  **дата приема (Sdate)** , тип данных: DATE, обязательное поле;  **адрес (Address)**, тип данных: VARCHAR(200), обязательное поле;  2. Вторая таблица **JOB\_HISTORY –** таблица истории карьеры работников.  **Поля**:  **табельный номер работника** (**Num**), тип данных:  NUMBER(6, 0), внешний ключ для связи с таблицей **EMPLOYEES**  **дата начала работы на должности (StartDate)**, тип данных: DATE, обязательное поле;  **дата окончания работы на должности (EndDate)**, тип данных: DATE;  **должность (Job)**, тип данных: VARCHAR(30) , обязательное поле;  **номер отдела (NumDepartment)**, тип данных: NUMBER(2), ограничения на значения: >0; <=30; значение по умолчанию: 1. |
| 3. | 1. **Первая таблица MENU**– таблица для хранения данных о блюдах.  **Поля**:  **идентификатор блюда** (**ID**), тип данных:  NUMBER(3, 0), первичный ключ; допустимые значения: > 100  **Название блюда (Name)**, тип данных: VARCHAR(100), обязательное поле;  **общий вес в граммах (Mass)**, тип данных: NUMBER(4,0); значение по умолчанию: 100; допустимые значения: >0  **калорийность (Calories)**, тип данных: NUMBER(4,0); обязательно поле; допустимые значения: >0  **тип (Type),** тип данных: NUMBER(1),  **стоимость (Price)**, тип данных: NUMBER(6, 2), обязательно поле; допустимые значения: >0;  **время приготовления в минутах (Minutes)** , тип данных: NUMBER(3,0);  2. Вторая таблица **COMPOSITION –** таблица с составами блюд.  **Поля**:  **идентификатор блюда** (**ID**), тип данных:  NUMBER(3, 0), внешний ключ для связи с таблицей **MENU**  **ингредиент (Component)**, тип данных: VARCHAR(20), обязательное поле;;  **количество (Numb)**, тип данных: NUMBER(4,0), обязательное поле;  **единица измерения (Unit)**, тип данных: VARCHAR(30) , допустимые значения: gram, piece; значение по умолчанию: gram. |
| 4. | 1. **Первая таблица CLIENTS**– таблица для хранения данных о покупателях интернет-магазина.  **Поля**:  **идентификатор покупателя** (**ID**), тип данных:  NUMBER(6, 0), первичный ключ; допустимые значения: > 1000  **Имя (Login)**, тип данных: VARCHAR(100), обязательное поле;  **ФИО (Fname)**, тип данных: VARCHAR(100), обязательное поле;  **дата рождения (Bday)**, тип данных: DATE, обязательное поле;  **дата регистрации (Rday)**, тип данных: DATE, обязательное поле;  **пол (Gender)**, тип данных: CHAR(1), допустимые значения: f, m;  значение по умолчанию: m  **страна проживания (Country),** тип данных: VARCHAR(30), обязательное поле;  **адрес (Address)**, тип данных: VARCHAR(200), обязательное поле;  2. Вторая таблица **SHOPPING\_CART –** таблица с данными о покупках.  **Поля**:  **идентификатор покупки** (**ID\_buy**), тип данных:  NUMBER(3, 0)**;**  **идентификатор покупателя** (**ID**), тип данных:  NUMBER(6, 0), внешний ключ для связи с таблицей **CLIENTS;**  **товар (Goods)**, тип данных: NUMBER(3), обязательное поле;  **количество (Amount)**, тип данных: NUMBER(4,0), обязательное поле; значение по умолчанию: 1  **общая сумма заказа (Total)**, тип данных: NUMBER(6,0), обязательное поле; |
| 5. | 1. **Первая таблица PARTICIPANTS**– таблица для хранения данных об участниках научной конференции.  **Поля**:  **идентификатор участника** (**ID**), тип данных:  NUMBER(4, 0), первичный ключ; допустимые значения: > 1000  **ФИО (Fname)**, тип данных: VARCHAR(100), обязательное поле;  **академическая степень (AcademicDegree)**, тип данных: VARCHAR(50); допустимые значения: Bachelor, Master, Ph.D, Sc.D, Non-Degree; значение по умолчанию: Non-Degree;  **место работы (PlaceOfWork)**, тип данных: VARCHAR(100); обязательно поле;  **должность (Position),** тип данных: VARCHAR(30), обязательно поле;  **гражданство (Citizenship)**, тип данных: VARCHAR(30), обязательно поле;  **дата рождения (Bdate)** , тип данных: DATE; допустимые значения: < ’01.01.1996’  2. Вторая таблица **Papers –** таблица с данными о докладах участников.  **Поля**:  **идентификатор доклада** (**ID**), тип данных:  NUMBER(3, 0);  **название доклада (Title)**, тип данных: VARCHAR(30), обязательно поле;  **идентификатор участника** (**ID**), тип данных:  NUMBER(4, 0), внешний ключ для связи с таблицей **PARTICIPANTS**  **тип доклада (Type),** тип данных: VARCHAR(1), допустимые значения: P (постерный), O (устный); значение по умолчанию: O.  **название секции (SectionName)**, тип данных: NUMBER(2), обязательное поле;  **дата выступления (Sdate)**, тип данных: DATE, обязательное поле |