

Отчет по лабораторной работе № 2 «Основные компоненты СУБД Oracle и создание БД»			
дата	Оценка (max 5)	Бонус за сложность	подпись

Цели работы:

Изучение основных компонентов и инструментария СУБД Oracle, создание тестовой базы данных и ее основных элементов

Задачи работы:

- Установка СУБД и изучение базового инструментария для работы с СУБД Oracle
- Создание базы данных
- Создание основных элементов БД (таблиц, ограничений и т.п.)

Задание повышенной сложности (бонус за сложность – 10 баллов):

- создание тестовой базы данных на СУБД Oracle под LINUX

Краткий конспект теоретической части (ответы на контрольные вопросы)

Основной инструментарий для работы с СУБД Oracle _____

DDL и DML _____

Создание пользователей _____

Управление привилегиями и ролями _____

Перед тем, как приступить к разработке базы данных для ИИС сформулируем основные общие требования к принципам разработки баз данных.

1. База должна состоять из модулей (совокупности объектов), каждый из которых по возможности должен являться независимой единицей и должен быть способен работать максимально автономно от других модулей.
2. Взаимодействие между модулями должно осуществляться с помощью хранимых процедур и представлений, что имеет целью сделать возможным изменение структуры базы модуля без переделки других модулей, которые с ним взаимодействуют. Запрещается прямое обращение к таблицам другого модуля. При таком подходе при изменении структуры базы (таблиц) в модуле для сохранения работоспособности других модулей будет достаточно переделать соответствующие представления и процедуры.
3. Взаимодействие клиентской части и серверной рекомендуется осуществляться с помощью хранимых процедур и представлений.
4. База данных должна создаваться под единым **ORACLE-пользователем** (владельцем объектов). Конечные пользователи должны обращаться к процедурам, функциям, представлениям владельца объектов через общие синонимы (**PUBLIC SYNONYM**).
5. Конечные пользователи системы не имеют привилегий на действия с объектами БД.
6. Все ORACLE-права должны раздаваться через роли, которые создаются администратором по согласованию с разработчиком. Разработчики должны включать привилегии на создаваемые объекты в ранее созданные роли.
7. С целью обеспечения уникальности и мнемоничности имен объектов базы необходимо придерживаться

СОГЛАШЕНИЯ (Префиксное наименование таблиц) о наименовании объектов:

- Имена таблиц и хранимых процедур модуля должны начинаться с какого-либо определенного префикса (2-4 буквы), за которым следует знак подчеркивания «_». Префикс должен быть одним и тем же для всех таблиц и процедур одного модуля.
- Имена создаваемых общих синонимов должны совпадать с именами самих объектов.
- Имена последовательностей (SEQUENCE) должны начинаться с символов «S_», за которыми следует префикс модуля с последующим подчеркиванием, после чего идет смысловое имя индекса. При создании последовательности для первичного ключа таблицы рекомендуется такое имя последовательности: **S ИМЯ ТАБЛИЦЫ**.
- Имена представлений (View) должны начинаться с символов «V_», далее префикс модуля и смысловая часть (например: имя основной таблицы).
- Имена индексов (INDEX) должны начинаться с символов «I_», далее имя таблицы (или его сокращение), после чего следует смысловая часть. Например: для первичных ключей рекомендуется применять аббревиатуру «PK», для FOREIGN KEY - имена столбцов или сокращений, на которые осуществляется ссылка.
- Имена ограничений на таблицы (CONSTRAINT) должны начинаться с символов «C_» (ограничения типа PRIMARY KEY должны начинаться с символов «I_», т.к. в соответствии с этим именем создается индекс), далее аналогично именам последовательностей.
- Имена триггеров (TRIGGER) должны начинаться с символов «T_», далее аналогично именам последовательностей.
- Имена связей (DB LINK) должны начинаться с символов «L_», далее аналогично именам последовательностей.

Требования к именам объектов базы данных

- должны иметь длину от 1 до 30 байт, за исключением имен баз данных, длина которых ограничена 8 байтами;
- не могут содержать кавычек;
- не могут совпадать с именами других объектов.

Имена, которые всегда заключены в двойные кавычки, могут нарушать, приведенные ниже правила. В противном случае, имена

- должны начинаться с букв A-Z;
- могут содержать только символы A-Z, 0-9, _, \$ и #;
- не могут дублировать зарезервированные слова SQL.

Различие между прописными и строчными буквами учитывается только в именах, заключенных в двойные кавычки.

Рекомендации:

1 Рекомендуется, чтобы поля таблиц имели префиксы (для каждой таблицы свой префикс).

2 Все создаваемые ограничения (CONSTRAINT) должны быть поименованы. Например:

```
CREATE TABLE DAMN_TABLE  
(A CHAR CONSTRAINT I DAMN_TABLE PK PRIMARY KEY  
USING INDEX TABLESPACE  
&&index_tablespace);
```

а не:

```
CREATE TABLE DAMN_TABLE  
(A CHAR PRIMARY KEY);
```

т.к. во втором варианте название индекса генерируется автоматически и не связано с именем таблицы.

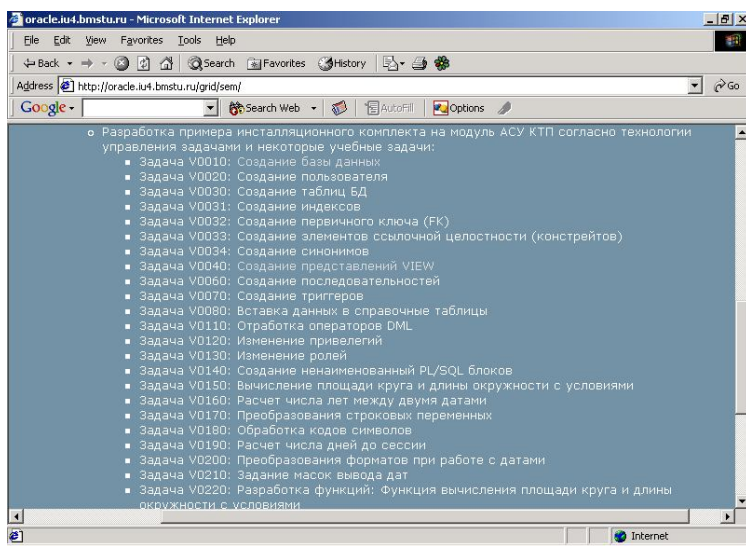
3 Создание общих синонимов должно прописываться в тех же SQL-программах, что и создание объектов.

4 Требуется писать комментарии в тексте хранимых процедур и комментировать таблицы, представления и их поля с помощью SQL команды COMMENT.

5 Необходимо использовать существующую систему разграничения доступа или, по крайней мере, предусмотреть в коде представлений и хранимых процедур места, куда в будущем можно было бы вставить процедуры системы разграничения доступа. Эти процедуры отвечают на вопрос «имеет ли текущий пользователь право сделать данное действие над данным объектом», а представления показывают совокупность разрешенных действий над объектами для текущего пользователя.

- 6 Хранимые процедуры не должны находиться в состоянии ожидания освобождения заблокированных другими пользователями записей длительное время. Для этого перед попыткой захвата данных необходимо убедиться в отсутствии чужих блокировок (SELECT ... FOR UPDATE ...NOWAIT), и в случае наличия блокировок пользователю должно быть выдано соответствующее сообщение.
- 7 При написании хранимых процедур нужно считать, что на Delphi-клиенте выбран режим работы SHARED NOAUTOCOMMIT.
- 8 Ошибки хранимых процедур, как логические так и подключения к ORACLE, должны обрабатываться в самих процедурах с выполнением ROLLBACK, по крайней мере, до точки вызова процедуры. При этом к клиенту не должен возвращаться текст сообщения ORACLE SERVER об ошибке. Вместо этого клиенту должен возвращаться код ошибки и подробное описание ошибки, созданное разработчиком.
- 9 В начале процедуры должны быть комментарии, отражающие следующие сведения:
- наименование подсистемы;
 - наименование модуля;
 - версия и дата последнего обновления;
 - фамилия разработчика;
 - краткое назначение;
 - комментарий к каждому входному и выходному параметру;
 - производит ли процедура фиксацию транзакций во время своей работы;
 - до какой точки производится откат транзакций в случае ошибки;
 - оставляет ли процедура заблокированные записи в таблицах после выполнения своей работы.

ТРЕБОВАНИЕ К ОФОРМЛЕНИЮ SQL-ПРОГРАММ ПРИ ПРОВЕДЕНИИ ИЗМЕНЕНИЙ В БАЗЕ ДАННЫХ



Разрабатываемые скрипты для создания объектов базы данных и внесения изменений рекомендуется оформлять в виде отдельных sql программ (скриптов), каждый из которых (либо группа скриптов) размещается в отдельном каталоге на диске с именем, например, V0001 – где цифра обозначает номер текущей задачи для рассматриваемого набора скриптов. В каталоге находится файл index.htm (с описанием сути задачи), файл start.sql (в котором прописана очередность запуска скриптов) и файлы самих скриптов. Все файлы index.htm связаны в единый перечень задач корневым файлом inex.htm, образуя тем самым гипертекстовую справочную систему по всему созданному в проекте инсталляционному комплекту (рис)

Необходимо предусмотреть возможность запуска sql-файлов в назначенной последовательности несколько раз.

При выполнении sql-программы должен включаться вывод результатов в файл. Для этого необходимо добавить:

- в начало файла:

SPOOL xxxxxxxx.lst

где xxxxxxxx - имя файла sql-программы;

- в конец файла:

SPOOL off.

Необходимо, чтобы в sql-программе при выполнении действий с объектом базы данных отражалось имя этого объекта, например:

PROMPT создается TABLE1

CREATE TABLE1 ...

Следует включить в команду CREATE INDEX опцию TABLESPACE и предусмотреть возможность интерактивной подстановки имени табличного пространства администратору базы данных в момент выполнения sql-программы:

CREATE INDEX i_table1 ON table1 ...TABLESPACE &&tablespace;

Перед командой создания синонимов должна ставиться команда удаления синонимов с тем же именем:

DROP PUBLIC SYNONYM xxx. ...

CREATE PUBLIC SYNONYM xxx ...

1 Изучение инструментария для работы с СУБД Oracle

Инструментарий:

- проектирования и моделирования - Oracle Designer;
- разработки экранных форм - Oracle Forms Developer;
- разработки отчетных форм - Oracle Reports Developer;
- быстрой разработки приложений на языке Java - Oracle JDeveloper;
- разработки и внедрения корпоративных хранилищ данных и интеллектуальных приложений электронного бизнеса - Oracle Warehouse Builder;
- выполнения DDL- и DML-запросов к базе данных - SQL*Plus.

Для расширения информационной системы предприятия за счет дополнительных аналитических средств поиска, анализа и моделирования предназначены семейства продуктов Oracle Discoverer (ROLAP) и Oracle Express (MOLAP).

Для обеспечения функций интерактивного управления Oracle Enterprise Grids — сетями распределенных вычислений предприятия предназначен специальный модуль - Oracle Enterprise Manager 10g. Этот модуль значительно упрощает управление бизнес-приложениями при одновременном сокращении затрат на их поддержку и обновление. Oracle Enterprise Manager 10g представляет собой управляющее ПО на основе тонкого клиента, которое дает полную картину вычислительной инфраструктуры предприятия. Системные администраторы получают возможность вносить изменения в информационные политики, уровни сервисного обслуживания, а также в распределение существующих вычислительных ресурсов и приложений в зависимости от изменяющихся потребностей бизнеса, обеспечивая высочайшее качество обслуживания.

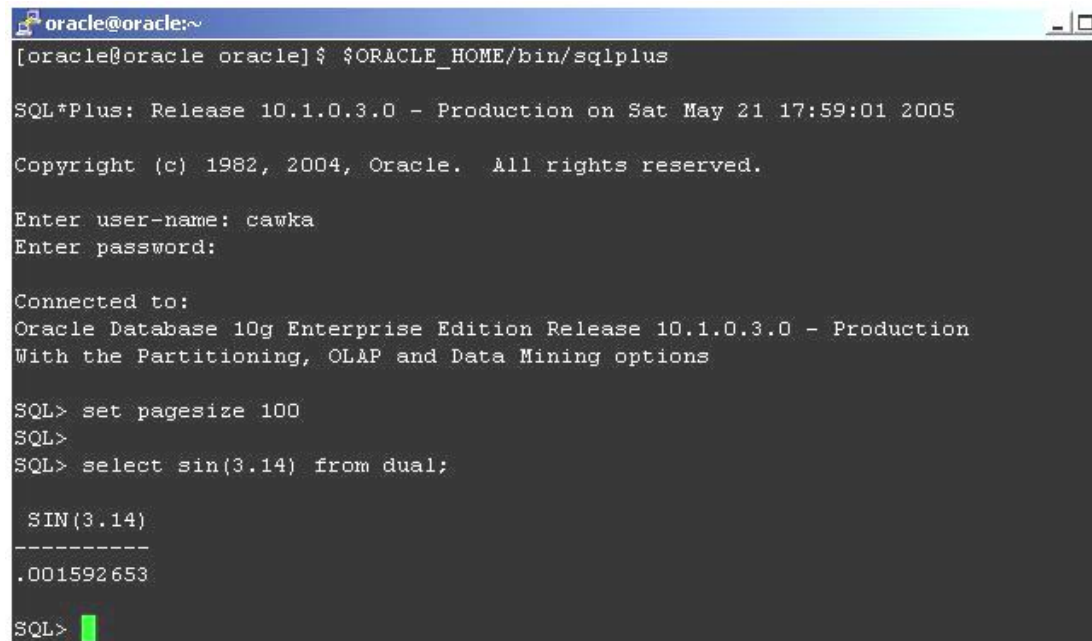
1.1. Использование SQL*Plus

Одним из основных инструментов по работе с СУБД Oracle является **SQL*Plus**, позволяющий задавать текст команд (скриптов) на языке SQL и выполнять их в интерактивном режиме. Приложение принимает от пользователя инструкции для доступа к базе данных и направляет их серверу Oracle, а результаты отображает на экране монитора. Кроме того среда SQL*Plus имеет набор специальных команд, используемых для:

- форматирования результатов, возвращаемых сервером
 - установки опций настройки среды SQL*Plus
 - редактирования и хранения SQL-команд.
- Приложение SQL*Plus можно запустить разными способами.
- **Как консольную программу** Программа выполняется из оболочки или командной строки (окружения, которое иногда называют консолью).
 - **В качестве программы с псевдографическим пользовательским интерфейсом.** Эта разновидность SQL*Plus доступна только в Microsoft Windows. Ее называют псевдографическим интерфейсом, поскольку она очень похожа на консольную программу, отличаясь от таковой, кроме прочего, наличием растровых шрифтов.
 - **Через iSQL*Plus (в Oracle9: или более поздней версии).** Программа выполняется из web-браузера машины, на которой работает HTTP-сервер Oracle и сервер iSQL*Plus.
 - **Через SQL*Plus Worksheet.** Графический пользовательский интерфейс Java с консольной версией SQL*Plus. Хотя он поддерживает небольшой журнал инструкций, ничего примечательного в этой версии среды разработки нет.

Запуск SQL*Plus в консольном режиме:

Консольный режим работы обеспечивает доступ к серверу, на котором установлена СУБД в консольном режиме (например, по telnet). На рисунке 1.1. показано обращение к серверу oracle.iu4.bmstu.ru по протоколу telnet, подключение к серверу и вызов sql+ из командной строки.



```
oracle@oracle:~  
[oracle@oracle oracle]$ $ORACLE_HOME/bin/sqlplus  
  
SQL*Plus: Release 10.1.0.3.0 - Production on Sat May 21 17:59:01 2005  
  
Copyright (c) 1982, 2004, Oracle. All rights reserved.  
  
Enter user-name: cawka  
Enter password:  
  
Connected to:  
Oracle Database 10g Enterprise Edition Release 10.1.0.3.0 - Production  
With the Partitioning, OLAP and Data Mining options  
  
SQL> set pagesize 100  
SQL>  
SQL> select sin(3.14) from dual;  
  
      SIN(3.14)  
-----  
      .001592653  
  
SQL>
```

Рис.1.1. Работа с SQL+ в командной строке.

Для запуска консольной версии SQL*Plus, нужно ввести команду *sqlplus* в ответ на приглашение операционной системы. Далее следуют текст, а также строки для ввода имени пользователя и пароля (например, scott, tiger – для тестовой схемы):

```
D:\>sqlplus  
  
SQL*Plus: Release 9.0.1.0.1 - Production on Sun Jan 23 14:23:03 2005  
  
<C> Copyright 2001 Oracle Corporation. All rights reserved.  
  
Enter user-name: scott  
Enter password:  
  
Connected to:  
Oracle9i Enterprise Edition Release 9.0.1.1.1 - Production  
With the Partitioning option  
JServer Release 9.0.1.1.1 - Production  
  
SQL> _
```

Можно также загрузить SQL*Plus, сразу задав имя пользователя и пароль в командной строке:

```
D:\>sqlplus scott/tiger
```

С помощью ключа */NOLOG* можно запустить SQL*Plus без подключения к базе данных, указав имя пользователя и пароль посредством команды *CONNECT*:

```
D:\>sqlplus /nolog  
  
SQL*Plus: Release 9.0.1.0.1 - Production on Sun Jan 23 14:36:08 2005  
  
<C> Copyright 2001 Oracle Corporation. All rights reserved.  
  
SQL> CONNECT scott/tiger  
Connected.  
SQL> _
```

Для подключения к удаленным базам данных (то есть серверам баз данных, работающих на других компьютерах) наряду с именем пользователя и паролем нужно знать *идентификатор подключения* Oracle Net (SID), называемый также *именем сервиса*. Идентификатор вводится после имени пользователя и пароля, отделяясь от них символом «@»:

```
SQL> CONNECT scott/tiger@ORADB
Connected.
SQL> _
```

Исполняемый файл для запуска SQL+ находится в подкаталоге bin домашней директории Oracle и называется sqlplus (sqlplus.exe для Windows 1).

i SQL * Plus:

iSQL*Plus представляет собой модную ныне Web-реализацию SQL*Plus. Продукт работает в трехзвенной архитектуре: Web браузер - iSQL*Plus - База Данных. Пользовательский интерфейс выполнен при помощи HTML+JScript. Впервые продукт появился вместе с Oracle 9.0.1 на платформе Windows. С версией 9.2 он доступен и на *nix системах. Между ним и консольным SQL*Plus есть существенные различия. Например, он не поддерживает следующие команды: EXIT, GET, HOST, SPOOL, WHENEVER... и это далеко не полный перечень.

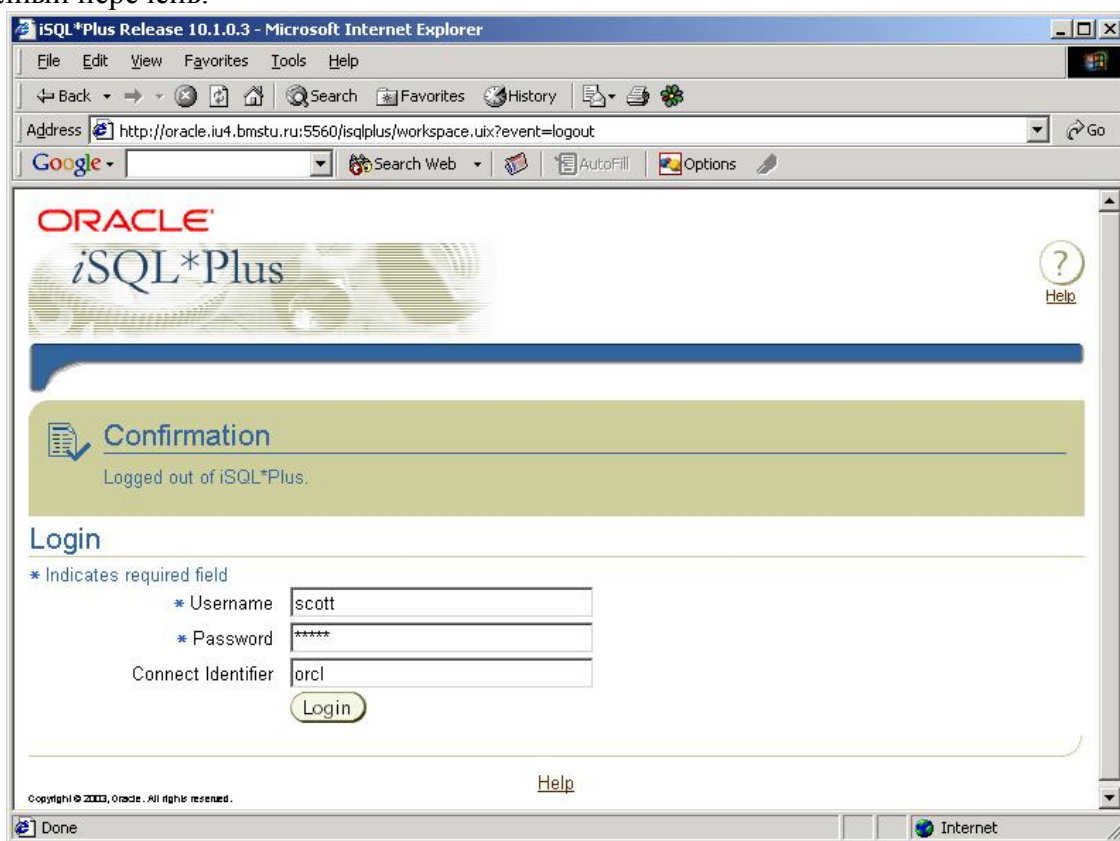


Рис.1.2. Подключение к iSQL*Plus.



Задание: Подключиться к iSQL*Plus на тестовом сервере и выполнить простейший скрипт:

Адрес тестового сервера в интернете: <http://oracle.iu4.bmstu.ru:5560/isqlplus>;
LOGIN: scott (доступна схема SCOTT, а также схемы, созданные Вами на занятиях);

PASSWORD: tiger (пароль пользователя SCOTT, либо Ваш); SID: ORCL

Пример SQL команды: Вычисление значения синуса

```
SELECT sin(3.14) FROM DUAL;
```

Посредством SELECT запроса обращаемся к встроенным тригонометрическим функциям Oracle, результат выводится в выходной поток. Специальная таблица DUAL используется для сохранения синтаксиса SELECT запроса.

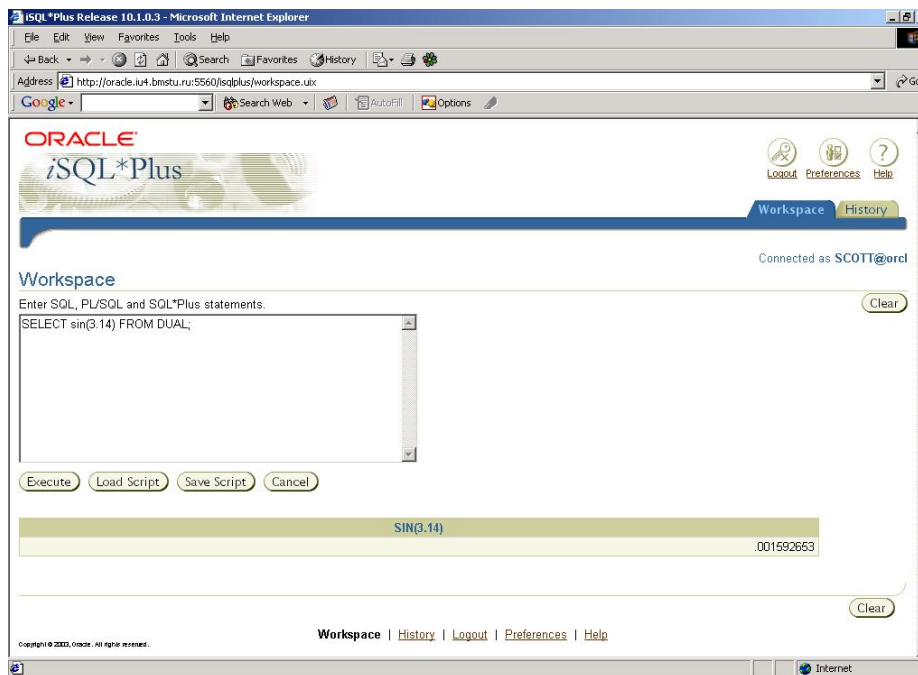


Рис. 1.3. Работа с iSQL*Plus.

Для запуска iSQL*Plus необходимо выполнить под юзером oracle \$ORACLE_HOME/bin/isqlplus start, после чего вы сможете обращаться к нему через WEB интерфейс <http://localhost:5560/isqlplus>. Кроме iSQL*Plus существует несколько различных его реализаций.

Графический SQL*Plus (Win):

Существует только на платформе Windows. Представляет собой такое же консольное приложение, но в графическом окне. По своим возможностям он практически идентичен консольному варианту. Исполняемый файл для запуска находится в подкаталоге bin домашней директории Oracle и называется sqlplusw (plus80w.exe для восьмой версии и plus73w.exe для 7.3).

Среди преимуществ этой реализации можно отметить ещё более простое использование выделения и вставки текста. Выделите мышью нужный фрагмент текста и, не отпуская левой кнопки мыши, нажмите на правую кнопку. Выделенный фрагмент будет вставлен в позицию курсора. Причем нужно отметить, что выделенный фрагмент не помещается в стандартный буфер Windows, который доступен через Ctrl-C/Ctrl-V, как и в любом графическом Windows приложении.

Ещё одним приятным и не слишком известным фактом, является возможность изменить название и размер используемого шрифта.

Для тех, кто использует эту реализацию SQL*Plus, не самой лучшей новостью станет известие о том, что в будущих версиях графического SQL*Plus больше не будет. Ему на смену идет iSQL*Plus (эту информацию можно найти в файле \$ORACLE_HOME92/sqlplus/doc/README.htm, поставляемым к SQL*Plus версии 9.2). Хотя никто и не запрещает поставить его и использовать совместно с Oracle 10g.

SQL*Plus Worksheet

SQL*Plus Worksheet является компонентом Enterprise Manager (до Oracle 9.x), хотя может вызываться и автономно. Как и у Enterprise Manager, пользовательский интерфейс SQL*Plus Worksheet написан на Java. Но это только пользовательский интерфейс. Для выполнения команд и скриптов используется обычный sqlplus в фоновом режиме.

Вообще говоря, это очень странный продукт. Единственное чем он выгодно отличается от рассмотренных ранее реализаций, так это наличием истории команд. Но при этом Worksheet не поддерживает: START, EXIT, GET, HOST, SPOOL, WHENEVER... и это далеко не полный перечень. Вероятно, нет смысла использовать сильно урезанную версию продукта, если доступна полнофункциональная версия (в наши планы не входит описание различных “hacks”, при помощи которых можно добиться от Worksheet полной функциональности SQL*Plus).

Забегая не много вперед, отметим, что вся информация по всем объектам Oracle хранится в таблицах системного словаря, обращаясь к которым мы можем получить ее. При этом каждому пользователю соответствует своя **схема** (имя которой совпадает с именем пользователя), в которой хранятся все объекты принадлежащие пользователю. Так, например, пользователю SCOTT соответствует схема SCOTT, а информация по всем объектам в этой схеме можно получить обратившись к таблице системного словаря USER_OBJECTS с правами пользователя SCOTT, но для начала надо узнать структуру этой таблицы, а для этого есть команда DESCRIBE (DESC).



Задание: Подключиться к iSQL*Plus на тестовом сервере и выполнить простейший скрипт:

Адрес тестового сервера в интернете: <http://oracle.iu4.bmstu.ru:5560/isqlplus>;

LOGIN: scott (доступна схема SCOTT, а также схемы, созданные Вами на занятиях);

PASSWORD: tiger (пароль пользователя SCOTT, либо Ваш);

SID: ORCL

Пример SQL команды: Получение информации о атрибутах таблицы.

DESC USER_OBJECTS;

Пример SQL команды: Вывести все объекты, принадлежащие схеме SCOTT

```
SELECT OBJECT_NAME
       ,OBJECT_TYPE
       ,CREATED
FROM USER_OBJECTS;
```

Name	Null?	Type
OBJECT_NAME		VARCHAR2(128)
SUBOBJECT_NAME		VARCHAR2(30)
OBJECT_ID		NUMBER
DATA_OBJECT_ID		NUMBER
OBJECT_TYPE		VARCHAR2(19)
CREATED		DATE
LAST_DDL_TIME		DATE
TIMESTAMP		VARCHAR2(19)
STATUS		VARCHAR2(7)

Рис.1.4. Результат выполнения команды DESC.

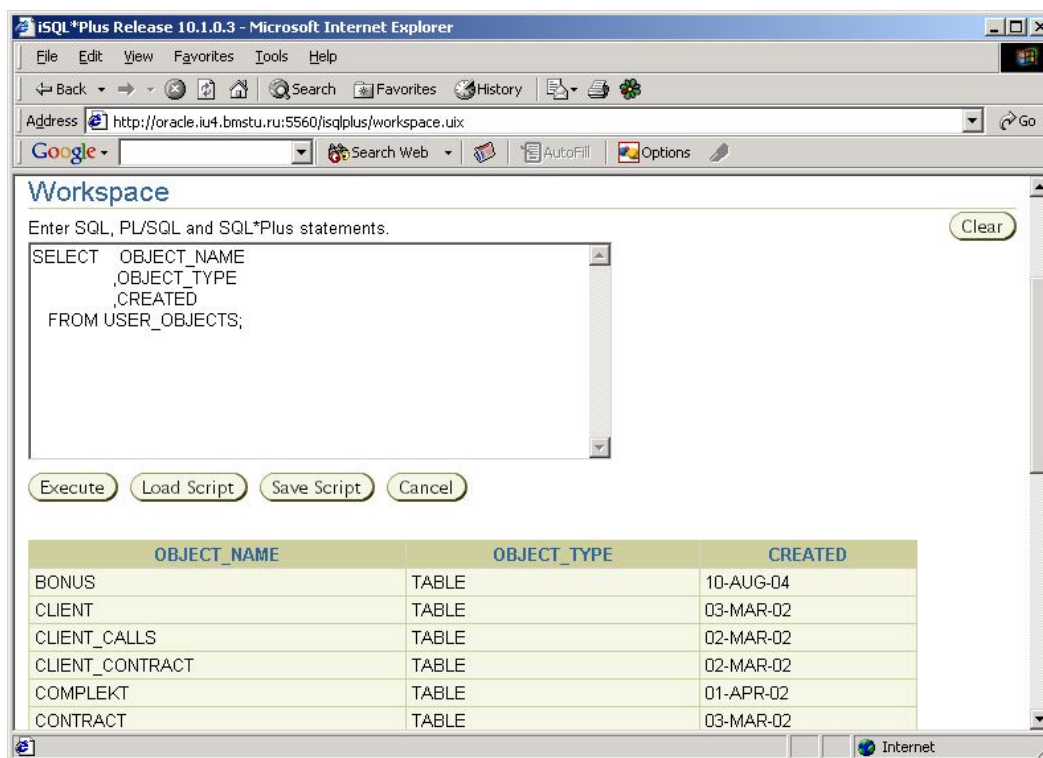


Рис. 1.5. Получение списка объектов принадлежащих схеме пользователя SCOTT.

Используя этот не хитрый набор команд можно в любой момент получить информации по всем объектам конкретной схемы.

1.2. Использование Oracle Enterprise Manager

Oracle Enterprise Manager 10g — модуль, предназначенный для управления Oracle Enterprise Grids — сетями распределенных вычислений предприятия. Этот модуль значительно упрощает управление бизнес-приложениями при одновременном сокращении затрат на их поддержку и обновление. Oracle Enterprise Manager 10g представляет собой управляющее ПО на основе тонкого клиента, которое дает полную картину вычислительной инфраструктуры предприятия. Системные администраторы получают возможность вносить изменения в информационные политики, уровни сервисного обслуживания, а также в распределение существующих вычислительных ресурсов и приложений в зависимости от изменяющихся потребностей бизнеса, обеспечивая высочайшее качество обслуживания. Открытость и расширяемость.

Oracle Enterprise Manager 10g создан на базе открытой архитектуры, основанной на стандартах Common Information Model (CIM) и Web-based Enterprise Management (WBEM), разработанные рабочей группой по управлению распределенными средами (Distributed Management Task Force, DMTF). Управление. Управление сетями распределенных вычислений является технологической дисциплиной, охватывающей вопросы, связанные с вычислительной архитектурой, персоналом, процессами, а также с накопленным опытом. Oracle Enterprise Manager 10g отлично справляется с этой задачей.

Oracle Enterprise Manager 10g — это компонент управления, который создан на базе накопленного корпорацией Oracle опыта в области управления центрами хранения и обработки данных; он помогает администраторам повысить производительность вычислительных систем и выйти на качественно новый уровень автоматизации и анализа. Oracle Enterprise Manager 10g — инструмент, который увеличит отдачу от использования решений Oracle для сетей распределенных вычислений.



Задание: Подключиться к ЕМ на тестовом сервере:

Адрес тестового сервера в интернете: <http://oracle.iu4.bmstu.ru:5500/em>;

LOGIN: SYSTEM (при первом обращении после установки, либо логин с ролью dba, сообщенный Вам на занятиях);

PASSWORD: MANAGER (пароль пользователя SYSTEM); SID: ORCL

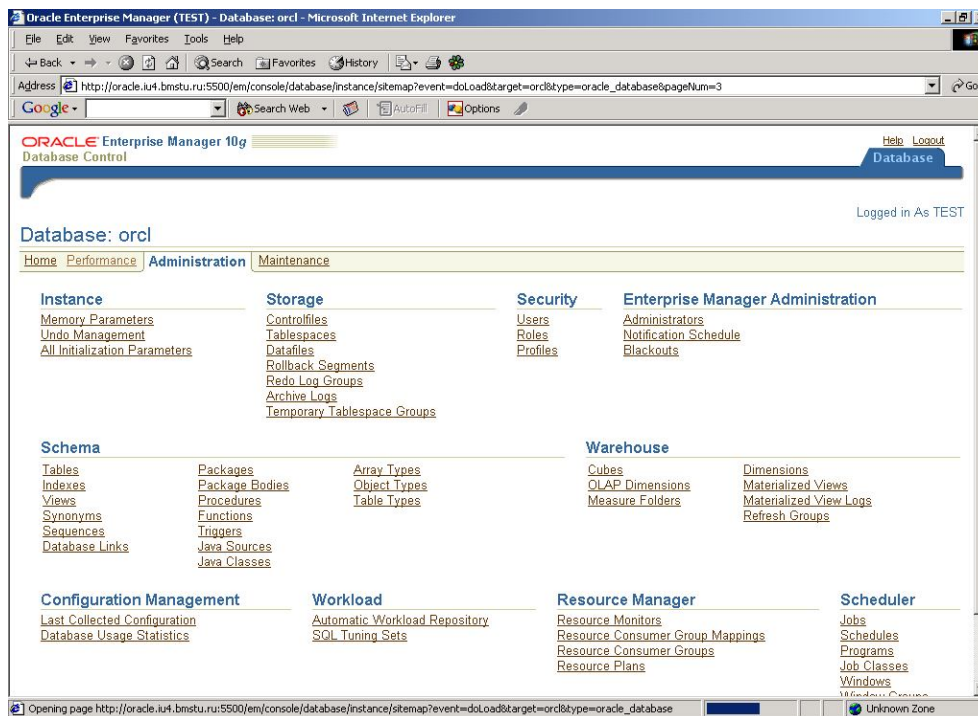


Рис.1.6. Вид WEB интерфейса

В случае перезагрузки системы, если инициализация сервисов не настроена автоматически ЕМ инициализируется следующей последовательностью команд (ОС LINUX):

```
#заходим в систему под специальным пользователем oracle
su - oracle -c "$ORACLE_HOME/bin/emctl start dbconsole"
#определяем переменные
ORACLE_HOME=/home/oracle/product/10.1.0/Db_1/
export ORACLE_HOME
ORACLE_SID=orcl
export ORACLE_SID
#Запускаем СУБД Oracle
#Стартуем базу
su - oracle -c "$ORACLE_HOME/bin/dbstart"
#Запускаем sqlplus
su - oracle -c "$ORACLE_HOME/bin/sqlplusctl start"
#Запускаем EM
su - oracle -c "$ORACLE_HOME/bin/emctl start dbconsole"
#Запускаем TNS Listener
su - oracle -c "$ORACLE_HOME/bin/lsnrctl start"
```

После этого мы можем обращаться к ЕМ по адресу <http://host:5500/em>;

Участники программы Oracle Technology Network (OTN) смогут бесплатно загрузить новое управляющее ПО, посетив сайт <http://otn.oracle.com>.

Вклейте отчет с конфигурацией созданного вами тестового сервера

2. ИЗУЧЕНИЕ ОПЕРАТОРОВ СОЗДАНИЯ И ИЗМЕНЕНИЯ ОБЪЕКТОВ БД

Перед тем как работать с данными БД ее надо создать, для этих целей используется специальная группа операторов, предназначенных для создания объектов базы данных, все операторы данной группы начинаются с ключевого слова **CREATE**:

- **CREATE DATABASE** - СОЗДАНИЕ БАЗЫ ДАННЫХ
- **CREATE USER** - СОЗДАНИЕ ПОЛЬЗОВАТЕЛЯ
- **CREATE TABLE** - СОЗДАНИЕ ТАБЛИЦ,
- **CREATE SYNONYM** - СОЗДАНИЕ СИНОНИМОВ,
- **CREATE INDEX** - СОЗДАНИЕ ИНДЕКСОВ,
- **CREATE VIEW** - СОЗДАНИЕ ПРЕДСТАВЛЕНИЙ.
- **CREATE SEQUENCE** - СОЗДАНИЕ ПОСЛЕДОВАТЕЛЬНОСТЕЙ

и т.д.

Для внесения изменений в уже созданные объекты используются операторы группы **ALTER**, которые начинаются с ключевого слова **ALTER**. Будем рассматривать операторы групп **CREATE** и **ALTER** попарно, применительно к каждому из рассматриваемых объектов БД.

Синтаксическая диаграмма – это графическое представление правил построения конструкций языка в наглядной форме. При этом принято, символы алфавита изображать блоками в овальных рамках, названия конструкций – в прямоугольных, а правила построения конструкций в виде линий со стрелками (часто овальные и прямоугольные рамки для упрощения опускают). Также выделяют осевую линию синтаксической конструкции, на которой размещаются все обязательные символы алфавита и конструктивные элементы. Все необязательные элементы и символы размещаются ниже или выше осевой линии конструкции. Если конструкция имеет значительную длину, то осевая линия переносится соответственно на следующую строку. Существуют и альтернативные формы отображения синтаксиса языка, например форма Бэкуса-Наура (БНФ) и т.п. [2].

Рассмотрим суть использования синтаксической диаграммы для представления синтаксиса языка HTML на простейшем примере. Так, например, стандарт определяет, что документ должен начинаться тэгом **<html>** и заканчиваться тэгом **</html>**. Сам документ, с всевозможными включениями (например, JavaScript или VBScript), располагается внутри этих тэгов. Синтаксическая диаграмма, описывающая эту конструкцию представлена на рис.2.1.

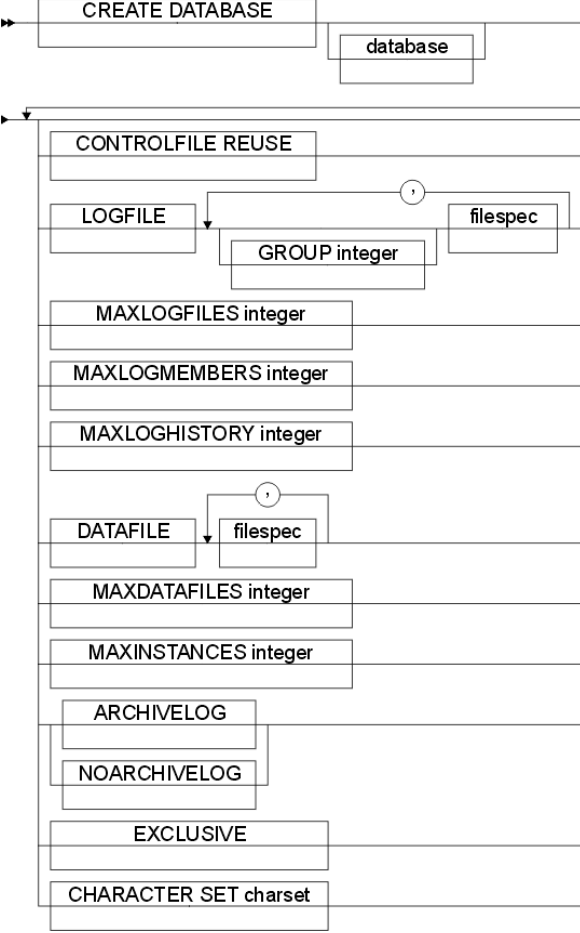



Рис.2.1 Синтаксическая диаграмма конструкции HTML документа.

В дальнейшем обозначение символов алфавита в овальных рамках будем опускать.

CREATE/ALTER DATABASE - СОЗДАНИЕ БАЗЫ ДАННЫХ

Задаёт и определяет максимальное число экземпляров файлов данных и журнальных файлов, устанавливает режим архивирования.

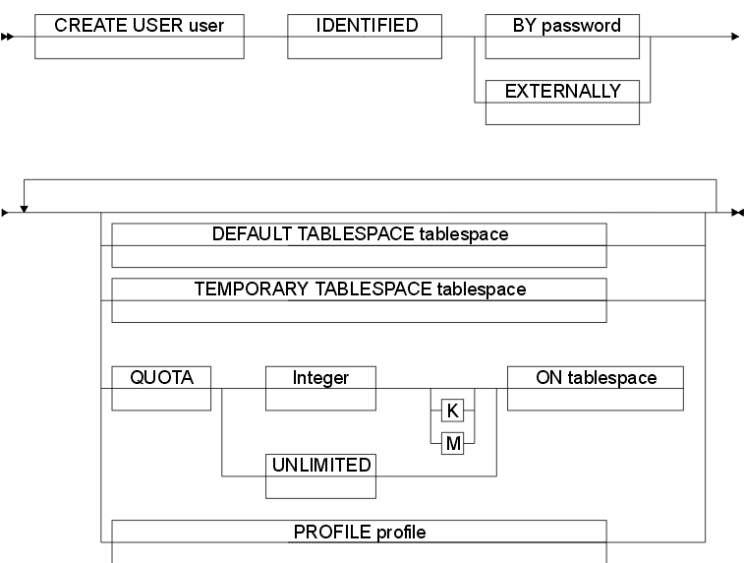

Синтаксическая диаграмма	Тело скрипта
<div></div>	<div><p>Пример:</p><pre>PROMPT Создаем БД CLINICS SPOOL clinic.log CONNECT internal STARTUP NOMOUNT pfile=/oracle/dbs/initclinic.ora CREATE DATABASE "clinics" MAXINSTANCES 1 MAXLOGFILES 10 CHARACTER SET "RU8PC866" NATIONAL CHARACTER SET "RU8PC866" DATAFILE '/oracle/db/system01.dbf' size 100M LOGFILE '/oracle/db/lo g01.dbf' size 1M, '/oracle/db/log02.dbf' size 1M; PROMPT Создаем БД CLINICS; SPOOL clinic.log; ✓ CREATE DATABASE "clinics" MAXINSTANCES 1 MAXLOGFILES 10 CHARACTER SET "RU8PC866" NATIONAL CHARACTER SET "RU8PC866" DATAFILE '/opt/oracle/oradata/CLINICS/system01.dbf' SIZE 100M LOGFILE '/opt/oracle/oradata/CLINICS/log01.dbf' SIZE 1M ,'/opt/oracle/oradata/CLINICS/log02.dbf' SIZE 1M; SPOOL OFF;</pre></div> <div></div> <div>Вклеить текст скрипта создания своей БД</div>

Вклеить HTML отчет с конфигурацией созданной тестовой базы данных

3 УПРАВЛЕНИЕ ПОЛЬЗОВАТЕЛЯМИ. ПРИВИЛЕГИИ И РОЛИ

CREATE/ALTER USER - СОЗДАНИЕ И ИЗМЕНЕНИЯ ПОЛЬЗОВАТЕЛЯ

Задания по управлению профилями пользователей выполняются на тестовом сервере с права DBA.

Синтаксическая диаграмма	Тело скрипта
	<p>Пример: PROMPT Скрипт создания пользователя test</p> <pre>CREATE USER "test" PROFILE "DEFAULT" IDENTIFIED BY "123456" DEFAULT TABLESPACE "USERS" QUOTA UNLIMITED ON CWM Lite QUOTA UNLIMITED ON DRSYS QUOTA UNLIMITED ON EXAMPLE QUOTA UNLIMITED ON INDX QUOTA UNLIMITED ON SYSTEM QUOTA UNLIMITED ON TEMP QUOTA UNLIMITED ON TOOLS QUOTA UNLIMITED ON USERS ACCOUNT UNLOCK; GRANT "CONNECT" TO "test"; GRANT "DBA" TO "test";</pre>
	Вклеить текст собственного скрипта

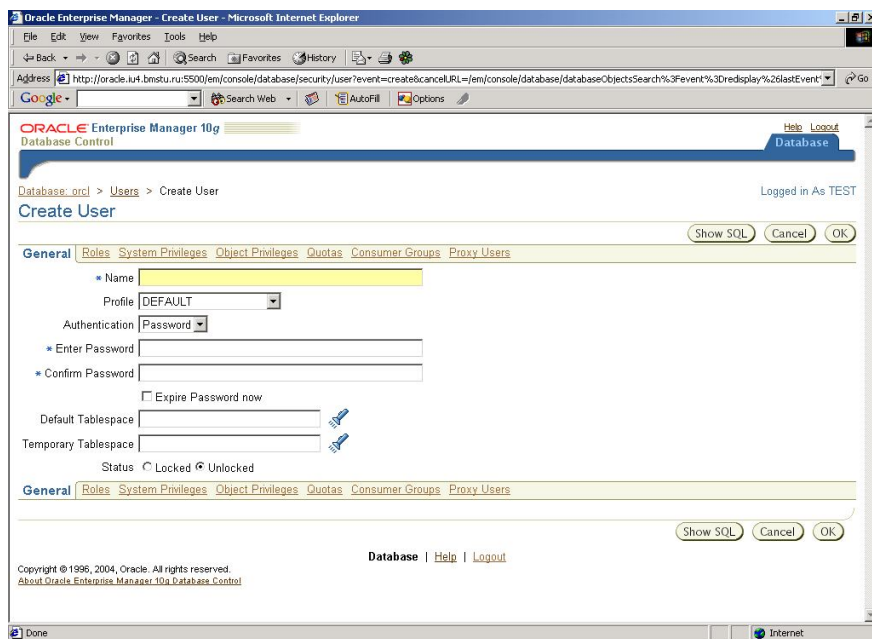


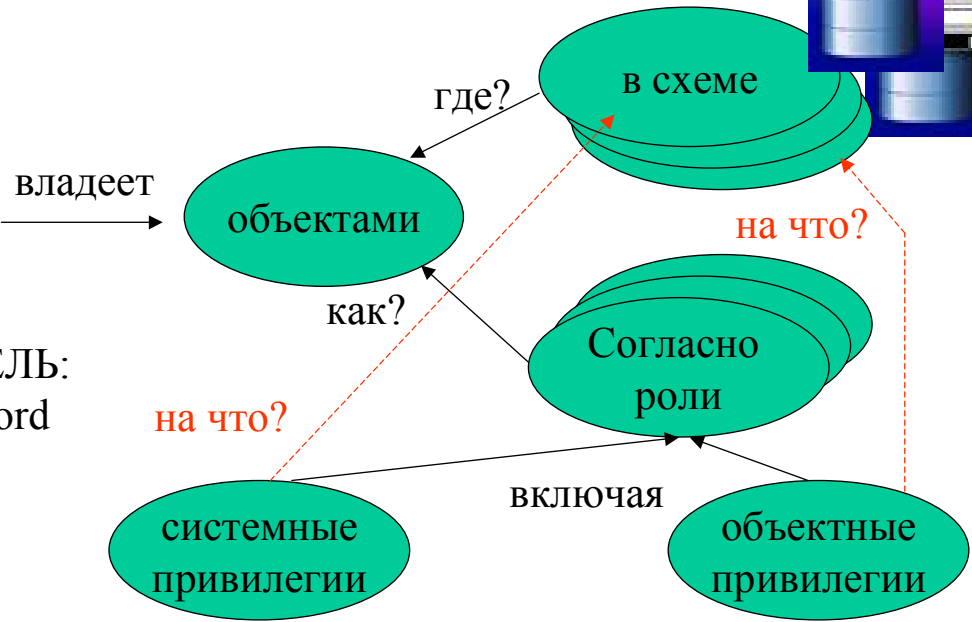
Рис.2.2. Создание пользователя через интерфейс EM.

! При первом подключении к установленной СУБД рекомендуется также сменить все пароли пользователей SYS и SYSTEM, если вы не сделали это раньше.

СИСТЕМНЫЕ ПРИВЕЛЕГИИ И РОЛИ



ПОЛЬЗОВАТЕЛЬ:
login & password



Вместо того, чтобы назначать привилегии непосредственно пользователям, необходимо назначать привилегии ролям, которые затем назначаются пользователям.



Задание 1:
Вывести, используя SQL запрос, отчет по ролям и привилегиям пользователя “SCOTT”, пользователя “HR” и своей учетной записи.

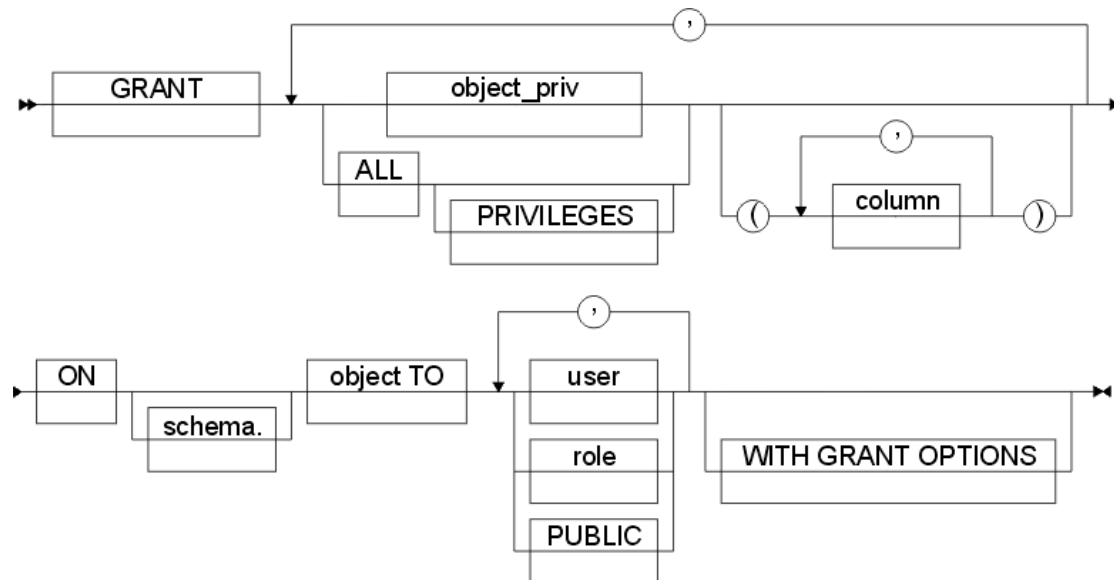
```
SQL> SELECT
2      USERNAME
3      , GRANTED_ROLE
4  FROM    USER_ROLE_PRIVS; (USER_SYS_PRIVS)
```

Для пользователя SCOTT	Для пользователя HR	Для своей учетной записи

GRANT (привилегии доступа к объектам)

Предоставляет привилегии доступа к определенным объектам (таблицам, представлениям, синонимам, пакетам, процедурам и т.д.) пользователям и ролям.

Установка объектных привилегий



Задание 2:

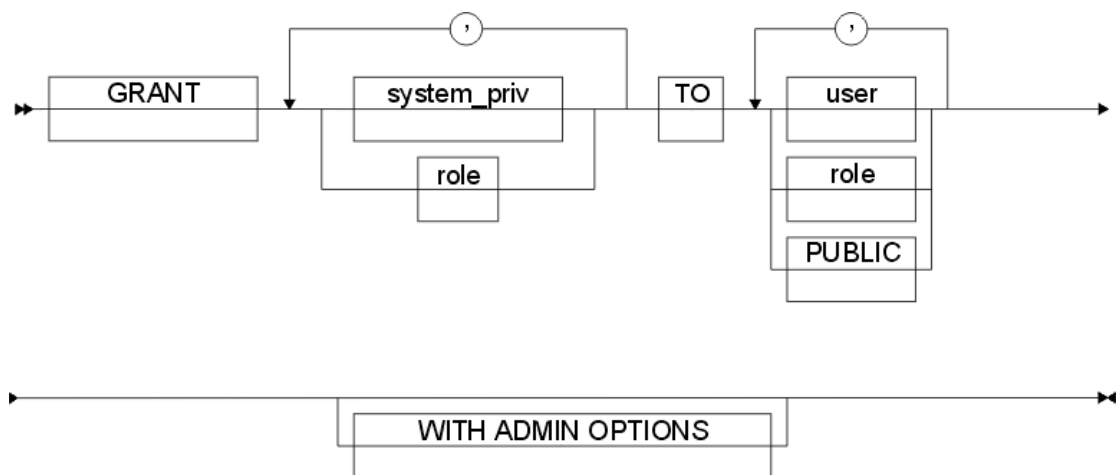
Предоставить пользователю SCOTT объектные привилегии на объекты вашей схемы (объекты и виды привилегий выбрать самостоятельно)

```
GRANT SELECT ON hr_jobs TO kruglovvs;
```

Результат обращения к объекту Вашей схемы пользователем SCOTT до предоставления привилегии

Результат обращения к объекту Вашей схемы пользователем SCOTT после предоставления привилегии

Установка системных привилегий



Задание 3:

Предоставить вашей учетной записи системные привилегии на работу с триггерами, функциями и процедурами.

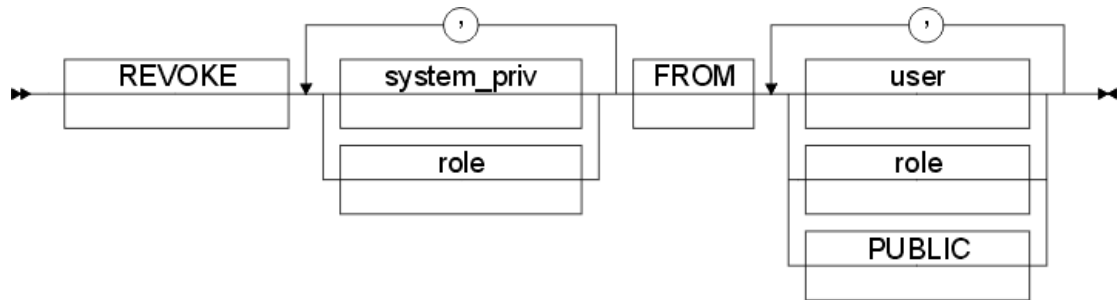
GRANT

```
ALTER ANY PROCEDURE
, ALTER ANY TRIGGER
, CREATE ANY PROCEDURE
, CREATE ANY TRIGGER
, DROP ANY PROCEDURE
, DROP ANY TRIGGER
, EXECUTE ANY PROCEDURE
```

TO "kruglovvs";

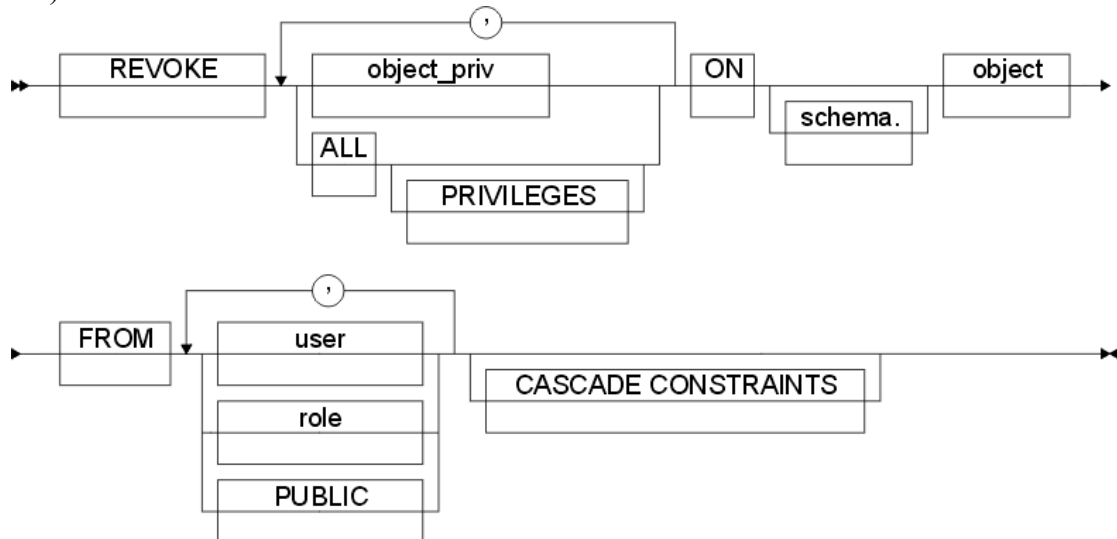
REVOKE (системные привилегии и роли)

Отменяет системные привилегии и роли, ранее предоставленные пользователям и ролям. Действие, обратное команде GRANT (системные привилегии и роли).



REVOKE (привилегии доступа к объектам)

Отменяет привилегии доступа к определенному объекту, ранее предоставленные пользователям и ролям. Действие, обратное команде GRANT (привилегии доступа к объектам).

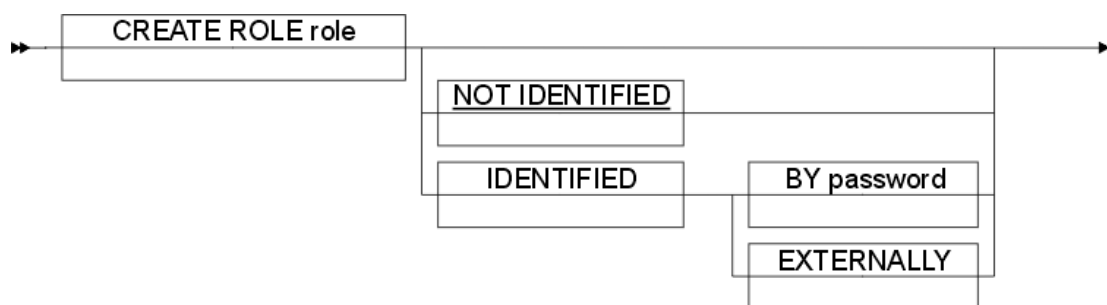


Задание 4:

Отменить действия, реализованные в примере 2 и 3.

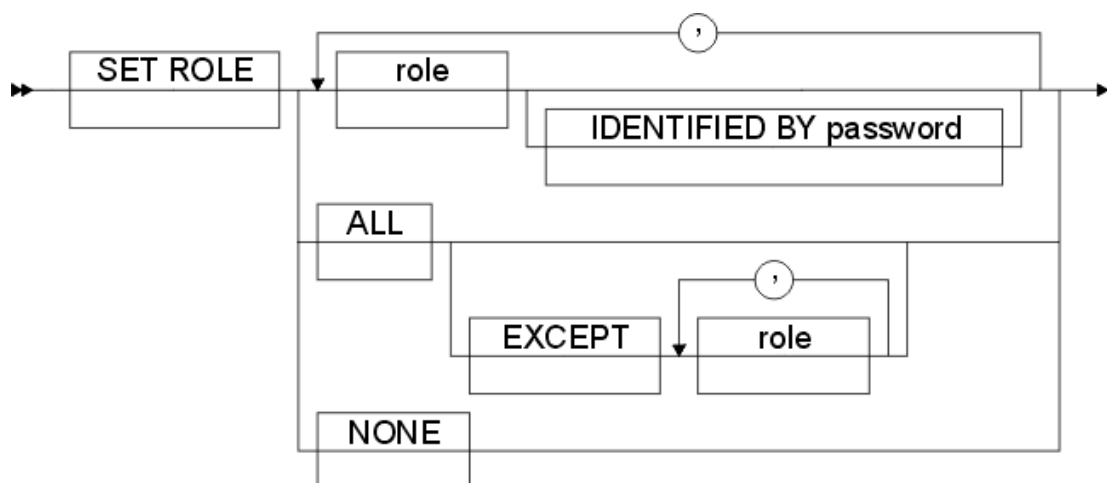
```
REVOKE
ALTER ANY PROCEDURE
,ALTER ANY TRIGGER
,CREATE ANY PROCEDURE
,CREATE ANY TRIGGER
,DROP ANY PROCEDURE
,DROP ANY TRIGGER
,EXECUTE ANY PROCEDURE
FROM "kruglovvs";
```

CREATE ROLE - Создает роль



SET ROLE - управление сеансом

Разрешает заданную роль в текущем сеансе и запрещает все другие роли пользователя. Должна выполняться в начале транзакции вместе с оператором `SET TRANSACTION`.



Задание 5:

Создать и назначить роль для своей учетной записи (набор объектных и системных привилегий выбрать самостоятельно).

```
CREATE ROLE "table_creator" NOT IDENTIFIED;
GRANT
| CREATE ANY TABLE
TO "table_creator";
GRANT "table_creator" TO "kruglovvs";
```

Контрольные вопросы

1. Основной инструментарий для работы с СУБД Oracle?
2. DDL и DML?
3. Создание базы данных?
4. Создание таблиц?
5. Элементы управления безопасностью и разграничения прав?

СПИСОК ЛИТЕРАТУРЫ

1. Власов А.И., Лыткин С.Л., Яковлев В.Л. Краткое практическое руководство по языку PL/SQL - М.: Машиностроение. 2000. 64 с.
2. Сервер Oracle. Справочное руководство по языку SQL / Под ред. А.В.Емельяненко, Н.В. Емельяненко - Протвино, АО РДТех, 1994.
3. Сервер ORACLE. Основные концепции/ Под ред. А.В.Емельяненко, Н.В. Емельяненко - Протвино, АО РДТех, 1996.
4. Проектирование и эксплуатация конструкторско-технологических баз данных на основе СУБД Oracle/ Конспект лекций - М.: Москва, МГТУ им. Н.Э. Баумана, 2001, 120 с.