

Московский Государственный Технический Университет им. Н.Э. Баумана
Факультет «Информатика и системы управления»
Кафедра «Компьютерные системы и сети»

А.В. Брешенков, Е.В. Галямова, М.Н. Сурков

Методическое пособие по лабораторной работе №1
Предварительное знакомство с Oracle

Методическое пособие по лабораторной работе №2
Проектирование таблиц в Oracle Database 10g

Москва 2010

ОГЛАВЛЕНИЕ

Методическое пособие по лабораторной работе №1	
Предварительное знакомство с Oracle.....	4
1. Введение.....	4
2. Теоретическая часть.....	4
2.1. Запуск базы данных.....	4
2.2. Остановка базы данных.....	5
2.3. Доступ к домашней странице базы данных.....	6
2.4. Получение помощи.....	8
2.5. Описание домашней страницы.....	8
2.6. Навигация посредством кнопок меню.....	9
2.7. Навигация посредством цепочки ссылок.....	10
2.8. Импорт, экспорт, загрузка и выгрузка данных.....	10
2.9. Обзор резервного копирования и воссоздание данных.....	14
2.10. Восстановление удаленных таблиц.....	14
3. Практическая часть.....	16
3.1. Запуск Oracle.....	16
3.2. Выгрузка данных при помощи мастера загрузки.....	17
3.3. Загрузка данных при помощи мастера загрузки.....	18
3.4. Резервное копирование данных.....	19
3.5. Просмотр содержимого корзины.....	20
3.6. Восстановление таблицы из корзины.....	23
3.7. Очистка всего содержимого корзины.....	24
3.8. Управление базой данных с помощью Object Browser.....	26
3.9. Просмотр информации с помощью Object Browser.....	30
Контрольные опросы.....	30
Рекомендуемая литература.....	30
Методическое пособие по лабораторной работе №2	
Проектирование таблиц в Oracle Database 10g.....	31
1. Теоретическая часть.....	31
1.1. Таблицы.....	31
1.2. Представления.....	34
1.3. Индексы.....	36
1.4. Последовательности.....	37
1.5. Синонимы.....	38
2. Практическая часть.....	38

2.1. Таблицы.....	38
2.1.1. Создание таблицы.....	38
2.1.2. Добавление столбцов в таблицу.....	39
2.1.3. Изменение столбца в таблице.....	40
2.1.4. Удаление столбца из таблицы.....	41
2.1.5. Создание первичного ключа.....	41
2.1.6. Создание вторичного ключа.....	42
2.1.7. Добавление данных в таблицу.....	43
2.1.8. Изменение данных в таблице.....	43
2.1.9. Удаление строки данных в таблицею.....	44
2.1.10. Удаление таблицы.....	44
2.1.11. Добавление проверочного ограничения.....	44
2.1.12. Добавление уникального ограничения.....	45
2.1.13. Просмотр существующих ограничений.....	46
2.1.14. Включение или выключение ограничений.....	46
2.2. Индексы.....	47
2.2.1. Создание индекса.....	47
2.2.2. Просмотр индексов.....	48
2.2.3. Удаление индекса.....	48
2.3. Представления.....	49
2.3.1. Создание представления.....	49
2.3.2. Просмотр представления.....	50
2.3.3. Удаление представления.....	50
2.4. Последовательности.....	50
2.4.1. Создание последовательности.....	50
2.4.2. Просмотр последовательности.....	51
2.4.3. Удаление последовательности.....	51
2.5. Синонимы.....	51
2.5.1. Создание синонима.....	51
2.5.2. Просмотр синонима.....	52
2.5.3. Удаление синонима.....	52
Контрольные вопросы.....	53
Рекомендуемая литература.....	53

Методическое пособие по лабораторной работе №1

Предварительное знакомство с Oracle

Цель работы – Ознакомиться с основными возможностями Oracle Database 10g.

1. Введение

Oracle Database Express Edition (Oracle Database XE) - это бесплатная, меньшая по объему версия Oracle Database. Oracle Database XE проста в установке и управлении.

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 дополнение к Oracle Application Express release 2.1 Вы можете использовать любые языки и инструменты как Oracle, так и сторонних разработчиков для создания собственных приложений, работающих с Oracle Database XE.

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

Oracle Database Express Edition- реляционная база данных, которая хранит и восстанавливает наборы связанной информации. База данных, также названная сервером базы данных, является ключом к решению проблем информационного управления. В реляционной базе данных, коллекции связанной информации организованы в структуры, названные таблицами. Каждая таблица содержит строки (записи), которые составлены из колонок (полей). Таблицы сохранены в базе данных в структурах, под названием схемы,

которые являются логическими структурами данных, где пользователи базы данных хранят свои таблицы.

Схема HR, которая включена в Oracle Database XE, является примером схемы со связанными таблицами. В схеме HR есть таблицы, которые хранят информацию о служащих и отделах. Таблицы содержат общие колонки, которые позволяют данным от одной таблицы быть связанными с другим. Для каждого служащего в таблице employees название отдела из таблицы departments может быть восстановлено, т.к. колонка department ID находится в обеих таблицах.

2. Теоретическая часть

2.1. Запуск базы данных

Oracle Database Express Edition (Oracle Database XE) запускается автоматически сразу после инсталляции и после каждой системной перезагрузки. Таким образом, нет необходимости запускать базу данных, если предварительно Вы ее не остановили.

Вы можете запустить базу данных непосредственно с рабочего стола или при помощи командной строки SQL (SQL*Plus).

1) Запуск базы данных с рабочего стола

Для запуска базы данных с рабочего стола:

- Подключитесь к серверу Oracle Database XE как администратор Windows - т.е., как пользователь, являющийся членом группы «Администраторы»
- Щелкните Start, укажите на Programs (или All Programs), перейдите на Oracle Database 10g Express Edition, после чего выберите Start Database

2) Запуск базы данных с использованием командной строки SQL

- Подключитесь к серверу Oracle Database XE под пользователем, входящим в группу ORA_DBA. Обычно это пользователь, под которым была произведена установка Oracle Database XE.
- Откройте терминальную сессию или командное окно, если оно еще не открыто
- На приглашение операционной системы введите следующую команду для запуска командной строки SQL и соединения с базой данных:

```
SQLPLUS / AS SYSDBA
```

Слэш (/) указывает, что база данных должна идентифицировать Вас средствами операционной системы.

- На приглашение командной строки SQL введите следующую команду:

STARTUP

- После успешного завершения команды, она отобразит следующий вывод. (Размер системной глобальной области может меняться в зависимости от количества физической памяти в Вашем сервере Oracle Database XE.)

ORACLE instance started.

```
Total System Global Area  599785472 bytes
Fixed Size                  1220804 bytes
Variable Size               180358972 bytes
Database Buffers           415236096 bytes
Redo Buffers                2969600 bytes
```

Database mounted.

Database opened.

- Введите следующую команду для выхода из командной строки SQL: EXIT

2.2. Остановка базы данных

Oracle Database Express Edition (Oracle Database XE) останавливается автоматически одновременно с завершением работы компьютера, на котором она установлена.

Перед остановкой Oracle Database XE рекомендуется убедиться, что все пользователи и приложения завершили свою работу и закрыли соединение с базой данных.

Если пользователи или приложения имеют открытые соединения на момент начала остановки базы данных, остановка произойдёт следующим образом:

- Будет запрещено выполнение новых соединений и запрещён запуск новых транзакций.
- Для всех неподтверждённых транзакций будет выполнен откат.
- Будут принудительно закрыты соединения всех пользователей и приложений.

Вы можете остановить базу данных как с рабочего стола, так и с помощью командной строки SQL (SQL*Plus).

1) Остановка базы данных с рабочего стола

Чтобы остановить базу данных с рабочего стола:

- Подключитесь к серверу Oracle Database XE как администратор Windows - т.е., как пользователь, являющийся членом группы «Администраторы»

- Щелкните Start (Пуск), укажите на Все программы (All Programs), перейдите на Oracle Database 10g Express Edition, после чего выберите Stop Database

2) Остановка базы данных с использованием командной строки SQL

Чтобы завершить работу базы данных с помощью командной строки SQL:

1. Подключитесь к серверу Oracle Database XE под пользователем, входящим в группу ORA_DBA. Обычно это пользователь, под которым была произведена установка Oracle Database XE.
2. Откройте терминальную сессию или командное окно, если оно еще не открыто.
3. На приглашение операционной системы введите следующую команду для запуска командной строки SQL и соединения с базой данных:

```
SQLPLUS / AS SYSDBA
```

Слэш (/) указывает, что база данных должна идентифицировать Вас средствами операционной системы.

4. На приглашение командной строки SQL введите следующую команду:

```
SHUTDOWN IMMEDIATE
```

Имейте в виду, что выполнение команды может занять некоторое время. Если команда выполнена успешно, на экране появится следующее сообщение:

```
Database closed.
```

```
Database dismounted.
```

```
ORACLE instance shut down.
```

5. Если в течение нескольких минут никакого сообщения не появляется, и кажется, что ничего не происходит, Вы можете нажать CTRL-C, чтобы прервать выполнение команды, после чего введите следующую команду:

```
SHUTDOWN ABORT
```

База данных должна будет пройти процесс восстановления при запуске после команды SHUTDOWN ABORT.

6. Рекомендуется выполнить восстановление сразу же, после чего Вы можете остановить базу данных обычным способом. Для этого выполните следующие команды по завершению SHUTDOWN ABORT:

```
STARTUP
```

```
SHUTDOWN IMMEDIATE
```

7. Введите следующую команду для выхода из командной строки SQL:

```
EXIT
```

2.3. Доступ к домашней странице базы данных

- Доступ к домашней странице базы данных с рабочего стола

1. Нажмите Start, укажите на Programs (или All Programs), перейдите на Oracle Database 10g Express Edition, после чего выберите Go to Database Home Page.

2. При появлении страницы авторизации, подключитесь к базе данных, используя действующие имя пользователя базы данных и пароль. Чтобы подключиться с правами администратора используйте имя SYSTEM и пароль, который вы указали в процессе установки. Если регистрация прошла успешно, появится домашняя страница базы данных.

- **Доступ к домашней странице базы данных из веб-браузера (локально)**

1. Введите в своем веб-браузере следующую ссылку:

`http://127.0.0.1:port/apex`

здесь port - это TCP-порт для HTTP-запросов на соединение. Обычное его значение - 8080. Вы могли изменить это значение при инсталляции.

Например, если Вы установили Oracle Database XE с номером порта по умолчанию, то можете подключиться к домашней странице базы данных по этой ссылке:

`http://127.0.0.1:8080/apex`

2. При появлении страницы авторизации, подключитесь к базе данных, используя действующие имя пользователя базы данных и пароль.

Чтобы подключиться с правами администратора используйте имя SYSTEM и пароль, который вы указали в процессе установки.

Если регистрация прошла успешно, появится домашняя страница базы данных.

- **Доступ к домашней странице базы данных из веб-браузера (удаленно)**

Чтобы открыть домашнюю страницу базы данных через веб-браузер удаленного компьютера:

1. Введите в своем веб-браузере следующую ссылку:

`http://host:port/apex`

где:

- host - имя хоста или IP адрес компьютера, на котором установлена Oracle Database XE.
- port - это TCP-порт для HTTP-запросов на соединение. Обычное его значение - 8080. Вы могли изменить это значение при инсталляции.

Например, если Вы инсталировали Oracle Database XE на компьютер с именем хоста myhost.mydomain.com и номером порта по умолчанию, Вы получите доступ к домашней странице базы данных по этому URL-адресу:

`http://myhost.mydomain.com:8080/apex`

При появлении страницы авторизации, подключитесь к базе данных, используя действующие имя пользователя базы данных и пароль.

Чтобы подключиться с правами администратора используйте имя SYSTEM и пароль, который вы указали в процессе установки.

Если регистрация прошла успешно, появится домашняя страница базы данных.

2.4. Получение помощи

Вы можете получить доступ к контекстно-зависимой оперативной помощи следующими способами:

- Для получения помощи по работе с текущей страницей в графическом интерфейсе Oracle Database XE нажмите иконку Help в правом верхнем углу страницы. Откроется окно помощи. Кроме просмотра помощи по текущей странице, Вы можете использовать навигацию по разделам, раскрыв содержание в левой панели. Инструмент поиска в окне помощи позволяет осуществлять поиск по всем темам оперативной помощи. Чтобы выполнить поиск, нажмите ссылку Find в верхней части окна помощи.

- Для получения подсказки по отдельным элементам страницы, остановите указатель мыши над обозначением элемента. Если обозначение элемента изменит свой цвет на красный, и указатель мыши изменится на стрелку со знаком вопроса, значит для выбранного элемента страницы имеется подсказка. Нажмите на обозначение элемента для просмотра помощи в отдельном окне.

2.5. Описание домашней страницы

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:
 1. SQL Commands—Запуск команд SQL и анонимных блоков PL/SQL. Запуск скриптов и сохраненных запросов.
 2. 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.

Примечание:

При входе на домашнюю страницу базы данных под пользователем, отличным от SYSTEM, то становится доступной пиктограмма Application Builder. Вы можете использовать Application Builder для создания приложений Oracle Application Express.

2.6. Навигация посредством иконок и меню

Вы можете перемещаться по пользовательскому графическому интерфейсу Oracle Database XE щелкая большие иконки на домашней странице базы данных или на других навигационных страницах. При использовании этих иконок, у Вас есть два варианта:

- Щелчок по иконке — Щелкните иконку для перехода на страницу, на которую указывает название иконки.

Возможно, Вам понадобится щелкнуть несколько этих иконок, каждый раз опускаясь на один уровень в иерархии страниц, прежде чем Вы достигнете нужной страницы.

- Выбор из выпадающего меню иконки — Щелкните стрелку вниз с правой стороны иконки, чтобы увидеть выпадающее меню, а затем выберите пункт меню или любого из его подменю.

Это более короткий путь для перехода на некоторые страницы.



На данной иллюстрации изображены две большие пиктограммы, расположенные слева-направо по домашней странице базы данных. Левая из них изображена с раскрытым выпадающим меню. В меню находятся связанные с пиктограммой команды. Некоторые из команд имеют подпункты.

2.7. Навигация посредством цепочек ссылок

Цепочки ссылок (называемые также локационными ссылками) расположены в верхней части каждой страницы пользовательского графического интерфейса Oracle Database XE. Хвост в цепочке ссылок показывает положение текущей страницы в иерархии страниц. Вы можете использовать цепочки ссылок для прямого перехода на предыдущую страницу или на любую страницу, находящуюся выше текущей в иерархии. В следующем примере, который был взят со страницы Построителя Запросов, нажатие на Home переместит Вас на домашнюю страницу базы данных:



Это изображение показывает хвост цепочки ссылок со страницы Query Builder, переход на страницу SQL и голову цепочки (переход на страницу Home). Хвост идет слева-направо и самая левая локационная ссылка представляет вершину иерархии. Символ "больше чем" (>) разделяет ссылки в цепочке.

2.8. Импорт, экспорт, загрузка и выгрузка данных

Oracle Database XE позволяет копировать данные между базами данных Oracle, а также обмениваться данными с внешними файлами. Копирование данных осуществляется

посредством экспорта и импорта, а также посредством выгрузки и загрузки. В следующей таблице имеются определения перечисленных терминов.

Термин	Определение
Экспорт	Копирование данных во внешние файлы только для импорта в другую базу данных Oracle. Такие файлы имеют собственный бинарный формат.
Импорт	Копирование данных в базу из внешних файлов, которые были созданы экспортом из другой базы данных Oracle.
Выгрузка	Копирование данных базы во внешние текстовые файлы для использования в другой базе данных Oracle или в другом приложении (например в табличном процессоре). Текстовые файлы имеют общепринятый формат, например с разделением табуляцией или запятыми (csv).
Загрузка	Копирование данных в базу из внешних текстовых файлов, которые сохранены в формате с разделением табуляцией, запятыми или в любом другом формате, поддерживаемом утилитой SQL*Loader.

2.8.1. Выбор подходящего варианта импорта, экспорта, загрузки, выгрузки

Oracle Database Express Edition (Oracle Database XE) предоставляет на выбор несколько полноценных вариантов для импорта, экспорта, загрузки и выгрузки данных. В таблице перечислены эти возможные варианты.

Функционал или утилита	Описание
Мастера загрузки/выгрузки данных графического пользовательского интерфейса Oracle Database XE	<ul style="list-style-type: none"> • Простой в использовании графический интерфейс • Загружает/выгружает из и во внешние текстовые файлы (в формате с разделителями) или XML-файлы • Загружает/выгружает только таблицы, по одной за раз • Доступ только к схеме подключившегося пользователя • Нет фильтрации данных
Утилита SQL*Loader	<ul style="list-style-type: none"> • Интерфейс командной строки, вызываемый

	<p>командой <code>sqlldr</code></p> <ul style="list-style-type: none"> • Массовая загрузка данных из внешних файлов в базу данных • Поддерживает множество форматов ввода, включая формат с ограничителями, формат с фиксированной длиной записи, формат с переменной длиной записи, потоковый формат • Загружает одновременно несколько таблиц • Полноценные возможности для фильтрации данных
<p>Утилиты "Помпа данных экспорта" и "Помпа данных импорта"</p>	<ul style="list-style-type: none"> • Интерфейс командной строки, вызываемый командами <code>expdp</code> и <code>impdp</code> • Выполняют экспорт и импорт из одной базы данных Oracle в другую (собственный бинарный формат) • Импорт/экспорт все типов объектов схемы • Импорт/экспорт всей базы данных, всей схемы, нескольких схем, нескольких табличных пространств и нескольких таблиц • Полноценные возможности для фильтрации данных • Высокая скорость выполнения операций • Отсутствует поддержка XMLType
<p>Export and Import utilities</p>	<ul style="list-style-type: none"> • Интерфейс командной строки, вызываемый командами <code>exp</code> и <code>imp</code> • Выполняют экспорт и импорт из одной базы данных Oracle в другую (собственный бинарный формат) • Имеется поддержка XMLType • Отсутствует поддержка типов данных <code>FLOAT</code> и <code>DOUBLE</code> • Возможности аналогичны помпе данных. <p>Предпочтительно использовать помпу данных (Data Pump), если только вам не надо выполнить импорт или экспорт данных XMLType</p>

2.8.2. Выгрузка и загрузка данных

Вы можете выгрузить и загрузить данные следующими способами:

- Выгрузить и загрузить данные при помощи мастеров Выгрузки/Загрузки данных пользовательского графического интерфейса Oracle Database XE

Мастера могут использовать только текстовые файлы в формате с разделителями.

- Загрузить данные с помощью утилиты Oracle SQL*Loader
SQL*Loader поддерживает множество форматов текстовых файлов и предоставляет очень широкие возможности.

Мастера загрузки/выгрузки данных графического пользовательского интерфейса Oracle Database XE позволяют Вам легко загружать и выгружать текстовые данные в формате с разделителями в/из базы данных. Пошаговые мастера имеют следующие возможности:

- Вы можете импортировать или экспортировать XML файлы или текстовые файлы с разделителями (такие, как файлы с разделителем-запятой (.csv) или табулятором).
- Вы можете загружать данные копированием и вставкой из электронных таблиц.
- Вы можете не включать (пропускать) столбцы при загрузке или выгрузке.
- Вы можете загружать в уже существующую таблицу или создать новую таблицу из загружаемых данных.
- При загрузке в новую таблицу, первичный ключ может быть взят из данных или сгенерирован из новой, или существующей последовательности Oracle.
- При загрузке в новую таблицу, наименования столбцов могут быть взяты из загружаемых данных.
- При каждой загрузке из файлов, они сохраняются в архиве загрузки текстовых данных. Вы можете получить к ним доступ из архива в любой момент.

Ограничения заключаются в следующем:

- Мастера загружают и выгружают только табличные данные. Они не могут загрузить или выгрузить другие типы объектов схемы.

- Вы можете загружать или выгружать только в/из принадлежащей вам схемы. Это относится и к пользователям с административными привилегиями.

- Вы можете загрузить или выгрузить только одну таблицу за раз.
- Нет ограничений на типы данных для выгрузки в текст, выгрузки в XML или загрузки XML данных. Тем не менее, загрузка текстовых данных и загрузка данных

электронных таблиц (через копирование и вставку) поддерживают только следующие типы данных: NUMBER, DATE, VARCHAR2, CLOB, BINARY_FLOAT и BINARY_DOUBLE.

2.9. Обзор резервного копирования и восстановления базы данных

Резервное копирование и восстановление Oracle Database XE основано на защите физических файлов, составляющих базу данных: файлов данных, управляющего файла, серверного файла параметров (SPFILE) и, если включен режим ARCHIVELOG, журнальных файлов.

В Oracle Database XE функции резервного копирования и восстановления выполняет утилита Recovery Manager (RMAN), которая интегрирована в базу данных. Несмотря на то, что существует клиент командной строки RMAN (похожий на командную строку SQL), вам нет необходимости использовать его напрямую для создания резервных копий и восстановления базы данных. В Oracle Database XE включены готовые скрипты для резервного копирования и восстановления, которые вы можете запустить прямо с рабочего стола. Эти скрипты позволяют создать полную резервную копию базы данных и ее восстановление из резервной копии, которая сохраняется в области мгновенного восстановления.

2.9.1. Автоматическое управление хранилищем резервных копий

В Oracle Database XE реализована политика сохранения резервных копий, согласно которой всегда должны существовать две последние полные резервные копии базы данных, для обеспечения избыточного уровня защиты базы данных. В режиме ARCHIVELOG все архивные журналы, необходимые для восстановления носителя, также сохраняются. База данных автоматически управляет резервными копиями и архивными журналами в области мгновенного восстановления, удаляя все устаревшие (т.е. которые больше не нужны согласно политике сохранения резервных копий) по мере возникновения необходимости в свободном пространстве для новых файлов. Включенный в состав Oracle Database XE скрипт создания резервной копии также удаляет устаревшие резервные копии и архивные журналы по окончании задания создания резервной копии.

2.10. Восстановление удаленных таблиц

При удалении таблицы, база данных не очищает сразу же место, занимаемое этой таблицей. Сервер базы данных переименовывает таблицу и помещает её и все зависимые

объекты в корзину. В случае если удаление было произведено ошибочно, таблица может быть позднее восстановлена.

2.10.1. О корзине

Корзина - это таблица словаря данных, содержащая информацию об удалённых объектах. Удаленные объекты и любые зависимые объекты (такие как индексы, ограничения целостности, вложенные таблицы и т.д.) не удалятся и продолжают занимать место, пока вы не вычистите их из корзины или пока они не будут вычищены из-за ограничений объема в табличном пространстве.

Вы можете восстанавливать объекты из корзины, т.е. как бы "вынимать" их. При восстановлении объекта, он возвращается в том состоянии, в котором находился до операции удаления. При восстановлении таблицы, также автоматически восстанавливаются все зависимые объекты.

Вы можете использовать графический интерфейс Oracle Database XE для просмотра содержимого корзины, воссоздания таблиц и зависимых объектов из корзины, а также для очистки корзины. Для просмотра, очистки или восстановления объектов определенного пользователя, вы должны подключиться под учетной записью этого пользователя.

2.10.2. Очистка корзины

При удалении объектов, занимаемое ими место не будет освобождено до тех пор, пока не выполнит очистку корзину, или пока корзина не будет очищена автоматически при нехватке свободного пространства. Вы можете очищать как отдельные объекты из корзины, так и все содержимое корзины целиком. Вы можете очистить из корзины только те объекты, которыми владеете.

3. Практическая часть

4.

4.1. Запуск Oracle.

1. Перед тем как приступить к работе необходимо проверить запущена ли база данных.
2. Далее необходимо запустить главную страницу Oracle. Чтобы открыть домашнюю страницу базы данных необходимо выполнить одно из следующих действий:
 - Доступ к домашней странице базы данных с рабочего стола
 - Доступ к домашней странице базы данных из веб-браузера (локально)
 - Доступ к домашней странице базы данных из веб-браузера (удаленно)
3. В появившемся окне надо ввести следующее:
 - В поле Username необходимо ввести HR
 - В поле Password необходимо ввести пароль, который вам выдаст преподаватель
 - Затем нажать на кнопку Login

4. В случае успешного входа Вы попадаете на главную страницу Oracle

4.2. Выгрузка данных при помощи мастера выгрузки

Предположим, вы хотите выгрузить таблицу `REGIONS`, которая является частью пробной схемы `HR`, для того, чтобы ее можно было использовать в другом приложении. Предположим также, что вы хотите создать текстовый файл с разделителем-табулятором, и вы хотите сохранить данные в файл под именем `regions.txt`.

Для выгрузки таблицы `REGIONS`:

1. Подключитесь к графическому пользовательскому интерфейсу Oracle Database XE в качестве пользователя `HR`.
2. На домашней странице базы данных, щелкните иконку `Utilities`, и далее иконку `Data Load/Unload`.
3. На странице `Data Load/Unload` щелкните иконку `Unload`, и далее иконку `Unload to Text`.

Откроется страница `Unload to Text`, показывая шаг мастера `Schema`. На этом шаге мастер отображает раскрывающийся список `Schema`, в котором выбрана `HR`. Так как вы можете экспортировать только из своей собственной схемы, то нельзя изменить этот выбор.

4. Нажмите `Next`.

Откроется шаг `Table Name` мастера.

5. Из выпадающего списка `Table` выберите `REGIONS` и нажмите `Next`.

Откроется шаг `Columns` мастера.



6. Выберите все столбцы щелчком-перетягиванием или `shift`-щелчком, а затем нажмите `Next`. Откроется шаг `Options` мастера.
7. Выполните следующие действия:

- a. В поле Separator удалите запятую (если есть) и введите обратный слэш и символ T в нижнем регистре (\t), чтобы в качестве разделителя использовать символ табуляции. (Вы можете использовать любой символ в качестве разделителя.)
- b. Установите флажок Include Column Names.
Установка этого флажка приведет к тому, что первой выгруженной строкой будут являться наименования столбцов, а не данные. Позднее вы сможете использовать эту первую строку для установки наименований столбцов при загрузке.
- c. В выпадающем списке File Character Set выберите Unicode UTF-8.

Separator \t Optionally Enclosed By Include Column Names
 File Format DOS
 File Character Set Unicode UTF-8

8. Нажмите Unload Data.

Появится окно Save As с именем файла `regions.txt`. В зависимости от используемого браузера этому окну может предшествовать другое окно с запросом, хотите ли вы сохранить или открыть файл. В этом случае выберите вариант для сохранения файла на диск.

9. Сохраните файл `regions.txt` на рабочем столе или в каталоге по своему усмотрению.
10. Откройте файл `regions.txt` в текстовом редакторе или в табличном процессоре и убедитесь, что таблица `REGIONS` выгружена должным образом.

4.3. Загрузка данных при помощи мастера загрузки

Предположим, что вашему приложению требуется таблица `REGIONS`, каждая строка которой состоит из кода и названия части света. Также предположим, что Вы выгрузили данные о частях света из настольной базы данных в размеченный табуляцией текстовый файл, который называется `regions.txt`.

Код региона Вы хотите использовать в качестве бизнес-ключа, а не в качестве первичного ключа, поэтому решаете использовать мастер загрузки так же и для генерации числового первичного ключа для каждой загружаемой записи.

Для загрузки таблицы `REGIONS`:

1. Войдите в графический интерфейс Oracle Database XE под любой учетной записью, кроме `SYSTEM` `SYSTEM` и `HR`.

2. На домашней странице базы данных выберите иконку Utilities, после чего - Data Load/Unload.
3. На странице Data Load/Unload щелкните по иконке Load, а затем по Load Text Data. Откроется страница Load Data на шаге Target and Method мастера.
4. Под заголовком Load To выберите New table, а под заголовком Load From выберите Upload file (comma separated or tab delimited).
5. Нажмите Next.
Откроется шаг File Details мастера.
6. Выполните следующие действия:
 - a. Нажмите Browse, выберите файл regions.txt и нажмите Open.
 - b. В поле Separator замените запятую на обратный слэш и T в нижнем регистре (\t) для указания того, что разделителем полей является символ табуляции.
 - c. В выпадающем списке File Character Set выберите Unicode UTF-8.

* File C:\Documents and Settings\sfogel.ST-USER\ST-Users\Desktop\regions.txt

* Separator: (\t for tab) \t

Optionally Enclosed By

First row contains column names.

File Character Set Unicode UTF-8

- d. Нажмите Next.

Откроется шаг Table Properties мастера.

7. Выполните следующие действия:
 - a. В поле Table Name введите REGIONS.

Load Data

* Schema STEVE

* Table Name REGIONS Preserve Case

Set Table Properties

Column Names	REGION_ID	REGION_NAME
Data Type	NUMBER	VARCHAR2
Format		
Column Length	1	30
Upload	Yes	Yes
Row 1	1	Europe
Row 2	2	Americas
Row 3	3	Asia
Row 4	4	Middle East and Africa

- b. Примите умолчания (Yes) во всех выпадающих списках Upload.
Установкой Upload в No можно исключить столбец из операции загрузки.
- c. Нажмите Next.

Откроется шаг Primary Key мастера.

8. Выполните следующие действия:
 - a. В Primary Key From выберите Create new column.
 - b. В Primary Key Population выберите Generated from a new sequence.

Schema: **STEVE**
Table Name: **REGIONS**

Primary Key From: Use an existing column
 Create new column

* New Primary Key Column:

* PK Constraint Name:

Primary Key Population: Generated from a new sequence
 Generated from an existing sequence
 Not generated

* Sequence:

9. Сделанный выбор приведет к следующим действиям Oracle Database XE:
 - Создание в таблице дополнительного столбца ID, который будет использован в качестве первичного ключа для REGIONS.
 - Создание новой последовательности REGIONS_SEQ.
 - Использование значений из последовательности при заполнении поля ID для каждой добавляемой новой строки в таблице.

Если вы не хотите создавать новый первичный ключ, а хотите использовать существующее поле REGION_ID в качестве первичного ключа, выполните следующие действия:

- Выберите Use an existing column.
- В выпадающем списке Primary Key выберите REGION_ID(NUMBER).
- Выберите Not generated.

Schema: **HR**
Table Name: **REGIONS**

Primary Key From: Use an existing column
 Create new column

* Primary Key:

* PK Constraint Name:

Primary Key Population: Generated from a new sequence
 Generated from an existing sequence
 Not generated

10. Нажмите кнопку Load Data.

Загрузка начнется, а когда она закончится, то появится страница Text Data Load Repository, показывающая файл `regions.txt` вверху списка загруженных файлов.

11. Проверьте статус загрузки файла `regions.txt` в колонках Succeeded и Failed.

Числа в этих колонках показывают количество успешно или неудачно импортированных строк.

4.4. Резервное копирование базы данных

Для выполнения резервного копирования базы данных:

1. Войдите на сервер Oracle Database XE под пользователем, который является членом группы `ORA_DBA`. Обычно это тот пользователь, который произвел установку Oracle Database XE.
2. Нажмите Start, укажите на Programs (or All Programs), затем на Oracle Database 10g Express Edition и выберите Backup Database.

Откроется консольное окно, чтобы вы могли управлять действиями скрипта создания резервной копии.

Если экземпляр базы данных работает в режиме `ARCHIVELOG`, то скрипт покажет следующее сообщение:

```
Doing online backup of the database.
```

Если экземпляр базы данных работает в режиме `NOARCHIVELOG`, то скрипт покажет следующее сообщение:

```
Warning: Log archiving (ARCHIVELOG mode) is currently disabled. If
you restore the database from this backup, any transactions that take
place between this backup and the next backup will be lost. It is
recommended that you enable ARCHIVELOG mode before proceeding so
that all transactions can be recovered upon restore. See the section
'Enabling ARCHIVELOG Mode...' in the online help for instructions.
Backup with log archiving disabled will shut down and restart the
database. Are you sure [Y/N]?
```

3. Если потребуется, введите `y` и нажмите Enter для подтверждения остановки базы данных и начала создания резервной копии.

По завершению резервного копирования, скрипт отобразит сообщение:

```
Backup of the database succeeded.
```

```
Log file is at location
```

Press ENTER key to exit

где location - это это расположение файла журнала выполненных действий.

4. Нажмите Enter чтобы закрыть окно Backup Database.

4.5. Просмотр содержимого корзины

Для просмотра содержимого корзины:

1. Откройте домашнюю страницу базы данных и подключитесь под владельцем интересующей вас схемы.
2. Щелкните по пиктограмме Utilities, а затем по Recycle Bin.
3. Щелкните по иконке Dropped Objects.

Откроется страница Dropped Objects, в которой показан список удаленных объектов и зависимых объектов. (Например, если в списке имеется таблица, то также будут показаны все ее индексы и другие зависимые объекты.)

User: HR

Home > Utilities > Recycle Bin > **Dropped Objects**

Object Type Original Name Display

Original Name	Type	Drop Date	Dropped
JHIST_JOB_IX	INDEX	2005-10-24:21:30:17	15 hours ago
JHIST_EMPLOYEE_IX	INDEX	2005-10-24:21:30:17	15 hours ago
JOB_HISTORY	TABLE	2005-10-24:21:30:17	15 hours ago
JHIST_EMP_ID_ST_DATE_PK	INDEX	2005-10-24:21:30:17	15 hours ago
JHIST_DEPARTMENT_IX	INDEX	2005-10-24:21:30:17	15 hours ago

row(s) 1 - 5 of 5

4. Можно ограничить содержимое списка, выбрав определенный тип объектов в выпадающем списке Object Type и нажав Go.
5. Можно ограничить содержимое списка, если ввести искомый текст в поле Original Name и нажать Go.
Будут отобраны все объекты, у которых в исходном наименовании встречается искомый текст. К примеру, если выведен список аналогичный тому, что показан на предыдущей иллюстрации, то при вводе JOB в качестве искомой строки, в списке будут отобраны только объекты JOB_HISTORY и JHIST_JOB_IX.
6. Щелкните по ссылке объекта (в столбце Original Name) для просмотра подробной информации об объекте.

4.6. Восстановление таблицы из корзины

Предположим, что вы удалили таблицу `JOB_HISTORY` из схемы `HR`, а потом решили, что хотите восстановить ее. Вы можете восстановить (вынуть) таблицу с помощью восстановления из корзины.

Чтобы таблицу `JOB_HISTORY`, а затем восстановить ее из корзины:

1. Откройте домашнюю страницу базы данных и войдите под пользователем `HR`.
2. Кликните по иконке Object Browser.

Откроется страница Object Browser, в которой отображены таблицы схемы `HR`.

3. Щелкните по таблице `JOB_HISTORY`.

Информация об этой таблице откроется с правой стороны страницы.

JOB_HISTORY					Create ▾					
Table	Data	Indexes	Model	Constraints	Grants	Statistics	UIDefaults	Triggers	Dependencies	SQL
Add Column	Modify Column	Rename Column	Drop Column	Rename	Copy	Drop	Truncate	Create Lookup Table		
Column Name	Data Type	Nullable	Default	Primary Key						
EMPLOYEE_ID	NUMBER(6,0)	No	-	1						
START_DATE	DATE	No	-	2						
END_DATE	DATE	No	-	-						
JOB_ID	VARCHAR2(10)	No	-	-						
DEPARTMENT_ID	NUMBER(4,0)	Yes	-	-						
					1 - 5					

4. Нажмите кнопку Drop (под ссылкой Triggers).

Откроется окно Drop Object Confirmation.

5. Нажмите Finish чтобы удалить таблицу.

Таблица будет удалена из списка таблиц в левой стороне страницы.

6. Щелкните по хвосту Home в цепочке ссылок в верхней части страницы, чтобы вернуться на домашнюю страницу базы данных.
7. Щелкните по Utilities, а затем по Recycle Bin.
8. Щелкните по иконке Dropped Objects.

Откроется страница Dropped Objects, отображая таблицу `JOB_HISTORY` и ее индексы в списке.

User: HR

Home > Utilities > Recycle Bin > **Dropped Objects**

Object Type Original Name Display

Original Name	Type	Drop Date	Dropped
JHIST_JOB_IX	INDEX	2005-10-24:21:30:17	15 hours ago
JHIST_EMPLOYEE_IX	INDEX	2005-10-24:21:30:17	15 hours ago
JOB_HISTORY	TABLE	2005-10-24:21:30:17	15 hours ago
JHIST_EMP_ID_ST_DATE_PK	INDEX	2005-10-24:21:30:17	15 hours ago
JHIST_DEPARTMENT_IX	INDEX	2005-10-24:21:30:17	15 hours ago

row(s) 1 - 5 of 5

9. В столбце Original Name кликните по имени таблицы `JOB_HISTORY`.

Откроется страница Object Details с информацией о таблице `JOB_HISTORY`.

User: HR

Home > Utilities > Recycle Bin > Dropped Objects > **Object Details**

Base Object Name:	JOB_HISTORY
Original Object Name:	JOB_HISTORY
Object Name:	BIN\$A/hg/zIEIKTgGFemiAdwVg==\$0
Type:	TABLE
Operation:	DROP
Created On:	2005-10-10:03:30:19
Dropped On:	2005-10-25:13:21:03
Can be restored ?	Yes
Can be purged ?	Yes

10. Нажмите Restore Object.

Откроется страница с подтверждающим сообщением. Таблица и все ее зависимые объекты теперь восстановлены.

4.7. Очистка всего содержимого корзины

Чтобы очистить все содержимое корзины:

1. Щелкните по иконке Utilities, а затем по Recycle Bin.
2. Щелкните по иконке Purge Recycle Bin.

Откроется страница с просьбой подтвердить операцию.

3. Щелкните по Purge Recycle Bin снова.

Появится подтверждающее сообщение.

Очистка индивидуальных объектов из корзины

Для очистки корзины от отдельных объектов:

1. Щелкните по иконке Utilities, а затем по Recycle Bin.
2. Щелкните по иконке Dropped Objects.

Откроется страница Dropped Objects, отображая список удаленных объектов.

User: HR

Home > Utilities > Recycle Bin > **Dropped Objects**

Object Type Original Name Display

Original Name	Type	Drop Date	Dropped
JHIST_JOB_IX	INDEX	2005-10-24:21:30:17	15 hours ago
JHIST_EMPLOYEE_IX	INDEX	2005-10-24:21:30:17	15 hours ago
JOB_HISTORY	TABLE	2005-10-24:21:30:17	15 hours ago
JHIST_EMP_ID_ST_DATE_PK	INDEX	2005-10-24:21:30:17	15 hours ago
JHIST_DEPARTMENT_IX	INDEX	2005-10-24:21:30:17	15 hours ago

row(s) 1 - 5 of 5

3. В столбце Original Name щелкните по наименованию объекта, который вы хотите вычистить.

Откроется страница Object Details, отображая подробную информацию об объекте.

User: HR

Home > Utilities > Recycle Bin > Dropped Objects > **Object Details**

Base Object Name:	JOB_HISTORY
Original Object Name:	JOB_HISTORY
Object Name:	BIN\$A/ng/zIEIKTgGFeMiAdwVg==\$0
Type:	TABLE
Operation:	DROP
Created On:	2005-10-10:03:30:19
Dropped On:	2005-10-25:13:21:03
Can be restored ?	Yes
Can be purged ?	Yes

4. Нажмите Purge Object.

Откроется страница с подтверждающим сообщением. Объект и все его зависимые объекты теперь вычищены.

4.8. Управление базой данных с помощью Object Browser

Вы можете использовать **Object Browser** для создания, изменения или для просмотра вашей базы данных. Например, с помощью **Object Browser** Вы можете создать таблицу и затем изменить ее путем добавления и удаления колонок или добавляя ограничения. Вы можете также просмотреть все объекты, которые в настоящее время используются в схеме.

1. На главной странице нажмите на иконку **Object Browser**

На странице **Object Browser** показаны две секции:

- Слева показан список объектов базы данных для выбранного типа
- Справа показана детальная информация об объекте и действия, которые можно применить для данного объекта

2. Выберите таблицу `EMPLOYEES` для просмотра структуры данной таблицы.

Column Name	Data Type	Nullable
EMPLOYEE_ID	NUMBER(6,0)	No
FIRST_NAME	VARCHAR2(20)	Yes
LAST_NAME	VARCHAR2(25)	No
EMAIL	VARCHAR2(25)	No
PHONE_NUMBER	VARCHAR2(20)	Yes
HIRE_DATE	DATE	No
JOB_ID	VARCHAR2(10)	No
SALARY	NUMBER(8,2)	Yes
COMMISSION_PCT	NUMBER(2,2)	Yes
MANAGER_ID	NUMBER(6,0)	Yes
DEPARTMENT_ID	NUMBER(4,0)	Yes

Home > Object Browser

Column Name	Data Type	Nullable	Default	Prim
EMPLOYEE_ID	NUMBER(6,0)	No	-	
FIRST_NAME	VARCHAR2(20)	Yes	-	
LAST_NAME	VARCHAR2(25)	No	-	
EMAIL	VARCHAR2(25)	No	-	
PHONE_NUMBER	VARCHAR2(20)	Yes	-	
HIRE_DATE	DATE	No	-	
JOB_ID	VARCHAR2(10)	No	-	
SALARY	NUMBER(8,2)	Yes	-	
COMMISSION_PCT	NUMBER(2,2)	Yes	-	
MANAGER_ID	NUMBER(6,0)	Yes	-	
DEPARTMENT_ID	NUMBER(4,0)	Yes	-	

3. Изучите структуру других таблиц
4. В списке объектов выберите **Views** и кликните на emp_details_view для отображения структуры данного представления
5. В списках объектов выберите другие типы данных и изучите данные типы

4.9. Просмотр данных в таблице с помощью Object Browser

1. На главной странице нажмите на иконку **Object Browser**
2. В списке объектов выберите **Tables** и кликните на таблицу employees
3. Нажмите на кнопку Data для просмотра содержимого таблицы

EMPLOYEES										
Table	Data	Indexes	Model	Constraints	Grants	Statistics	UI Defaults	Triggers	Dependencies	SQL
Query	Count Rows	Insert Row								
EDIT	EMPLOYEE_ID	FIRST_NAME	LAST_NAME	EMAIL	PHONE_NUMBER	HIRE_DATE				
	100	Steven	King	SKING	515.123.4567	17-JUN-87	AC			
	101	Neena	Kochhar	NKOCHHAR	515.123.4568	21-SEP-89	AC			
	102	Lex	De Haan	LDEHAAN	515.123.4569	13-JAN-93	AC			
	103	Alexander	Hunold	AHUNOLD	590.423.4567	03-JAN-90	IT			
	104	Bruce	Ernst	BERNST	590.423.4568	21-MAY-91	IT			
	105	David	Austin	DAUSTIN	590.423.4569	25-JUN-97	IT			
	106	Valli	Pataballa	VPATABAL	590.423.4560	05-FEB-98	IT			
	107	Diana	Lorentz	DLORENTZ	590.423.5567	07-FEB-99	IT			
	108	Nancy	Greenberg	NGREENBE	515.124.4569	17-AUG-94	FL			

4.10. Просмотр информации с помощью Object Reports

1. На главной странице кликните по иконке **Utilities**
2. На странице **Utilities** выберите **Object Reports**.
3. На странице **Object Reports** кликните на иконку **All Objects**
4. На странице **All Objects** кликните на иконку **Invalid Objects**
5. На странице **Invalid Objects** выберите в списке **Type** -All-
6. Нажмите на кнопку **Go** для просмотра неправильных объектов

Home > Utilities > Object Reports > All Objects > Invalid Objects				Schema	HR	
Object Name	<input type="text"/>	Type	- All -	Display	15	Go
Object Name	Object Type	Status				
AUDIT_SAL	TRIGGER	INVALID				
LOG_SALARY_INCREASE	TRIGGER	INVALID				
row(s) 1 - 2 of 2						

Контрольные вопросы

1. Перечислите основные пункты меню главной страницы Oracle Database
2. Для чего необходимо запускать базу данных?
3. Перечислите варианты запуска домашней страницы Oracle Database
4. Какие объекты находятся на главной странице?

5. Какие операции можно выполнять с данными? Дайте определения им.
6. С помощью каких инструментов Oracle Database можно выполнять резервное копирование данных?
7. Для чего необходима корзина?
8. Назовите способы навигации по меню

Рекомендуемая литература

1. Дейт К. Введение в системы баз данных, 8-е издание: Пер. с англ. – М.:Издательский дом "Вильямс". 2005.-1328 с.
2. Грабер М. Введение в SQL: Пер с англ. – М.:Изд-во'ЛОРИ',1996. – 380с.
3. Джеймс Перри, Джеральд Пост. Введение в Oracle 10g "И.Д. Вильямс", 2006. – 700 с.
4. Гринвальд Рик, Становьяк Роберт, Додж Гери, Кляйн Дэвид, Шапиро Бен, Челья Кристофер Дж. Программирование баз данных Oracle для профессионалов.: Пер. с англ.: – М. : ООО "И.Д. Вильямс", 2007. – 784 с.
5. Кайт Томас. Oracle для профессионалов: архитектура, методика программирования и основные особенности версии 9i и 10j.: Пер с англ. – М.:Издательский дом "Вильямс". 2008. – 848 с.
6. Кевин Луни, Боб Брилла. Oracle Database 10g. Настольная книга администратора баз данных. – М.:Издательство'Лори',2008. – 732 с.
7. Райан стивенс, Рональд Плю. SQL. Пер с англ. – М.:ЗАО 'Издательство Бином',1998.-400 с.

Методическое пособие по лабораторной работе №2

Проектирование таблиц в Oracle Database 10g

Цель работы – Научиться проектировать и работать с таблицами в СУБД Oracle 10g

1. Теоретическая часть

1.1. Таблицы

Таблица – это структура, которая хранит данные в реляционной базе данных. Таблица состоит из строк и столбцов. Таблица может представлять единственную сущность, которую вы хотите отобразить в вашей системе. Такая таблица может представлять, например, список сотрудников вашей организации, или заказы, размещенные на продукты вашей компании. Таблица может также представлять отношение между двумя сущностями. Такая таблица может, например, отображать ассоциацию между сотрудниками и их профессиями, или связи между продуктами и заказами. Внутри таблиц, такие отношения и связи представляются внешними ключами.

Oracle позволяет задать условия, которым должны удовлетворять все данные в БД. Условия задаются в декларативной форме, т. е. программист лишь задает условие, а сервер обеспечивает его проверку при любом изменении данных. Такие условия называются правилами целостности (integrity constraint).

В Oracle поддерживаются четыре правила целостности, налагающих ограничения на данные в пределах одной таблицы:

- запрет неопределенных значений (not null),
- проверка условия (check),
- уникальность значений (unique)
- первичный ключ (primary key).

Кроме того, для обеспечения ссылочной целостности существуют внешние ключи (foreign key).

Правило not null запрещает использование неопределенных значений. Если столбец объявлен как not null, данные в этом столбце не могут содержать значение null.

Более сложные правила целостности можно определить при помощи предложения `check`. Эти правила представляют собой логические выражения, которые должны выполняться для каждой строки таблицы. Они могут включать в себя выражения, содержащие значения столбцов, операции над ними и вызовы встроенных функций. Выражения в правилах `check` не могут содержать вызовов пользовательских хранимых функций и операций `select`.

Пусть, например, в таблице `employee` определены поля `salary`, `hiredate` и `firedate`. Тогда можно было бы ввести следующие правила `check`:

```
salary>100 -- минимальная зарплата
hiredate>='1-Jan-2001' -- защита от неправильного ввода
firedate>hiredate -- увольнение позже приема
```

Обратите внимание, что Oracle проверяет не положительный результат проверки, а отсутствие отрицательного результата. Это значит, что если для поля задано правило `salary>100`, то это поле может принимать значение `null`, т. к. результатом любого сравнения с `null` (кроме `is [not] null`) будет `null`, а не `true` или `false`.

Согласно правилу `unique`, в таблице не может существовать двух строк с одинаковыми значениями столбца или набора столбцов. При этом неопределенные значения столбцов не считаются одинаковыми, т. е. две и более строк могут содержать неопределенные (`null`) значения во всех столбцах, объявленных как `unique`. Пусть в таблице `test` существует правило целостности `unique`, наложенное на столбец `first`, и таблица пуста:

```
insert into test (first)      values (1) -- OK
insert into test (first)      values (1) -- ошибка
insert into test (first)      values (null) -- OK
insert into test (first)      values (null) -- OK
```

Пусть теперь правило `unique` задано для пары столбцов `first` и `second`:

```
insert into test      (first,second) values (1,2) -- OK
insert into test      (first,second) values (1,2) -- ошибка
insert into test      (first,second) values (1,null) -- OK
insert into test      (first,second) values (1,null) -- ошибка
insert into test      (first,second) values (null,null) -- OK
insert into test      (first,second) values (null,null) -- OK
```

Для того, чтобы поддержать правило целостности `unique`, Oracle использует индекс. Если подходящего индекса нет, то Oracle автоматически создает уникальный индекс по тому набору полей, на который наложено правило `unique`. Если правило будет уничтожено или запрещено, то автоматически созданный индекс будет уничтожен. Если для поддержания правила `unique` используется готовый индекс, то этот индекс не обязан быть

уникальным. Кроме того, в этом случае при уничтожении или запрещении правила индекс не уничтожается, соответственно, нет необходимости пересоздавать его при включении правила.

Правило `primary key` задает первичный ключ. Каждая строка таблицы однозначно определяется значениями столбца или набора столбцов, входящих в первичный ключ. Для обеспечения правила целостности `primary key` Oracle неявно создает правила `not null` для каждого столбца, входящего в ключ, и правило `unique` для набора столбцов. Правило `primary key` также может использовать готовый индекс, не обязательно уникальный. Таблица может иметь не более одного первичного ключа (а может и не иметь его вовсе).

Правило целостности может иметь имя. При создании безымянного правила целостности Oracle автоматически генерирует для него уникальное имя. Если индекс, с помощью которого поддерживается правило `unique` или `primary key`, создан автоматически, то его имя совпадает с именем правила целостности. Поскольку индекс не может строиться более чем по 16 полям, одно правило `unique` или `primary key` не может включать в себя более 16 столбцов. Кроме того, общая длина полей, входящих в правило `primary key` или `unique`, не может превышать 3128 символов.

Правила ссылочной целостности (`referential integrity`) позволяют добавлять или обновлять строки только в том случае, если значение столбца (набора столбцов) совпадает с каким-либо значением в связанной таблице. Примером могут служить таблицы, описывающие сотрудников и отделы: значение отдела, в котором работает сотрудник, должно представлять собой номер существующего отдела. В этом случае таблица отделов называется родительской, а таблица сотрудников — дочерней.

Oracle поддерживает три варианта действий, которые производятся над дочерней таблицей при изменении данных родительской таблицы — “`update, delete restrict`”, “`delete cascade`” и “`set null`”.

В случае “`update, delete restrict`” запрещается удалять или изменять ключевое значение в родительской таблице, если в дочерней таблице есть строки, ссылающиеся на изменяемую строку. В случае “`delete cascade`” при удалении строки из родительской таблицы автоматически будут удалены все строки дочерней таблицы, ссылавшиеся на удаленную строку, т. е., в нашем примере, при удалении отдела автоматически будут удалены все сотрудники этого отдела. В случае “`set null`” при удалении строки из родительской таблицы во всех строках дочерней таблицы, ссылавшихся на нее, значение поля-ключа будет установлено в `null`.

В Oracle ссылочная целостность определяется при помощи правила `foreign key`. Оно описывается следующим образом:

`foreign key (<внешний ключ>) references <родительская таблица> (<ключ>) [on delete (cascade | set null)]`

Внешний ключ — это поле или набор полей, перечисленных через запятую, количество и типы которых совпадают с количеством и типом полей первичного ключа в родительской таблице. Поле, объявленное как `foreign key`, может иметь неопределенное значение (`null`), если только для него явно не задано правило `not null`. По умолчанию правило `foreign key` запрещает удаление строк, на которые есть ссылки, из родительской таблицы, т. е. реализует вариант “update, delete restrict”. Если требуется создать внешний ключ с правилами “delete cascade” и “set null”, это надо указать явно во фразе `on delete`.

Для поля (или набора полей), объявленных как `foreign key`, Oracle не создает индекс автоматически. Рекомендуется делать это вручную.

Правило `foreign key` реализует отношение (dependence) много-к-одному (many-to-one) или один-к-одному (one-to-one). Так, например, отношение «сотрудник работает ровно в одном отделе, но в отделе может работать несколько сотрудников» (много-к-одному) может быть выражено введением в таблицу сотрудников (`employee`) поля `deptisn`, содержащего первичный ключ отдела в таблице `department`: `deptisn number(10) not null constraint fk_employee_dept references department(isn)`

Здесь и далее первичные ключи таблиц называются `isn` (Internal System Number). Фраза `constraint fk_employee_dept` задает имя правила целостности.

Для того, чтобы реализовать отношение многие-ко-многим, заводится промежуточная таблица, связанная отношениями много-к-одному с двумя исходными. Пусть, например, есть таблицы `employee` (сотрудники) и `project` (проекты), и один сотрудник может участвовать в нескольких проектах. Тогда, чтобы связать их отношением многие-ко-многим, создается вспомогательная таблица.

1.2. Представления

Представление — это сохраненный запрос к БД. Результатом выполнения запроса является отношение, что позволяет использовать представления в командах `insert`, `update`, `delete` и `select` так же, как и таблицы. При изменении данных представления изменяются данные в таблицах, входящих в это представление, а при использовании представления в команде выборки данных Oracle подставляет его текст в

запрос и затем разбирает и выполняет запрос целиком. В представлении нельзя определять правила целостности, однако можно определять достаточно сложную логику работы при изменении данных представления — с помощью триггеров `instead of`.

Представления используются для того, чтобы скрыть физическую структуру данных, обеспечив большую гибкость приложения, чтобы упростить запросы или чтобы ограничить доступ к данным. Представление не требует места в БД, кроме места на хранение своего определения в словаре данных.

При использовании представления Oracle использует национальные настройки текущей сессии, если эти настройки не заданы явно в SQL-запросе, образующем представление.

Создается представление командой `create view`:

```
create [or replace] [force] view <имя> [(<имя столбца>
{,<имя столбца>})] as <запрос> [with (read only | check option
[constraint <имя правила>])]
```

Если запрос некорректен (содержит ошибку или пользователь, создающий представление, не имеет привилегий на обращение к указанным в запросе объектам), представление не будет создано. Однако если указать параметр `force`, то представление будет создано в любом случае, а попытка обратиться к нему приведет к ошибке:

```
ORA-04063: view "<имя>" has errors
```

Представление может становиться ошибочным и в результате изменения структуры объектов, к которым оно обращается.

Столбцы представления получают те имена, которые возвращает SQL-запрос. Если же эти имена не устраивают программиста, то можно явно перечислить имена столбцов после имени представления в скобках. Количество имен должно в точности соответствовать количеству столбцов, возвращаемых запросом.

Фраза `with read only` говорит о том, что представление можно использовать только на чтение. Фраза `with check option` заставляет Oracle при выполнении изменения данных представления командами `insert` и `update` проверять новые значения на соответствие условиям запроса. Если запрос содержит подзапросы или изменение данных выполняется триггером `instead of`, то проверка работать не будет.

Уничтожается представление командой `drop`:

```
drop view <имя>
```

1.3. Индексы

Индексы — дополнительные структуры данных, связанные с таблицами. Индексы предназначены для ускорения поиска в таблицах, а в некоторых случаях данные могут читаться прямо из индексов, без чтения таблиц. Например, все данные индексных таблиц хранятся непосредственно в индексах, т. е. эти таблицы не имеют сегмента данных.

Индекс строится по одному или нескольким полям; в индекс может входить до 16 полей. Порядок перечисления полей в составном индексе имеет значение. Если значение всех полей, по которым построен индекс, в некоторой строке не определено (равно `null`), то такая строка в индекс не входит.

Oracle допускает создание уникальных индексов, т. е. индексов, в котором каждый набор значений индексированных полей уникален. Однако создавать такие индексы вручную не рекомендуется, т. к. уникальность значения — понятие логическое, а не физическое. Oracle автоматически создает уникальные индексы при задании правил целостности `unique` и `primary key`.

Создается индекс командой `create index`, синтаксис которой приведен ниже:

```
create [ unique | bitmap ] index <имя> on <имя таблицы>
(<имя столбца> {,<имя столбца>})
```

Ключевое слово `unique` позволяет создать уникальный индекс.

Ключевое слово `bitmap` создает индекс, построенный на битовых картах. Такие индексы могут быть значительно эффективнее и компактнее традиционных индексов, основанных на B-деревьях¹. В `bitmap`-индексах индексируются все значения, включая `null`. Составной `bitmap`-индекс может быть использован для поиска по любому из полей, входящих в индекс, при этом порядок перечисления полей в составном индексе не имеет значения. Однако, у `bitmap`-индексов есть ряд недостатков, существенно ограничивающих их применение. Во-первых, они эффективны только при небольшом количестве различных значений столбца. Индексировать, например, даты или суммы проводок следует традиционными индексами. Во-вторых, `bitmap`-индексы теряют эффективность при частом обновлении данных в таблице.

Индекс в любой момент может быть уничтожен — это не повлияет на работоспособность приложений, хотя может существенно их замедлить.

Уничтожается индекс командой `drop index`: `drop index <имя>`

1.4. Последовательности

Последовательности — объекты, генерирующие неповторяющиеся номера. Эти объекты полезны для генерации первичных ключей.

Создаются последовательности командой `create sequence`:

```
create sequence <имя> [start with <число>] [increment by
<число>] [maxvalue <число>] [minvalue <число>] [cycle | nocycle]
[cache <число> | nocache] [order | noorder]
```

Значения `start with` и `increment by` задают начальное значение последовательности и шаг, с которым будет увеличиваться значение. Задание начального значения и шага, отличных от 1, может оказаться полезным в распределенной базе данных — если, например, на двух серверах заданы начальные значения 1 и 2 и шаг 10, то ключи, сгенерированные разными серверами, будут уникальны в пределах обоих серверов.

Oracle не следит, чтобы все значения, выбранные из последовательности, были как-то использованы. Если, например, транзакция, выбравшая очередное значение, будет отменена, то выбранное значение будет утеряно.

Параметры `minvalue` и `maxvalue` определяют минимальное и максимальное значения, которые могут быть выбраны из последовательности. Если минимальное (максимальное) значение будет достигнуто, то очередное обращение к последовательности приведет к ошибке:

```
ORA-08004: sequence <имя>.NEXTVAL exceeds MAXVALUE and
cannot be instantiated
```

Если указан параметр `cycle`, то по достижении одной из границ последовательность будет снова генерировать числа, начиная с другой границы. Параметр `cache` управляет кэшированием значений. Если этот параметр задан, то обращения к последовательности будут кэшироваться, т. е. за один раз будет выбираться сразу несколько значений. Это ускоряет работу, но приводит к потере большого количества значений, если за сессию происходит мало обращений к последовательности.

В случае, если в команде `create sequence` указано ключевое слово `order`, Oracle гарантирует, что выбранное из последовательности значение будет больше (при отрицательном шаге — меньше), чем все предыдущие. В противном случае порядок может быть нарушен, но, как правило, это неважно.

Последовательность имеет два атрибута: `nextval` — следующее уникальное значение и `currval` — последнее выбранное значение. Обратиться к атрибуту `currval`

можно только в том случае, если в текущей сессии уже было обращение к `nextval`. Таким образом, значение `currval` для каждой сессии свое.

Для обращения к последовательности (а также к любым объектам, не являющимся данными таблицы, например, функциям) применяется конструкция `select ... from dual`:

```
select seq_emp.currval from dual
ORA-08002: sequence SEQ_EMP.CURRVAL is not yet defined in
this session
select seq_emp.nextval from dual
```

1.5. Синонимы

Синоним — дополнительное имя для объекта БД — таблицы, представления, последовательности или хранимой процедуры. Синоним не требует физического пространства, кроме как для своего определения в словаре данных. Создается синоним командой

```
create synonym:
create [public] synonym <имя> for <имя объекта>
```

Уничтожается синоним командой `drop`:

```
drop synonym <имя>
```

Синоним может быть переименован командой `rename`: `rename <имя> to <новое имя>`

2. Практическая часть

2.1. Таблицы

2.1.1. Создание таблицы

1. На главной странице нажмите на иконку **Object Browser**.
2. В списке выбрать **Create**, а затем **Table**.
3. В поле **Table Name** ввести имя таблицы (*personal_info*)
4. Введите следующие имена столбцов. Выберите тип данных и не оставляйте поле **Not Null** пустым, где будут вводиться данные.

employee_id	NUMBER(6,0)	NOT NULL
birth_date	DATE	NOT NULL
social_security_id	VARCHAR2(12)	NOT NULL
marital_status	VARCHAR2(10)	
dependents_claimed	NUMBER(2,0)	
contact_name	VARCHAR2(45)	NOT NULL
contact_phone	VARCHAR2(20)	NOT NULL
contact_address	VARCHAR2(80)	NOT NULL

В итоге должно получиться следующее:

Column Name	Type	Precision	Scale	Not Null	Move
employee_id	NUMBER	6	0	<input checked="" type="checkbox"/>	▼ ▲
birth_date	DATE			<input checked="" type="checkbox"/>	▼ ▲
social_security_id	VARCHAR2		12	<input checked="" type="checkbox"/>	▼ ▲
marital_status	VARCHAR2		10	<input type="checkbox"/>	▼ ▲
dependents_claimed	NUMBER	2	0	<input type="checkbox"/>	▼ ▲
contact_name	VARCHAR2		45	<input checked="" type="checkbox"/>	▼ ▲
contact_phone	VARCHAR2		20	<input checked="" type="checkbox"/>	▼ ▲
contact_address	VARCHAR2		80	<input checked="" type="checkbox"/>	▼ ▲

5. Далее необходимо нажать **Next**.
6. На странице **Primary Key** оставить все без изменений и нажать **Next**.
7. На странице **Foreign Key** оставить все без изменений и нажать **Next**.
8. На странице **Constraints** оставить все без изменений и нажать **Finish**.
9. На странице **Create Table** можно нажать кнопку **SQL** для просмотра SQL-запросов, которые использовались для создания таблицы.
10. Далее необходимо нажать кнопку **Create** для создания таблицы.

2.1.2. Добавление столбцов в таблицу.

1. На главной странице нажмите на иконку **Object Browser**.
2. В **Object list** необходимо выбрать **Tables**. Далее выбрать имя таблицы, которую необходимо изменять (*personal_info*).
3. Затем нажать кнопку **Add Column**.

4. На странице **Add Column** введите имя добавляемого столбца (*contact_email*). В поле Nullable выбрать значение NULL, тип данных - VARCHAR2, длина – 30.

The screenshot shows the 'Add Column' dialog box. The title bar is 'Add Column' with 'Cancel' and 'Next >' buttons. The main area shows 'Schema: HR' and 'Table: PERSONAL_INFO'. Below that, there is a red asterisk followed by 'Add Column' and a text input field containing 'contact_email'. To the right of this field is a checkbox labeled 'Preserve Case'. Below the text input is a dropdown menu for 'Nullable' with the selected value 'NULL (do not require a value)'. Below that is a dropdown menu for 'Type' with the selected value 'VARCHAR2'. Below the type dropdown is a text input field for 'Length' containing '30'. Below the length field are three disabled text input fields for 'Precision', 'Scale', and another 'Scale'.

5. Далее нажать кнопку **Next**, а затем на следующей странице нажать кнопку **Finish** для добавления столбца. На последней странице так же можно посмотреть SQL-запросы, нажав на кнопку **SQL**.

2.1.3.Изменение столбца в таблице

1. На главной странице нажмите на иконку **Object Browser**
2. В **Object list** необходимо выбрать **Tables**. Далее выбрать имя таблицы, которую необходимо изменять (*personal_info*).
3. Затем нажать кнопку **Modify Column**.
4. Выбрать столбец *contact_email*. Измените длину типа данных VARCHAR2 на 40. В поле Nullable выбрать значение NOT NULL.

The screenshot shows the 'Modify Column' dialog box. The title bar is 'Modify Column' with 'Cancel' and 'Next >' buttons. Below the title bar is the text 'Use this page to increase the length of character column data types.'. The main area shows 'Schema: HR' and 'Table: PERSONAL_INFO'. Below that is a dropdown menu for 'Column' with the selected value 'CONTACT_EMAIL (VARCHAR2)'. Below the column dropdown is a dropdown menu for 'Datatype' with the selected value 'VARCHAR2'. Below the datatype dropdown is a text input field for 'Length' containing '40'. Below the length field are three disabled text input fields for 'Precision', 'Scale', and another 'Scale'. Below the scale fields is a dropdown menu for 'Nullable' with the selected value 'NOT NULL (require a value)'.

5. Далее нажать **Next**.
6. Нажать кнопку **Finish** для завершения операции.

2.1.4.Удаление столбца из таблицы

1. На главной странице нажмите на иконку **Object Browser**.
2. В **Object list** необходимо выбрать **Tables**. Далее выбрать имя таблицы, которую необходимо изменять (*personal_info*).
3. Нажмите на кнопку **Drop Column**.
4. Выберите столбец `contact_address` и нажмите **Next**.
5. Нажмите **Finish** для завершения операции

2.1.5.Создание первичного ключа

1. На главной странице нажмите на иконку **Object Browser**.
2. В **Object list** необходимо выбрать **Tables**. Далее выбрать имя таблицы, которую необходимо изменять (*personal_info*)
3. Выберите вкладку **Constraint**.
4. Нажмите на кнопку **Create**.
5. В окне **Add Constraint** в **Constraint Type** выберите тип **Primary Key** и введите следующую информацию:

Constraint Name: PERSONAL_INFO_PKEY

Constraint Type: Primary Key

Primary Key Column 1: EMPLOYEE_ID(NUMBER)

The screenshot shows the 'Add Constraint' dialog box. At the top right, there are 'Cancel' and 'Next >' buttons. The main area contains the following fields:

- Schema: HR
- Table: PERSONAL_INFO
- * Constraint Name: PERSONAL_INFO_PKEY (with a 'Preserve Case' checkbox below it)
- Constraint Type: Primary Key (dropdown menu)
- * Primary Key Column 1: EMPLOYEE_ID (NUMBER) (dropdown menu)
- Primary Key Column 2: - Select Column - (dropdown menu)
- Primary Key Column 3: - Select Column - (dropdown menu)

6. Нажмите кнопку **Next**.
7. Нажмите кнопку **Finish** для завершения операции.

2.1.6.Создание вторичного ключа

1. На главной странице нажмите на иконку **Object Browser**.
2. В **Object list** необходимо выбрать **Tables**. Далее выбрать имя таблицы, которую необходимо изменять (*personal_info*)
3. Выберите вкладку **Constraint**.

4. Нажмите на кнопку **Create**.
5. В окне **Add Constraint** в **Constraint Type** выберите тип **Foreign Key** и введите следующую информацию:

Constraint Name: PERSONAL_INFO_FKEY
 Foreign Key Column: EMPLOYEE_ID
 Reference Table Name: EMPLOYEES
 Reference Table Column List: EMPLOYEE_ID

6. Поставьте галочку на **On Delete Cascade**

The screenshot shows the 'Add Constraint' dialog box with the following fields and values:

- Schema: HR
- Table: PERSONAL_INFO
- Constraint Name: PERSONAL_INFO_FKEY
- Preserve Case:
- Constraint Type: Foreign Key
- On Delete Cascade:
- Foreign Key Column(s): EMPLOYEE_ID
- Reference Table Name: EMPLOYEES
- Reference Table Column List: EMPLOYEE_ID

7. Нажмите **Next**
8. Нажмите **Finish** для завершения операции.

2.1.7. Добавление данных в таблицу

1. На главной странице нажмите на иконку **Object Browser**.
2. В **Object list** необходимо выбрать **Tables**. Далее выбрать имя таблицы, которую необходимо изменять (*personal_info*)
3. Выберите вкладку **Data**.
4. Нажмите кнопку **Insert Row**.
5. В окне **Create Row** введите следующую информацию:

employee_id: 142
 birth_date: 01-SEP-65
 social_security_id: 555-11-4444
 marital_status: Married
 dependents_claimed: 4
 contact_name: Marilyn Davies
 contact_phone: 15552229999
 contact_email: marilyn.davies@mycompany.com

Create Row	
<input type="button" value="Cancel"/> <input type="button" value="Create"/> <input type="button" value="Create and Create Another"/>	
Table:	PERSONAL_INFO
* Employee Id	142
* Birth Date	01-SEP-65
* Social Security Id	555-11-4444
Marital Status	Married
Dependents Claimed	4
* Contact Name	Marilyn Davies
* Contact Phone	15552229999
Contact Email	marilyn.davies@mycompany.com

6. Нажмите на кнопку **Create and Create Another** для создания новой строки в таблице и для открытия нового окна для добавления данных
7. На новой странице введите следующую информацию:
 - employee_id: 143
 - birth_date: 01-MAR-72
 - social_security_id: 555-77-4444
 - marital_status: Single
 - dependents_claimed: 1
 - contact_name: Carolyn Matos
 - contact_phone: 15553338888
 - contact_email: carolyn.matos@myinternet.com
8. Нажмите на кнопку **Create** для создания новой строки и завершения операции

2.1.8.Изменение данных в таблице

1. На главной странице нажмите на иконку **Object Browser**.
2. В **Object list** необходимо выбрать **Tables**. Далее выбрать имя таблицы, которую необходимо изменять (*personal_info*)
3. Выберите вкладку **Data**.
4. Нажмите на иконку **Edit** возле записи с employee_id = 142
5. В появившемся окне измените столбец phone_number на 15551118888
6. Нажмите на кнопку **Apply Changes** для завершения операции

2.1.9. Удаление строки данных в таблице

1. На главной странице нажмите на иконку **Object Browser**.
2. В **Object list** необходимо выбрать **Tables**. Далее выбрать имя таблицы, которую необходимо изменять (*personal_info*)
3. Выберите вкладку **Data**
4. Нажмите на иконку **Edit** возле записи с `employee_id = 143`
5. Нажмите на кнопку **Delete** для удаления этой строки из таблицы
6. Нажмите кнопку **Ok** для завершения операции.

2.1.10. Удаление таблицы

1. На главной странице нажмите на иконку **Object Browser**.
2. В **Object list** необходимо выбрать **Tables**. Далее выбрать имя таблицы, которую необходимо изменять (*personal_info*)
3. Нажмите на кнопку **Drop**
4. Нажмите на кнопку **Finish** для завершения операции

2.1.11. Добавление проверочного ограничения

1. На главной странице нажмите на иконку **Object Browser**.
2. В **Object list** необходимо выбрать **Tables**. Далее выбрать имя таблицы, которую необходимо изменять (*personal_info*)
3. Перейдите на вкладку **Constraints**
4. Нажмите на кнопку **Create**
5. На странице **Add Constraint** введите следующие данные:
Constraint Name: PERSONAL_INFO_CHECK_CON
Constraint Type: Check
Constraint on Column: DEPENDENTS_CLAIMED(NUMBER)
Constraint Expression: > 0

Add Constraint Cancel Next >

Schema: **HR**
Table: **PERSONAL_INFO**

* Constraint Name: PERSONAL_INFO_CHECK_CON
 Preserve Case
 Constraint Type: Check
 * Constraint on Column: DEPENDENTS_CLAIMED (NUMBER)
 * Constraint Expression:
 > 0

6. Нажмите кнопку **Next**
7. Нажмите кнопку **Finish** для завершения операции

2.1.12. Добавление уникального ограничения

1. На главной странице нажмите на иконку **Object Browser**.
2. В **Object list** необходимо выбрать **Tables**. Далее выбрать имя таблицы, которую необходимо изменять (*personal_info*)
3. Перейдите на вкладку **Constraints**
4. Нажмите на кнопку **Create**
5. На странице **Add Constraint** введите следующие данные:

Constraint Name: PERSONAL_INFO_UNIQUE_CON

Constraint Type: Unique

Unique Column 1: SOCIAL_SECURITY_ID(VARCHAR2)

Add Constraint Cancel Next >

Schema: **HR**
Table: **PERSONAL_INFO**

* Constraint Name: PERSONAL_INFO_UNIQUE_CO
 Preserve Case
 Constraint Type: Unique
 * Unique column 1: SOCIAL_SECURITY_ID (VARCHAR2)
 Unique column 2: - Select Column -
 Unique column 3: - Select Column -

6. Нажмите кнопку **Next**
7. Нажмите кнопку **Finish** для завершения операции

2.1.13. Просмотр существующих ограничений

1. На главной странице нажмите на иконку **Object Browser**.
2. В **Object list** необходимо выбрать **Tables**. Далее выбрать имя таблицы, которую необходимо изменять (*personal_info*)
3. Перейдите на вкладку **Constraints**

Constraint	Type	Table	Search Condition	Delete Rule	Status
SYS_C004175	C	PERSONAL_INFO	"EMPLOYEE_ID" IS NOT NULL	-	ENABLED
SYS_C004176	C	PERSONAL_INFO	"BIRTH_DATE" IS NOT NULL	-	ENABLED
SYS_C004177	C	PERSONAL_INFO	"CONTACT_NAME" IS NOT NULL	-	ENABLED
SYS_C004178	C	PERSONAL_INFO	"CONTACT_PHONE" IS NOT NULL	-	ENABLED
PERSONAL_INFO_PKEY	P	PERSONAL_INFO	-	-	ENABLED
SYS_C004180	C	PERSONAL_INFO	"CONTACT_EMAIL" IS NOT NULL	-	ENABLED
PERSONAL_INFO_FKEY	R	PERSONAL_INFO	-	CASCADE	ENABLED
PERSONAL_INFO_UNIQUE_CON	U	PERSONAL_INFO	-	-	ENABLED
PERSONAL_INFO_CHECK_CON	C	PERSONAL_INFO	"DEPENDENTS_CLAIMED" > 0	-	ENABLED

2.1.14. Включение или выключение ограничений

1. На главной странице нажмите на иконку **Object Browser**.
2. В **Object list** необходимо выбрать **Tables**. Далее выбрать имя таблицы, которую необходимо изменять (*personal_info*)
3. Перейдите на вкладку **Constraints**
4. Нажмите на кнопку **Disable**
5. На странице **Disable Constraint** выберите ограничение, которое хотите изменять (*dependents_claimed column (PERSONAL_INFO_CHECK_CON)*)
6. Нажмите кнопку **Next**
7. Нажмите кнопку **Finish** для завершения операции

Для включения ограничения необходимо повторить те же действия, но вместо кнопки **Disable** необходимо нажать на кнопку **Enable**

2.1.15. Удаление ограничений

1. На главной странице нажмите на иконку **Object Browser**.
2. В **Object list** необходимо выбрать **Tables**. Далее выбрать имя таблицы, которую необходимо изменять (*personal_info*)
3. Перейдите на вкладку **Constraints**
4. Нажмите на кнопку **Drop**
5. На странице **Drop Constraint** выберите ограничение, которое необходимо удалить (*contact_email*)
6. Нажмите на кнопку **Next**
7. Нажмите на кнопку **Finish** для завершения операции

2.2. Индексы

2.2.1. Создание индекса

1. На главной странице нажмите на иконку **Object Browser**.
2. В выпадающем меню выберите пункт **Create**, а затем пункт **Index**
3. На странице **Create Index** в поле **Table Name** введите название таблицы (*employees*).
4. Установите **Type of Index** как **Normal** и нажмите кнопку **Next**
5. На следующей странице в поле **Index Name** введите название индекса (*EMPLOYEES_HIREDATE_IDX*).
6. Не устанавливайте галочку напротив **Preserve Case**
7. Поле **Uniqueness** оставьте без изменений (**Non Unique**).
8. В поле **Index Column 1** выберите необходимый столбец (*HIRE_DATE*) и нажмите кнопку **Next**.
9. Нажмите кнопку **Finish** для завершения операции

Create Index Cancel < Previous Next >

Schema: **HR**
Table: **EMPLOYEES**

* Index Name: EMPLOYEES_HIREDATE_IDX
 Preserve Case
 Uniqueness: Non Unique

* Index Column 1: HIRE_DATE (DATE)
 Index Column 2: - Select Column -
 Index Column 3: - Select Column -
 Index Column 4: - Select Column -

2.2.2. Просмотр индексов

1. На главной странице нажмите на иконку **Object Browser**.
2. В **Object list** необходимо выбрать **Tables**. Далее выбрать имя таблицы (employees)
3. Перейдите на вкладку **Indexes**
4. Выберите необходимый индекс для просмотра его свойств (EMP_NAME_IX)

Object Details **Statistics** SQL

Disable Drop Rebuild

Index Name	EMP_NAME_IX
Index Type	NORMAL
Table Owner	HR
Table Name	EMPLOYEES
Table Type	TABLE
Uniqueness	NONUNIQUE
Compression	DISABLED
Prefix Length	-
Tablespace Name	USERS
Status	VALID
Last Analyzed	07-NOV-05

Index Columns

COLUMN_NAME	COLUMN_EXPRESSION	COLUMN_POSITION
LAST_NAME	-	1
FIRST_NAME	-	2

1 - 2

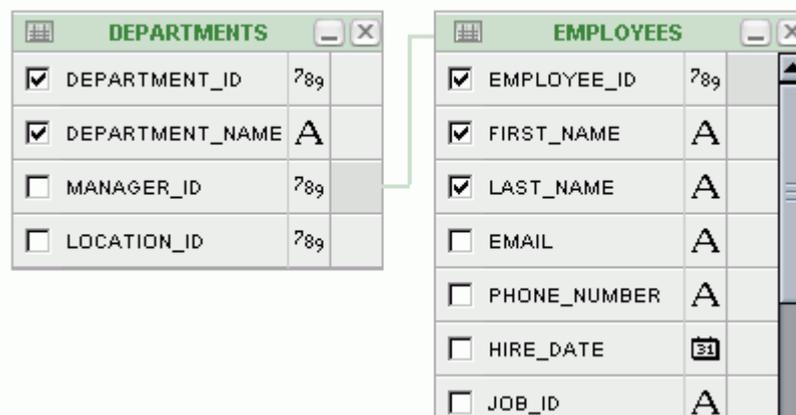
2.2.3. Удаление индекса

1. На главной странице нажмите на иконку **Object Browser**.
2. В выпадающем меню выберите пункт **Create**, а затем пункт **Index**
3. Выберите необходимый индекс (`EMPLOYEES_HIREDATE_IDX`)
4. Нажмите на кнопку **Drop** для удаления индекса
5. Нажмите на кнопку **Finish** для завершения операции

2.3. Представления

2.3.1. Создание представления

1. На главной странице нажмите на иконку **Object Browser**.
2. В выпадающем меню выберите пункт **Create**, а затем пункт **Views**
3. В поле **View Name** введите название (`my_emp_view`)
4. Нажмите на **Query Builder** для построения запроса для выборки
5. Выберите необходимую таблицу (`departments`) и в ней необходимые столбцы (`department_id` и `department_name`)
6. Выберите следующую таблицу (`employees`) и в ней необходимые столбцы (`employee_id`, `first_name`, `last_name`)
7. Выберите необходимую строку (`manager_id`) из одной таблицы (`departments`) и свяжите ее с такой же строкой в другой таблице (`employees`)



8. Нажмите на кнопку **Run** для просмотра результата выборки
9. Нажмите на кнопку **Return** для возврата в **Object Browser**
10. Нажмите на кнопку **Next**
11. Нажмите на кнопку **SQL** для просмотра SQL-запросов
12. Нажмите на кнопку **Create** для создания выборки

2.3.2. Просмотр представления

1. На главной странице нажмите на иконку **Object Browser**.
2. В Object list выберите пункт **Views**, а затем нужное представление (EMP_DETAILS_VIEW)
3. Нажмите на кнопку **SQL** для просмотра SQL-запроса.
4. Нажмите на кнопку **Data** для просмотра выборки данных

2.3.3. Удаление представления

1. На главной странице нажмите на иконку **Object Browser**.
2. В Object list выберите пункт **Views**, а затем нужное представление (my_emp_view)
3. Нажмите на кнопку **Drop**
4. Нажмите на кнопку **Finish** для завершения операции

2.4. Последовательности

2.4.1. Создание последовательности

1. На главной странице нажмите на иконку **Object Browser**.
2. В выпадающем меню выберите пункт **Create**, а затем пункт **Sequence**
3. В поле **Sequence Name** введите название последовательности (my_sequence)
4. В поле **Start With** введите **1000**
5. Другие поля оставьте без изменений и нажмите кнопку **Next**

The screenshot shows the 'Create Sequence' dialog box. At the top right, there are 'Cancel' and 'Next >' buttons. The main area contains the following fields and options:

- Schema: HR
- * Sequence Name: my_sequence
- Preserve Case
- Start With: 1000
- Minimum Value: (empty)
- Maximum Value: (empty)
- Increment By: 1
- CYCLE
- Number to Cache: (empty)
- ORDER

At the bottom, there is a descriptive text: "Database sequences are database objects used to generate integer values according to rules defined when the sequence is created. Database sequences are generally used to populate table primary keys."

6. Нажмите на кнопку **SQL** для просмотра SQL-запроса.
7. Нажмите на кнопку **Finish** для создания последовательности

Create Synonym Cancel Next >

Schema **HR**

* Synonym Name

Preserve Case

Public or Private ▾

Create synonym on:

* Schema ▲

* Object

Database Link

Use this Create Synonym screen to create a synonym for a database object to which you have access. A **synonym** is an alternative name for a table, view, sequence, procedure, stored function, package, or another synonym.

6. Нажмите на кнопку **SQL** для просмотра SQL-запроса.
7. Нажмите на кнопку **Finish** для создания синонима

2.5.2. Просмотр синонима

1. На главной странице нажмите на иконку **Object Browser**.
2. В Object list выберите пункт **Synonyms**, а затем нужный синоним для просмотра

2.5.3. Удаление синонима

1. На главной странице нажмите на иконку **Object Browser**.
2. В Object list выберите пункт **Synonyms**, а затем нужный синоним для удаления (emps)
3. Нажмите на кнопку **Drop**
4. Нажмите на кнопку **Finish** для завершения операции

Контрольные вопросы

1. Что такое таблица?
2. Что такое представление и для чего оно необходимо?
3. Что такое последовательность?
4. Что такое синоним и для чего он необходим?
5. Что такое индекс и для чего он необходим?
6. Для чего необходим первичный и вторичные ключи?
7. Какие атрибуты имеет последовательность?
8. Какие существуют правила целостности?

Рекомендуемая литература

1. Дейт К. Введение в системы баз данных, 8-е издание: Пер. с англ. – М.:Издательский дом "Вильямс". 2005.-1328 с.
2. Грабер М. Введение в SQL: Пер с англ. – М.:Изд-во 'ЛОРИ',1996. – 380с.
3. Джеймс Перри, Джеральд Пост. Введение в Oracle 10g "И.Д. Вильямс", 2006. – 700 с.
4. Гринвальд Рик, Становьяк Роберт, Додж Гери, Кляйн Дэвид, Шапиро Бен, Челья Кристофер Дж. Программирование баз данных Oracle для профессионалов.: Пер. с англ.: – М. : ООО "И.Д. Вильямс", 2007. – 784 с.
5. Кайт Томас. Oracle для профессионалов: архитектура, методика программирования и основные особенности версии 9i и 10j.: Пер с англ. – М.:Издательский дом "Вильямс". 2008. – 848 с.
6. Кевин Луни, Боб Брилла. Oracle Database 10g. Настольная книга администратора баз данных. – М.:Издательство 'Лори',2008. – 732 с.
7. Райан стивенс, Рональд Плю. SQL. Пер с англ. – М.:ЗАО 'Издательство Бином',1998.-400 с.