|  |
| --- |
| **Лабораторная работа №2**  Изучение возможностей «Oracle Application Express» (Oracle Database 11g Express Edition) |
| **Введение** Лабораторный практикум посвящен изучению инструментария для разработки информационных систем в формате веб-приложений на основе СУБД Oracle – Oracle Application EXpress (APEX). Oracle APEX позволяет создавать информационные системы с использованием веб-браузера (среда разработки представляет собой веб-приложение) и не требует от разработчика большого опыта программирования. Со стороны конечного пользователя приложения требуется только браузер и доступ к БД Oracle, на которой запущен APEX.  **Задание №1. Изучение структура APEX и начало работы**  **Задачи:**   1. Изучить структуру APEX. 2. Ознакомиться с технологией регистрации и начала работы в APEX. 3. **Структура APEX**   Oracle APEX инсталлируется в БД Oracle и представляет собой набор таблиц с данными и PL/SQL код. Запуск приложения, построенного с помощью APEX, аналогичен запуску самой среды разработки APEX и представляет собой следующий процесс: браузер посылает URL запрос, который транслируется в соответствующий PL/SQL-вызов APEX. После того, как БД выполнит PL/SQL код, результаты возвращаются обратно в браузер в виде HTML.  Возможны два варианта архитектуры Oracle APEX, зависящие от версии сервера Oracle:   * Версии до 11.1 требуют установки Oracle HTTP Server (Apache) с mod\_plsql: * Начиная с версии Oracle Database 11.1 или Oracle Database 10g Express Edition можно заменить Oracle HTTP Server (Apache) встроенным PL/SQL-шлюзом (embedded PL/SQL gateway):   Встроенный PL/SQL шлюз выполняет функции веб-сервера и обеспечивает необходимую инфраструктуру для создания динамических веб-приложений. Использование такого шлюза упрощает архитектуру и устраняет промежуточный слой.  Приложения разрабатываются в рамках рабочих областей. Рабочая область (workspace) – это виртуальная частная база данных, которая позволяет множеству пользователей работать с одной инсталляцией Oracle APEX, обеспечивая при этом приватность пользовательских объектов и приложений:  В Oracle APEX выделяются следующие роли пользователей:   * **Администраторы рабочих областей** – пользователи, которые выполняют задачи, специфичные для управления рабочими областями, – управление учетными записями пользователей, мониторинг активности, просмотр лог-файлов. * **Разработчики** – пользователи, которые создают и редактируют приложения; могут иметь отдельные рабочие области или разделять общие. * **Конечные пользователи** – не имеют привилегий разработчиков. * **Администраторы Oracle APEX** – это суперпользователи, которые управляют всеми ресурсами экземпляра Oracle APEX с помощью сервисов Application Express Administration Services.  1. **Начало работы**   Для начала работы в APEX необходимо зайти на главную страницу сервиса по адресу <http://apex.oracle.com/i/index.html>, нажать кнопку **Get Started** и зарегистрироваться через кнопку **Free Workspace.**  При регистрации нужно указать, что Вы студент и хотите ознакомиться с этим приложением, тогда на указанный адрес электронной почты придет подтверждение в виде идентификатора рабочей области (work space), логина (login или user name) и пароля (password) (поскольку диалог в среде APEX англоязычный, можно пользоваться переводчиком по адресу <http://translate.google.ru>).  Каждый следующий раз работы в данной среде после входа на главную страницу сервиса надо найти кнопку **Sign In** и ввести свои данные:  После чего появляется главная страница среды разработки APEX:  Назначение используемых в лабораторном практикуме опций (они дублируются пунктами главного меню вверху):   * **Application Builder** позволяет создавать HTML-интерфейс (или приложение) в дополнение к объектам БД - таблицам и процедурам. Каждое приложение в APEX представляет собой коллекцию страниц (page), связанных друг с другом при помощи вкладок (tabs), кнопок (buttons) или гипертекстовых ссылок (hyper text links). Страница – это основной строительный блок приложения, который содержит также элементы пользовательского интерфейса – вкладки, списки (lists), кнопки, элементы (items) и области (regions). При создании приложения и/или страницы им автоматически присваивается оригинальный номер: номер приложения – целое число, идентифицирующее приложение в системе в целом; номер страницы различает страницы внутри одного приложения и имеет структуру Р?, где ? – целое положительное число. Образование других идентификаторов можно проследить по ходу решения прикладных задач. * **SQL Workshop** предоставляет инструменты для просмотра и управления объектами БД – таблицами и процедурами:   + **Object Browser** – позволяет просматривать, создавать, изменять, читать содержимое и удалять объекты БД.   + **SQL Commands** – позволяет запускать команды SQL и PL/SQL, скрипты и сохраненные запросы.   + **SQL Scripts** – позволяет создавать, редактировать, просматривать, запускать и удалять файлы скриптов, а также загружать файлы скриптов ***из*** и ***в*** локальную файловую систему.   + **Query Builder** – позволяет искать и отфильтровывать объекты БД, выбирать объекты и столбцы, создавать связи между объектами, просматривать отформатированные результаты запросов, сохранять запросы. Графический интерфейс построителя позволяет делать это, обладая минимальными знаниями SQL.   + **Utilities** позволяет импортировать и экспортировать данные из БД, генерировать DDL-код, просматривать отчеты, восстанавливать удаленные объекты БД и выполнять другие задачи.   Правый столбик главной страницы среды APEX - вспомогательный.  В середине окна - новости, тут выводятся все сообщения, которые пользователи оставляют друг другу. Для примера можно добавить сообщение о начале работы в проекте.  Две нижних панели - статистика, показывающая, с какими приложениями и кто из пользователей работал в последнее время.  Можно пощелкать по кнопкам и посмотреть, что к чему. Например, можно сменить пароль, выбрав в меню **Administration** опцию **Change My Password**.  **Задание к работе**   1. Зарегистрироваться в системе:  * Имя рабочей области – фамилия студента в транслитерации. Транслитерация здесь и далее – это запись русскоязычного текста латинскими буквами, например, фамилия Иванов запишется как Ivanov. * Логин и пароль – произвольные. Рекомендуется имена рабочей области, логина и пароля записать и сохранить, поскольку они будут использоваться всякий раз при выполнении лабораторных работ.  1. Зайти в систему и ознакомиться со структурой главной страницы среды APEX.   **Задание №2. Создание базы данных**  **Цель работы –** получение навыков описания таблиц БД Oracle APEX и заполнения их данными.  **Задачи:**   1. Описать структуры простых таблиц БД. 2. Заполнить таблицы данными. 3. Создать связи между таблицами.   Для знакомства с технологией APEX в последующих работах решается ряд прикладных задач на БД, описывающей структуру некоторой торговой корпорации. Первоначальная схема БД, полученная с помощью СУБД Access, представлена на рис. 1.1 (некоторые таблицы будут добавляться по ходу выполнения лабораторных работ):  Безымянный.bmp  Рисунок 1.1. Логическая схема БД, представленная средствами СУБД Access  БД включает 5 основных таблиц (далее приведены описания этих таблиц в конструкторе СУБД Access):   1. Countries:      1. Departments:      1. Locations:      1. Employees:      1. Jobs:   Безымянный.bmp  Фрагменты самих таблиц, созданных в СУБД Access, показаны ниже. Они демонстрируют примеры данных, заполняющих БД:   |  |  | | --- | --- | | COUNTRIES | | | country\_id | country\_name | | 11 | Россия | | … | … |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | | LOCATIONS | | | | | | | location\_id | city | country\_id | postal\_code | street\_address | state\_province | | 123 | Калининград | 11 | 123456 | Советский пр. | Калининградская обл. | | … | … | … | … | … | … |  |  |  |  |  | | --- | --- | --- | --- | | DEPARTMENTS | | | | | department\_id | department\_name | manager\_id | location\_id | | 111 | Виктория | 1 | 123 | | … | … | … | … |  |  |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | --- | | EMPLOYEES | | | | | | | | | | employee\_id | first\_name | last\_name | hire\_date | salary | commission\_pct | phone\_number | job\_id | department\_id | | 1 | Иванов | Иван | 01.12.1989 | 12000 | 15 | 23-34-56 | 1 | 111 | | … | … | … | … | … | … | … | … | … |  |  |  | | --- | --- | | JOBS | | | job\_id | job\_title | | 1 | управляющий | | **…** | … |     Между таблицами сформированы связи (показаны на рис. 1.1 стрелками), позволяющие соединить между собой данные в единую систему. Наличие таких связей обеспечивает целостность данных в системе. Так, при попытке удалить родительскую запись выводится диагностическое сообщение о невозможности данной операции, поскольку существуют связанные записи. Таким образом, для удаления родительской записи сначала нужно удалить все подчиненные ей.  Кроме основных таблиц, в схеме представлена вспомогательная таблица Departments\_1, которая сформирована самой СУБД для представления связи между основными таблицами (формирование связей рассматривается далее в данной работе).   1. **Создание простых таблиц**   Рассмотрим для начала технологию создания простых, не связанных между собой, таблиц в APEX на примере таблиц Countries и Locations:   1. Запуск мастера создания таблиц: 2. Откройте SQL Workshop и запустите Object Browser. 3. Щелкните по кнопке **Create** в правом верхнем углу страницы. Отобразится список типов объектов БД, которые можно создать. 4. Щелкните по ссылке **Table**. Запустится мастер создания таблицы. Вверху в виде линейного графика отражаются шаги создания таблицы. Активный шаг – **Columns**. 5. В поле **Table Name** введите название таблицы - Countries: 6. Поля **Column Name** (Имя столбца), **Type** (Тип Данных), **Precision** (Точность, Максимальный размер), **Scale** (Размер, Количество знаков после запятой), **Not Null** (Обязательное) предназначены для описания столбцов создаваемой таблицы. Добавьте описания следующих столбцов:  * Столбец с названием country\_id, тип данных – Number(4), обязательный. В этом столбце будет храниться уникальный числовой идентификатор страны. * Столбец с названием country\_name, тип данных – Varchar2(20), обязательный. В этом столбце будет храниться название страны. * Нажмите **Next>**.  1. На шаге **Primary Key**:    * + В поле **Primary Key** выберите Populated from a new sequence (значения столбца первичного ключа будут браться из нового объекта-последовательности). Последовательность (Sequence) – это объект БД, который используется для генерации уникальных числовых значений.      + В полях **Primary Key Constraint Name** (Название ограничения целостности первичного ключа) и **Sequence Name** (Название последовательности) оставьте значения по умолчанию, сгенерированные APEX.      + В поле **Primary Key** (Первичный ключ) выберите столбец country\_id.      + Нажмите **Next>**. 2. Шаг **Foreign Key** (Внешний ключ) пропустите (т.е. нажмите **Next>**), т.к. в создаваемой таблице пока нет ссылок на другие таблицы. 3. На шаге **Constraints** (Ограничения целостности) добавьте уникальный ключ, определенный на столбце country\_name:    * + Выберите радиокнопку Unique (Уникальный ключ).      + В появившееся ниже поле-список **Key Column(s)** (Столбцы ключа) перенесите столбец country\_name.      + В поле **Name** должно быть указано название создаваемого ограничения целостности. Это имя должно быть уникальным в БД. Оставьте значение, предлагаемое по умолчанию.      + Нажмите кнопку **Add**, чтобы добавить ограничение в список ограничений целостности создаваемой таблицы.      + Нажмите **Next>**. 4. На шаге **Confirm** можно просмотреть сгенерированный мастером SQL-скрипт по созданию таблицы, щелкнув по ссылке . Подтвердите создание таблицы, нажав на кнопку **Create Table**.   Созданная таблица появится в списке таблиц. Для выбранной таблицы в центральной части **Object Browser** отображается ее детальное описание, организованное в виде вкладок (**Table**, **Data**, **Indexes**, **Model**, **Constraints**, …). На каждой вкладке представлены кнопки с операциями, которые можно совершать над таблицей (например, на вкладке **Table**: **Add Column**, **Modify Column** и т.д.).   1. Изучите содержимое вкладок описания созданной таблицы. 2. Найдите и изучите описание созданного объекта-последовательности **(Sequences**). 3. Аналогичным образом создайте новую таблицу, предназначенную для хранения сведений о местах размещения отделений торговой корпорации: 4. Название таблицы - Locations. 5. В таблице должны быть определены столбцы следующим образом:   location\_id - Number(4),обязательный,  city - Varchar2(40),обязательный,  country\_id - Number(4),  postal\_code – Number(6),  street\_address - Varchar2(40),  state\_province - Varchar2(40)  Обратите внимание на столбец country\_id: при формировании связей между таблицами он будет содержать ссылку на страну (на запись в соответствующей таблице). Поэтому он должен иметь тот же тип данных и размер, что и столбец первичного ключа в созданной ранее таблице countries.   1. В первичный ключ должен входить столбец location\_id, значения для него должны браться из новой последовательности, которая должна быть создана. 2. Изучите содержимое вкладок описания созданной таблицы. Обратите внимание на вкладку **Model**. На ней отображаются таблицы, на которые ссылается данная таблица, и которые ссылаются на данную таблицу. Поскольку мы создали простые, не связанные таблицы, в окне показана отдельная таблица.   Ниже в качестве справочного материала приведены некоторые соответствия между типами полей в Access и APEX:   | **Тип поля Access** | **Тип поля Oracle** | | --- | --- | | Числовой, все форматы, в том числе:  Счетчик, длинное целое,  Действительное  Денежный | NUMBER — числовые данные | | Текстовое | VARCHAR 2 — текстовые строки переменной длины до 4000байт | | Поле MEMO | LONG - текстовые строки длиной до 2 ГБ  CLOB – тексты длиной до 4 Гб | | Дата/время | DATE - даты  TIME - время |  1. **Заполнение таблиц**   Заполнение таблиц Oracle возможно двумя способами: путем импортирования данных из готовых Excel-таблиц (при этом возможно также и создание таблиц); путем добавления данных в таблицы в среде APEX.  Если данными заполняются простые таблицы, проблем нет. При заполнении связанных таблиц надо сначала заполнить родительскую таблицу, а затем - подчиненную.   * 1. **Импортирование данных из Excel-таблиц**   Для применения данной технологии структура таблицы в БД Oracle должна совпадать со структурой таблицы в Excel.  Для импортирования следует выполнить команды:  SQL Workshop→Utilities→Data Workshop→Spredsheet Data.  Далее требуется выбрать, куда (**Load to**) закачивать данные:   1. если таблица уже существует в БД Oracle, то выбирается **Existing Table**. 2. если таблица отсутствует, создают новую структуру, одновременно заполняя ее данными. Для этого выбирают опцию **New Table**.   Потом выбирается, откуда загружать данные (**Load from**). Существуют две возможности:   * из файла (**Upload file**), * копировать через буфер обмена Windows (**Copy and paste**) (рекомендуется).   Рассмотрим случай, когда данные копируются через буфер обмена:   1. выбираются соответствующие радиокнопки и нажимается **Next**>, 2. в новом окне устанавливается нужное имя таблицы, нажимается **Next>,** 3. на следующем шаге импорта мастер предлагает вставить данные из буфера в окошко, расположенное в центре страницы:  * открывается таблица в Excel, ее строки копируются в буфер обмена, * мышью щелкается в окошке на странице APEX и нажимается комбинация клавиш CTRL+V (вставить),  1. в окне появятся данные из таблицы. Рекомендуется проверить самую первую строчку - там должны быть названия столбцов. Если имена полей совпадают в обеих таблицах, нажимается **Next**>. Если предполагается переименовать поля, то исправляются их названия в окне со вставленными данными, чтобы они совпадали с именами полей в БД APEX, 2. на следующем шаге мастер выводит всю структуру таблицы и данные. Здесь нужно проверить и исправить тип, длину и имена полей. Если какие-то столбцы не нужны, можно выбрать **No** в строке **Upload**, 3. нажимается кнопка **Load Data**.   Появляется список всех загруженных таблиц, где можно просмотреть результаты импорта, например, возможные ошибки. Если загрузка данных прошла успешно, то щелкнув по имени таблицы, можно перейти на страницу **Object Browser** с открытой структурой таблицы. Чтобы увидеть свои данные, надо выбрать **Data** в списке команд над таблицей.  Теперь можно редактировать данные, щелкнув по значку в столбце **Edit** в строке, которую нужно изменить. APEX не позволяет изменять данные прямо в табличном формате, как это принято в Access или Excel. Вместо этого он открывает анкетную форму для редактируемой записи.  После внесения изменений в анкетную форму нажать кнопку **Apply Changes.**   * 1. **Добавление данных в среде APEX**   В среде APEX данные можно вводить через опции меню. Для этого сразу после входа в среду APEX выполнить команды SQL Workshop → Object Browser.  Выбрать нужную таблицу в левом столбце, выполнить команду **Data**. Нажимают кнопку **Insert Row** и в анкете вводят новые данные, после чего нажимают кнопку **Create**.  Следует отметить, что APEX-приложение, как правило, имеет специальные формы для пользователя, где последний сможет редактировать данные в таблицах, не обращаясь к среде разработки. Как и в Access, приложение более удобно для пользователя, чем средства, предназначенные для разработчиков.   1. **Создание связей между таблицами**   Из схемы БД видно, что между таблицами существуют связи, которые реализуются через соответствующие поля. Эти связи можно формировать как во время создания таблиц, так и после их создания и, возможно, заполнения данными.   * 1. **Создание связей между простыми таблицами**   Рассмотрим формирование связей после создания простых, не связанных между собой, таблиц. Для этого образуем связь подчинения Countries (страны) → Locations (местонахождение) (связующими являются поля этих таблиц сountry\_id, причем одноименное поле в таблице Locations ссылается на такое же поле в таблице Countries, показывая тем самым, в какой стране находится отделение корпорации):   * 1. Откройте вкладку **Constraints** описания созданной таблицы Locations.   2. Щелкните по кнопке **Create**. Отобразится мастер добавления ограничения целостности:      + В поле **Constraint Name** впишите название создаваемого ограничения (оно должно быть уникальным в БД) например, location\_con.      + В поле **Constraint Type** выберите Foreign Key и Disallow Delete.      + В списке **Foreign Key Column(s)** выделите столбец country\_id.      + В поле **Reference Table Name** (название родительской таблицы) выберите название созданной таблицы Countries.      + В списке **Reference Table Column** (столбцы родительской таблицы, на которые будут ссылаться столбцы внешнего ключа) выберите столбец первичного ключа родительской таблицы – country\_id.      + Нажмите **NEXT>.**   3. Подтвердите создание внешнего ключа, нажав на кнопку **Finish**. Созданное ограничение целостности отобразится в списке ограничений целостности таблицы.   4. Откройте вкладку **Model** для таблицы Locations:     Видно, что теперь таблицы связаны: таблица Locations подчиняется таблице Countries.   * 1. **Создание связанных таблиц**   Можно «закладывать» связь между таблицами уже во время их создания с помощью Object Browser. Рассмотрим эту технологию на примере тех же таблиц в предположении, что мы их создаем заново как связанные таблицы:   * 1. Создается таблица Countries по описанной в разделе «Создание простых таблиц» технологии.   2. Аналогично создается таблица Locations до шага **Foreing Key**:  1. На шаге **Foreign Key** (Внешний ключ) добавляется ограничение ссылочной целостности (внешний ключ) для столбца country\_id:    * + В секции **Add Foreign Key** (Добавить внешний ключ) в поле **Name** вписывается название создаваемого ограничения целостности, например, country\_con. Необходимо убедиться, что длина названия не превышает 30 символов (ограничение СУБД Oracle на длину идентификаторов объектов). При необходимости надо сократить название.      + Из группы радиокнопок {Disallow Delete, Cascade Delete, Set Null on Delete} оставить выбранным Disallow Delete (запрещать удаление родительской записи, если у нее есть дочерние).      + В качестве столбца, на который накладывается ограничение целостности (поле со списком **Key Column(s)**), выбрать столбец country\_id.      + В качестве родительской таблицы (на которую ссылается столбец country\_id) выбрать созданную ранее таблицу Countries.      + Для того чтобы появились поля со списками столбцов для родительской таблицы, щелкнуть по иконке  справа от поля References Table.      + В появившееся ниже поле со списком **Referenced Column(s)** добавить столбец country\_id, на который будет ссылаться столбец country\_id.      + Нажать кнопку **Add**, чтобы добавить ограничение целостности в список внешних ключей. 2. Закончить создание таблицы по описанной ранее технологии.   **Задание к работе** Разработать логическую схему БД, аналогичную рис. 1.1. Состав полей таблиц можно уточнить с преподавателем.   | Вариант | Предметная область, сущности и их атрибуты | | --- | --- | | 1 | Продукты на складе:  Продукт (наименование; количество; цена; код единицы измерения; код страны происхождения; …).  Справочник стран (код; наименование страны; …).  Справочник единиц измерения (код; наименование единицы измерения;…). |  1. Средствами APEX, следуя описанной технологии, описать таблицы и заполнить их данными. Состав полей может быть изменен по согласованию с преподавателем. В каждую таблицу включить идентификатор экземпляра сущности (поля типа \*\_id в примере). Связи подчинения между сущностями указать в таблицах с помощью идентификатора (так, например, в таблице EMPLOYEES показаны связи между подразделением и должностью через поля job\_id и department\_id). 2. Создать связи между таблицами для поддержания целостности данных. 3. Показать результаты преподавателю. |

# 