

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ
федеральное государственное бюджетное образовательное учреждение высшего образования
«УЛЬЯНОВСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»

А. В. Лебедев, О. Э. Чоракаев

УПРАВЛЕНИЕ ДАННЫМИ

Практикум

Ульяновск
УлГТУ
2018

УДК 004.65(075)
ББК 32.973я73
Л33

Рецензенты:

канд. техн. наук, главный конструктор АО «УКБП» К.В. Ларин;
д-р техн. наук, главный научный сотрудник ФНПЦ АО «НПО «Марс»
Г.П. Токмаков

*Утвержден редакционно-издательским советом университета
в качестве практикума.*

Лебедев, Анатолий Валерьевич

Л33 Управление данными : Практикум по выполнению практических и лабораторных работ / А.В. Лебедев, О.Э. Чоракаев. – Ульяновск : УлГТУ, 2018. – 99 с.

ISBN 978-5-9795-1839-8

Практикум предназначен для проведения практических и лабораторных занятий со студентами, обучающимися по программам высших учебных заведений очной и очно-заочной форм обучения по направлению 09.03.02 «Информационные системы и технологии» с учетом производственной тематики.

Работа подготовлена на кафедре «Самолетостроение».

**УДК 004.65 (075)
ББК 32.973я73**

ISBN 978-5-9795-1839-8

© Лебедев А. В.,
Чоракаев О. Э., 2018
© Оформление. УлГТУ, 2018

СОДЕРЖАНИЕ

ПРИНЯТЫЕ СОКРАЩЕНИЯ	5
ПРЕДИСЛОВИЕ	6
УСТАНОВКА ORACLE DATABASE 11G XE	7
ПРАКТИЧЕСКИЕ РАБОТЫ	13
Практическая работа № 1	
Настройка и использование утилиты SQL*Plus.....	13
Практическая работа № 2	
Команды описания данных (DDL)	18
Практическая работа № 3	
Команды манипулирования данными (DML)	23
Практическая работа № 4	
Настройка и использование Data Modeler	25
Практическая работа № 5	
Разработка ER-диаграмм	31
Практическая работа № 6	
Формирование таблиц БД в среде Data modeler	35
Практическая работа № 7	
Загрузка сформированных таблиц в БД Oracle	41
 ЛАБОРАТОРНЫЕ РАБОТЫ	 46
Лабораторная работа № 1	
Создание таблиц средствами SQL*Plus. Оператор CreateTable	46
Лабораторная работа № 2	
Ввод данных в таблицы средствами SQL*plus.	
Оператор Insert	49
Лабораторная работа № 3	
Виртуальное объединение таблиц баз данных	
(создание представлений)	51
Лабораторная работа № 4	
Построение запросов в SQL*Plus. Оператор Select	55

Лабораторная работа № 5	
Дополнительные команды управления данными	59
Лабораторная работа № 6	
Способы подключения к БД Oracle	62
Лабораторная работа № 7	
Создание приложений	70
Лабораторная работа № 8	
ORACLE SQL Developer для работы с запросами Oracle:	
преимущества и недостатки, автоматическое построение	
запросов в Query Builder	76
Лабораторная работа № 9	
Проектирование таблиц	80
ЗАКЛЮЧЕНИЕ	86
КОНТРОЛЬНЫЕ ВОПРОСЫ	87
ТЕСТОВЫЕ ВОПРОСЫ ДЛЯ КОНТРОЛЯ СФОРМИРОВАН-	
НОСТИ КОМПЕТЕНЦИЙ РАБОТЫ С БАЗАМИ ДАННЫХ	88
ГЛОССАРИЙ	97
БИБЛИОГРАФИЧЕСКИЙ СПИСОК	99

ПРИНЯТЫЕ СОКРАЩЕНИЯ

БД — база данных

БиД — банк данных

ДЛМ — даталогическая модель предметной области

ПО — предметная область

РБД — реляционная база данных

СМ — сетевая модель данных

СУБТ — система управления базами данных

ЯОД — язык описания данных

3GL — Third-Generation Language (язык третьего поколения)

4GL — Fourth-Generation Language (язык четвертого поколения)

ANSI — American National Standards Institute (Американский институт национальных стандартов)

API — Application Programming Interface (интерфейс прикладного программирования)

DDL — Data Definition Language (язык определения данных)

DQL — Data Query Language (язык запросов данных)

QBE — Query-By-Example (язык запросов по образцу)

SADT — Structured Analysis and Design Technique (метод структурного анализа и проектирования)

SGML — Standard Generalized Markup Language (стандартный общий язык)

SQL — Structured Query Language (структурированный язык запросов)

UML — Unified Modeling Language (унифицированный язык моделирования)

URL — Uniform Resource Locator (определитель местонахождения информации)

XE — Express Edition, облегченная редакция Oracle Database

ПРЕДИСЛОВИЕ

Источником значительной части информации являются базы данных, в которых содержатся различного рода сведения. Структура данных может мало интересовать конечного пользователя, поскольку для него более важны время получения информации, ее актуальность и доступный объем. Создание базы данных, отвечающей всем требованиям пользователей, становится задачей проектировщика, который должен обладать теоретическими знаниями и практическими навыками в области информационных технологий.

Данное пособие предполагает, что студенты владеют современными языками программирования и, используя системы управления базами данных, такие, как Access, Oracle, SQL могут создавать и поддерживать информационные базы.

Целью данного издания является приобретение студентами следующих профессиональных компетенций ФГОС:

- способность проводить техническое проектирование (ПК-2);
- способность проводить рабочее проектирование (ПК-3);
- способность проводить выбор исходных данных для проектирования (ПК-4);
- способность к проектированию базовых и прикладных информационных технологий (ПК-11);
- способность разрабатывать средства реализации информационных технологий (ПК-12);
- способность проводить сбор, анализ научно-технической информации, отечественного и зарубежного опыта по тематике исследования (ПК-22);
- способность составлять инструкции по эксплуатации информационных систем (ПК-33).

УСТАНОВКА ORACLE DATABASE 11GXE

Oracle Database Express Edition (XE) является базой данных начального уровня, занимающей небольшой объем. Она основана на коде старших редакций, бесплатна для разработки, развертывания и распространения; проста в установке и администрировании. Oracle Database XE может быть установлена на компьютер с любым количеством процессоров, любого размера (одна база данных на компьютер), но редакция XE позволяет хранить не более 4 Гб пользовательских данных, использовать до 1 Гб оперативной памяти и использовать только один процессор на машине. В случае необходимости существует возможность апгрейда редакции XE до одной из полных версий.


Рассмотрим установку для ОС Windows 7. Для установки Oracle Database XE нам потребуются администраторские права.

Установка Oracle 11g XE

1. Скачиваем файл OracleXE112_Win32.zip с официального сайта Oracle «<http://www.oracle.com/>». Скачивание Oracle Database XE возможно только зарегистрированным пользователям. Если у вас нет учетной записи Oracle, вам придется зарегистрироваться на сайте (рис. 1).

2. Распаковываем архив OracleXE112_Win32. В папке, в которую мы распаковали архив, находим папку DISK1. В ней находится инсталлятор setup.exe. Запускаем его.

Создание учетной записи Oracle

Уже есть учетная запись? [Вход в систему](#) 

Для создания учетной записи Oracle заполните все обязательные поля и подпишитесь на группы почтовой рассылки.

Эту учетную запись можно использовать для получения доступа к различным службам и приложениям Oracle, в том числе к службе поддержки My Oracle Support, форумам Oracle Technology Network, Oracle Store, Oracle Cloud, Oracle University и Oracle PartnerNetwork, а также для регистрации на мероприятия Oracle Events.

* Указывает на обязательное поле

Сведения Об Учетной Записи

* Электронная почта

Ваш адрес электронной почты одновременно является вашим именем пользователя.

* Пароль

Пароли должны быть не короче 8 символов и должны содержать буквы обоих регистров и не менее одной цифры.

* Подтверждение пароля

Личные Данные

* Имя

Отчество

* Фамилия

* Наименование должности

* Страна



* Штат/область



* Город

* Компания

* Адрес 1

Адрес 2

* Почтовый индекс

* Рабочий телефон

☐ Да, я хочу стать участником [Oracle Technology Network](#)
☐ Да, я хочу стать участником [сообщества Oracle C-Central](#)
☐ Да, я хочу получать сообщения о продуктах, услугах и мероприятиях Oracle

Чтобы отказаться от рассылки, щелкните ссылку «Не отправлять сообщения электронной почты» в конце страницы.

+ Центр Подписки

Нажимая «Создать», вы признаете и согласны с тем, что использование веб-сайтов Oracle осуществляется в соответствии с [условиями использования Oracle.com](#) и политикой Oracle в отношении конфиденциальности.

Рис. 1. Создание учетной записи и регистрация на сайте Oracle

3. После запуска инсталлятора появится окно приветствия Oracle. Нажимаем «Next» (рис. 2).

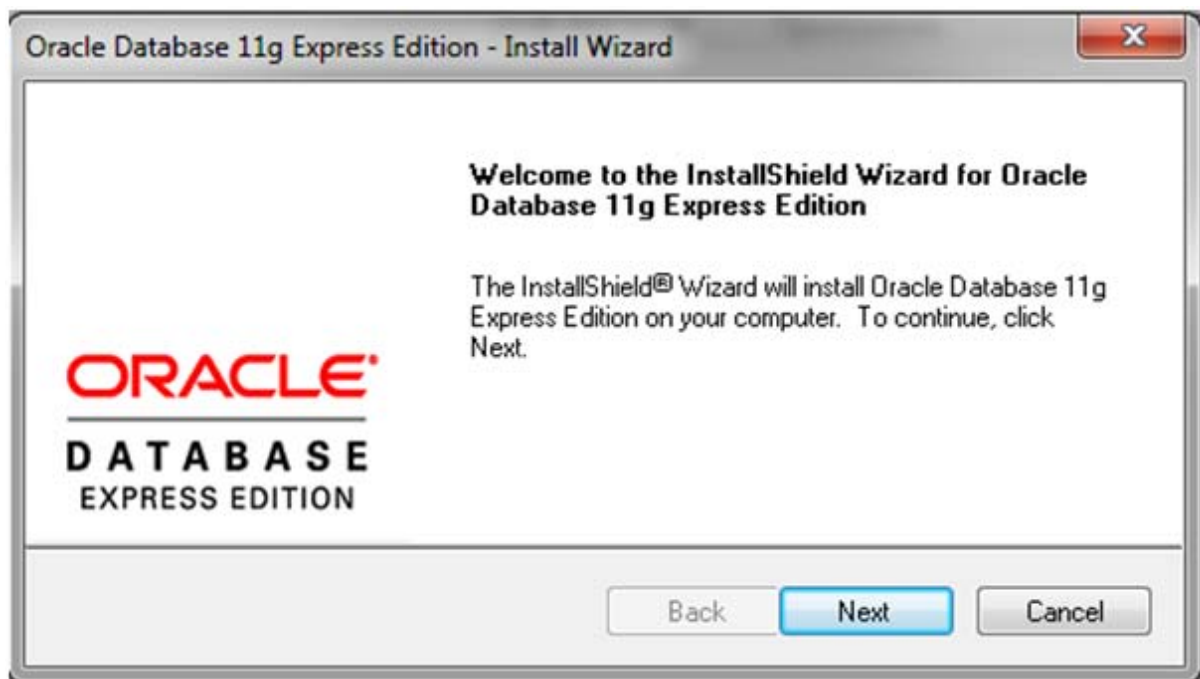


Рис. 2. Стартовое окно приветствия Oracle Database 11g XE

4. Следующее окно – лицензионное соглашение. Читаем, принимаем, нажимаем «Next» (рис. 3).

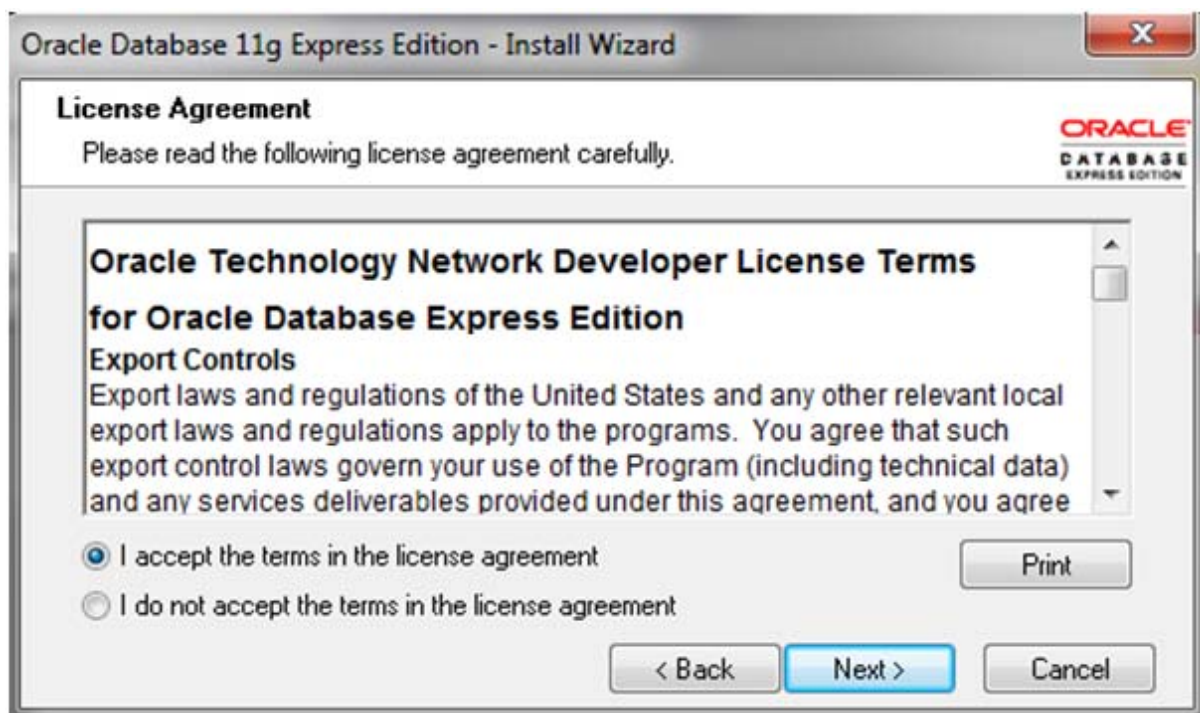


Рис. 3. Лицензионное соглашение Oracle Database 11g XE

5. Выбираем путь установки, по умолчанию «C:\oraclexe\». Для примера изменим наш путь на «D:\Program Files\». Нажимаем «Next» (рис.4).

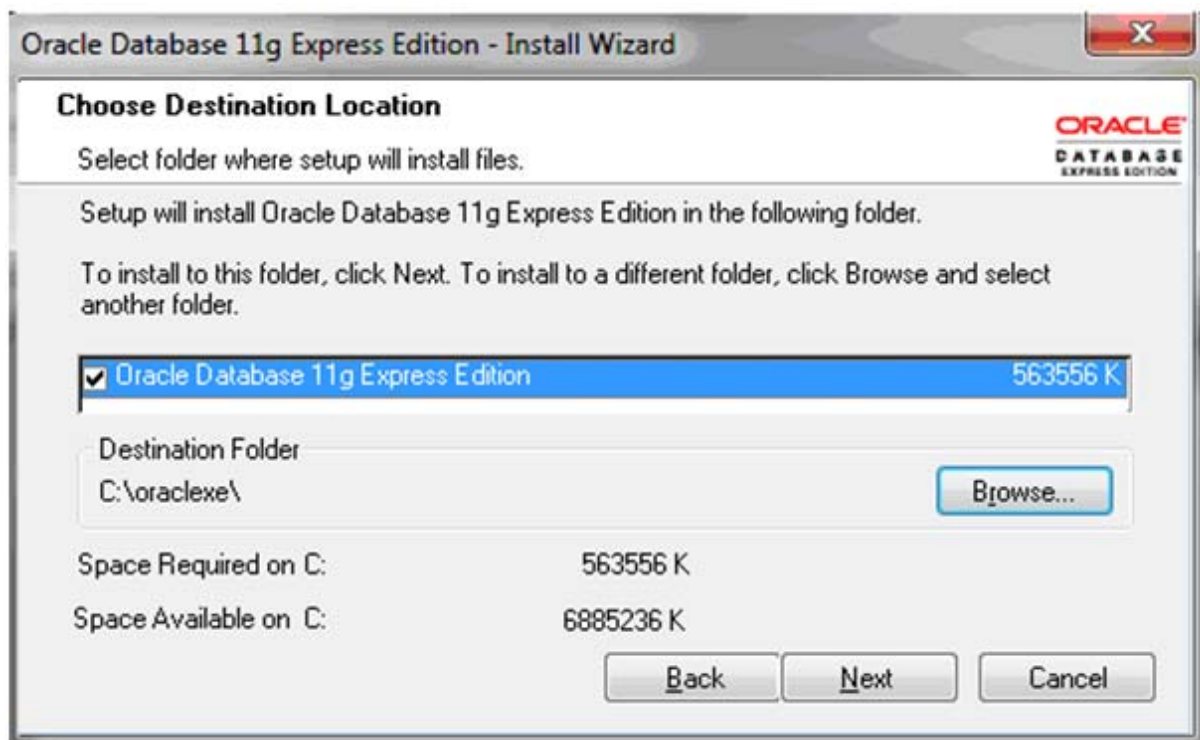


Рис. 4. Выбор пути установки Oracle Database 11g XE

6. Вводим пароль для учетных записей «SYS» и «SYSTEM». Эти учетные записи создаются автоматически при создании новой базы данных. Они предназначены для администрирования, создания таблиц и представления и т. д. Нажимаем «Next» (рис. 5).

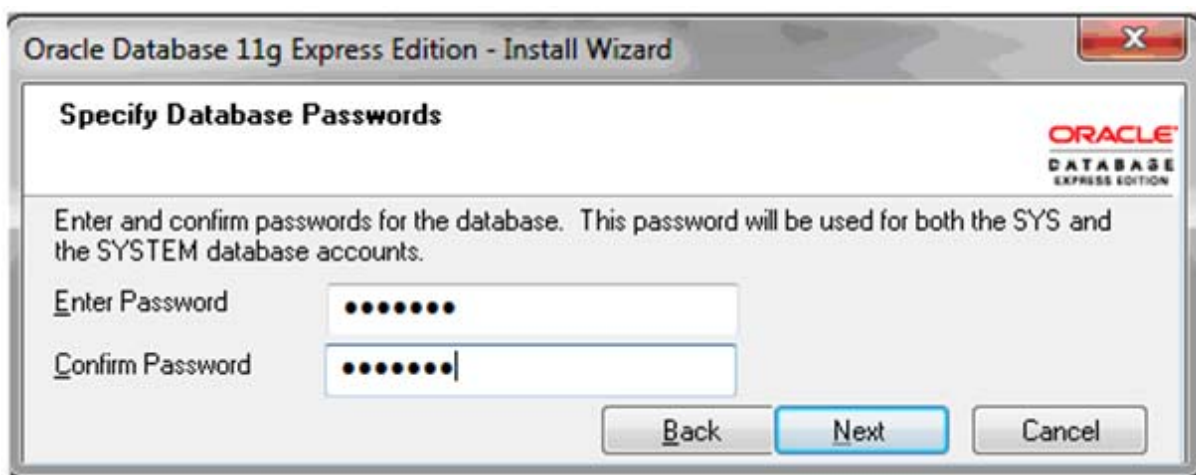


Рис. 5. Создание пароля для системной учетной записи Oracle Database 11g

7. Проверяем параметры установки.

Папки: Destination Folder – папка установки Oracle; Oracle Home – папка домашней страницы Oracle; Oracle Base – основная папка Oracle.

Порты: Oracle Database Listener – порт для удаленного доступа к Oracle; Oracle Services for Microsoft Transaction Server – порт для работы Oracle с сервером транзакций от Microsoft; Oracle HTTP Listener – порт для работы Oracle по протоколу HTTP. Если на вашем компьютере есть Firewall, то эти порты необходимо открыть.

Нажимаем «Install» (рис. 6).

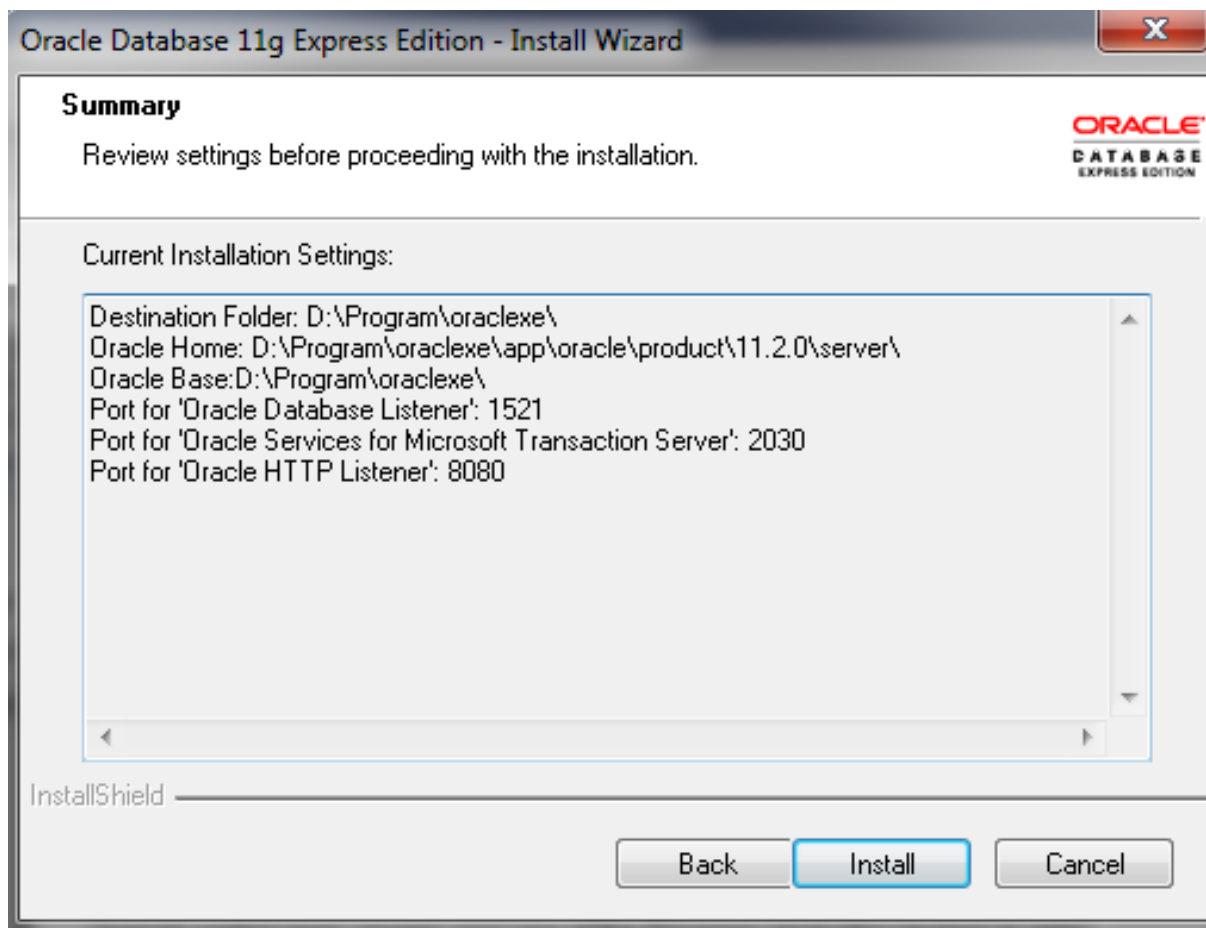


Рис. 6. Окно с итоговыми параметрами установки Oracle Database 11g XE

8. Ждем окончания установки (рис. 7).

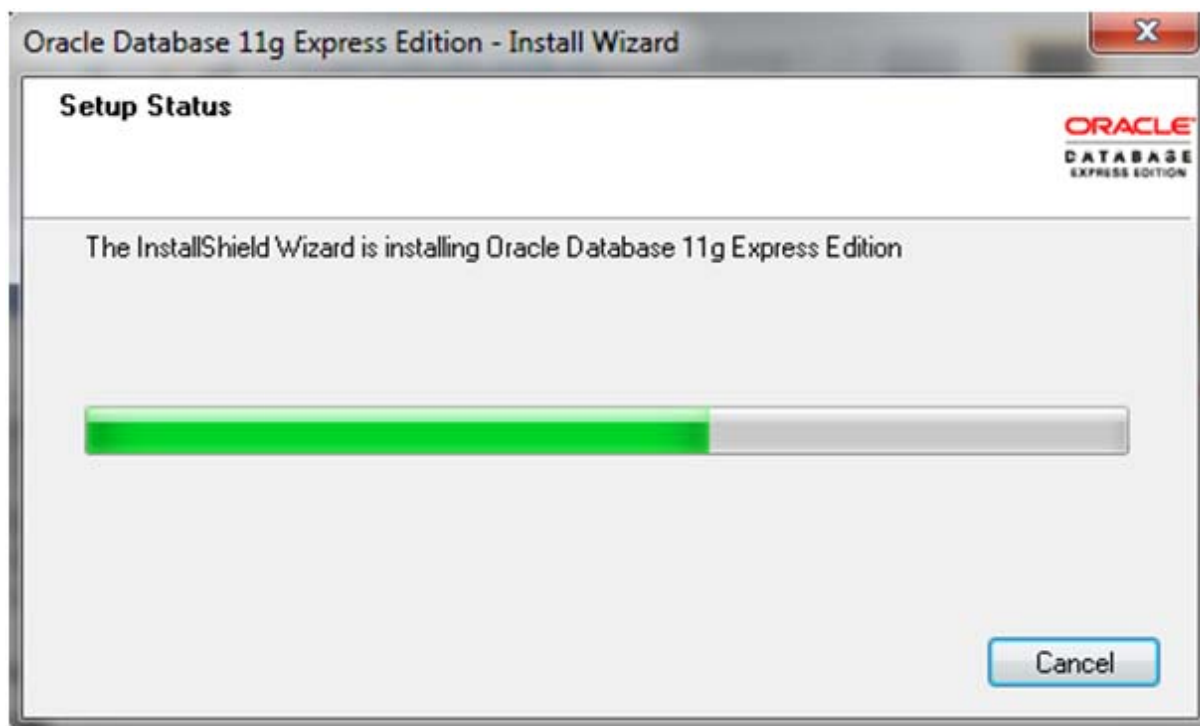


Рис. 7. Прогресс установки Oracle Database XE

9. По окончании установки нажимаем «Finish» (рис. 8).

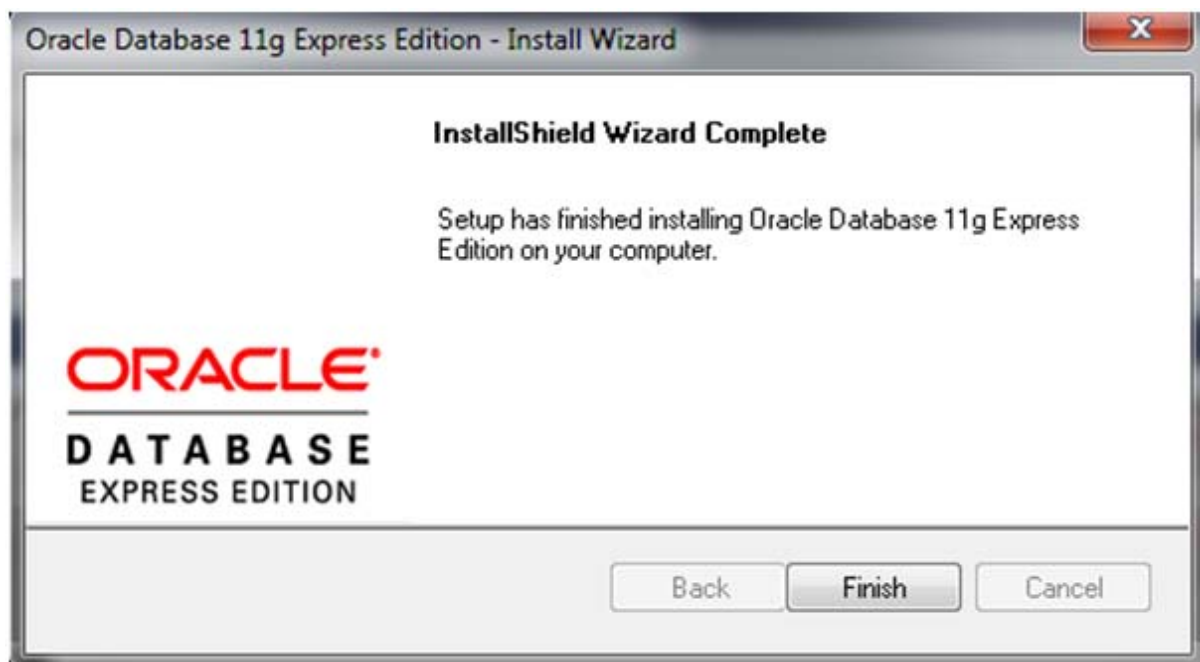


Рис. 8. Завершение установки Oracle Database 11g XE

Oracle Database 11g Express Edition установлена и готова к работе.

ПРАКТИЧЕСКИЕ РАБОТЫ

Практическая работа №1

Настройка и использование утилиты SQL*Plus

Фирма Oracle предоставляет два основных инструмента для общения с базами данных (БД) посредством SQL: SQL*Plus (входит в состав установщика Oracle) и SQL Developer (можно скачать с сайта «<http://oracle.com>»).

SQL*Plus – программа из обычного комплекта программного обеспечения (ПО) Oracle для диалогового общения с БД путем ввода пользователем текстов на SQL и PL/SQL и предъявления на экране компьютера результата, полученного от системы управления базой данных (СУБД). В отличие от SQL Developer она не имеет графического интерфейса и фактически представляет собой командную строку. SQL*Plus является минимумом для работы с Oracle. Несмотря на это, SQL*Plus все же обладает некоторыми достоинствами, такими как простота использования, минимальные требования к аппаратным и программным ресурсам, максимум функциональных возможностей и т. д.

Использование утилиты SQL*Plus

1. Запустить SQL*Plus можно при помощи меню (рис. 9): «Пуск->Oracle Database 11g Express Edition->Run SQL Command Line»

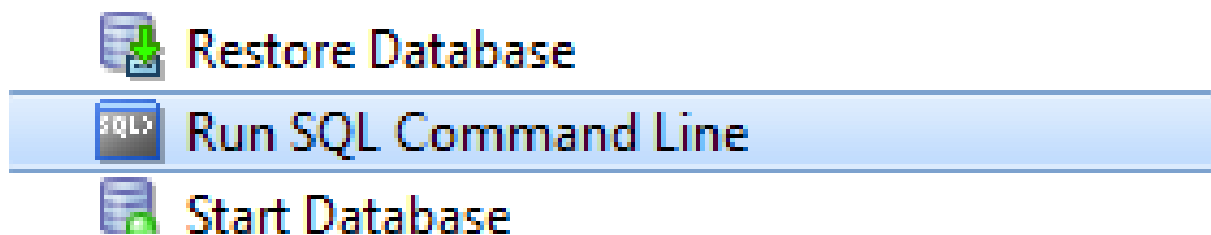


Рис. 9. Запуск утилиты SQL*Plus при помощи меню Пуск

2. После запуска появится окно работы с утилитой (рис. 10).

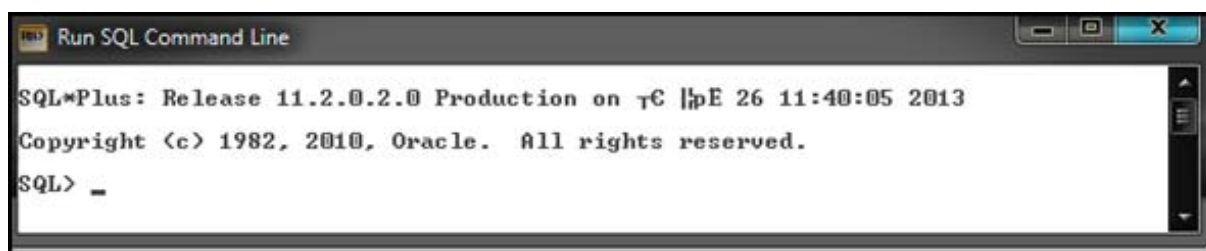


Рис. 10. Стартовое окно работы с утилитой SQL*Plus

3. SQL*Plus обрабатывает тексты на трех языках: SQL, PL/SQL и собственном. Во всех случаях регистр набора не имеет значения (за исключением пароля). Собственные команды SQL*Plus служат для настроек работы этой программы, установления форматов и выполнения некоторых действий.

Например, «**DESCRIBE**» — выдача на экран общего описания структуры таблиц, представлений данных, типов объектов или пакетов; «**SET**» – установка режимов выдачи данных на экран и т. д. (подробное описание команд можно найти на сайте Oracle). Символ «;» используется в качестве признака окончания ввода команды SQL (они могут быть многострочными), но в некоторых случаях для этого используется «/» в первой позиции новой строки.

4. В Oracle для учебных целей существуют предустановленные пользователи, но в целях безопасности они заблокированы. Мы будем использовать предустановленного пользователя «HR». Для начала соединимся с Oracle под пользователем «SYS». Для этого вводим команду:

		SQL
1	CONNECT SYS AS SYSDBA	

Далее необходимо ввести пароль, который вы указывали при установке Oracle (рис. 11).

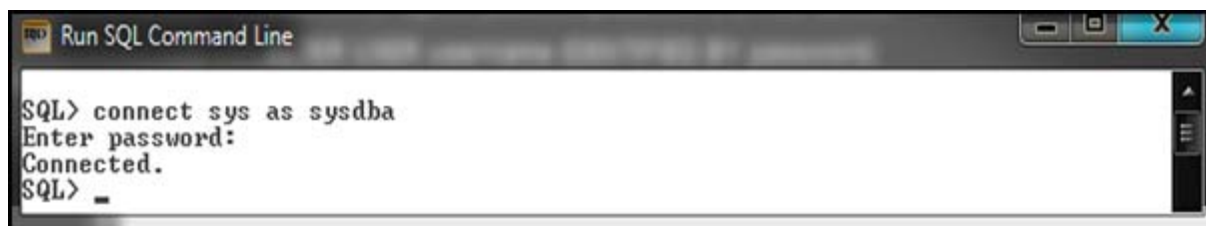


Рис. 11. Подключение к Oracle под системным пользователем

5. Следующим шагом для разблокировки пользователя «HR» вводим команду

		SQL
1	ALTER USER HR BY IDENTIFIED HR ACCOUNT UNLOCK	

Этой командой мы разблокировали пользователя «HR» и присвоили ему пароль «HR». Затем мы закончим сеанс пользователя «SYS» командой «DISCONNECT» и войдем под «HR», как на предыдущем шаге (рис. 12).

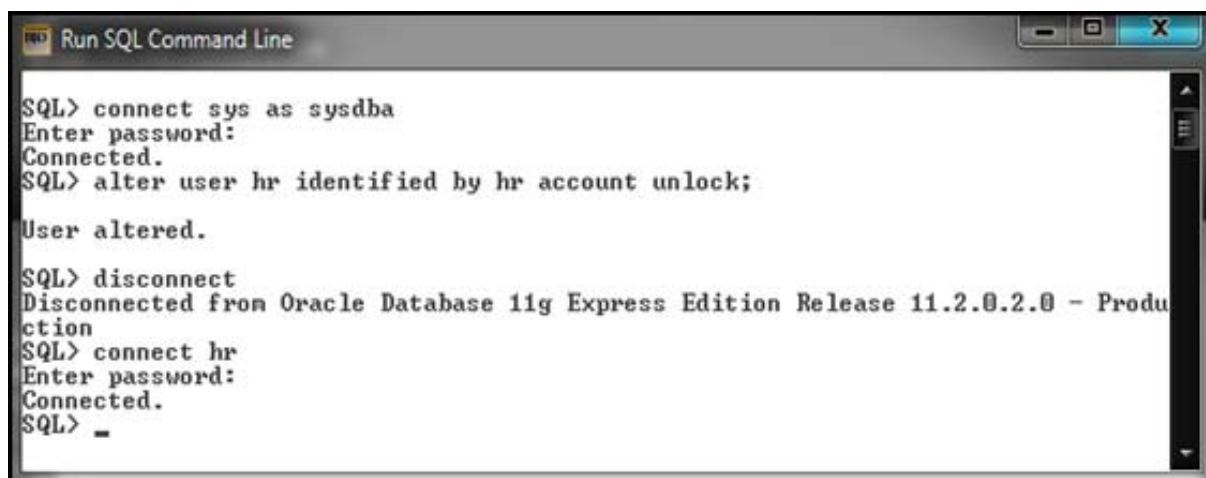


Рис. 12. Разблокировка предустановленного пользователя «HR»

6. В схеме «HR» имеется учебная БД о сотрудниках виртуальной фирмы. Всего в этой схеме имеется семь таблиц. Чтобы это проверить, мы напишем небольшой запрос (рис. 13):

		SQL
1	SELECT COUNT (*) FROM USER TABLES	

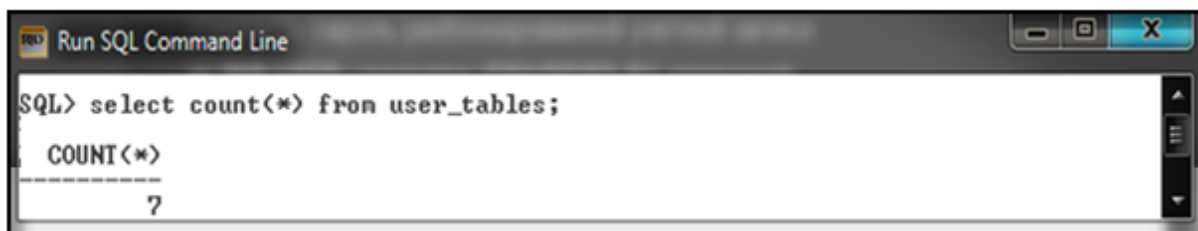


Рис. 13. Проверка количества таблиц в БД

7. Для примера выберем из таблицы «DEPARTMENTS» все записи (рис. 14). Для этого введем:

		SQL
1	SELECT * FROM DEPARTMENTS	

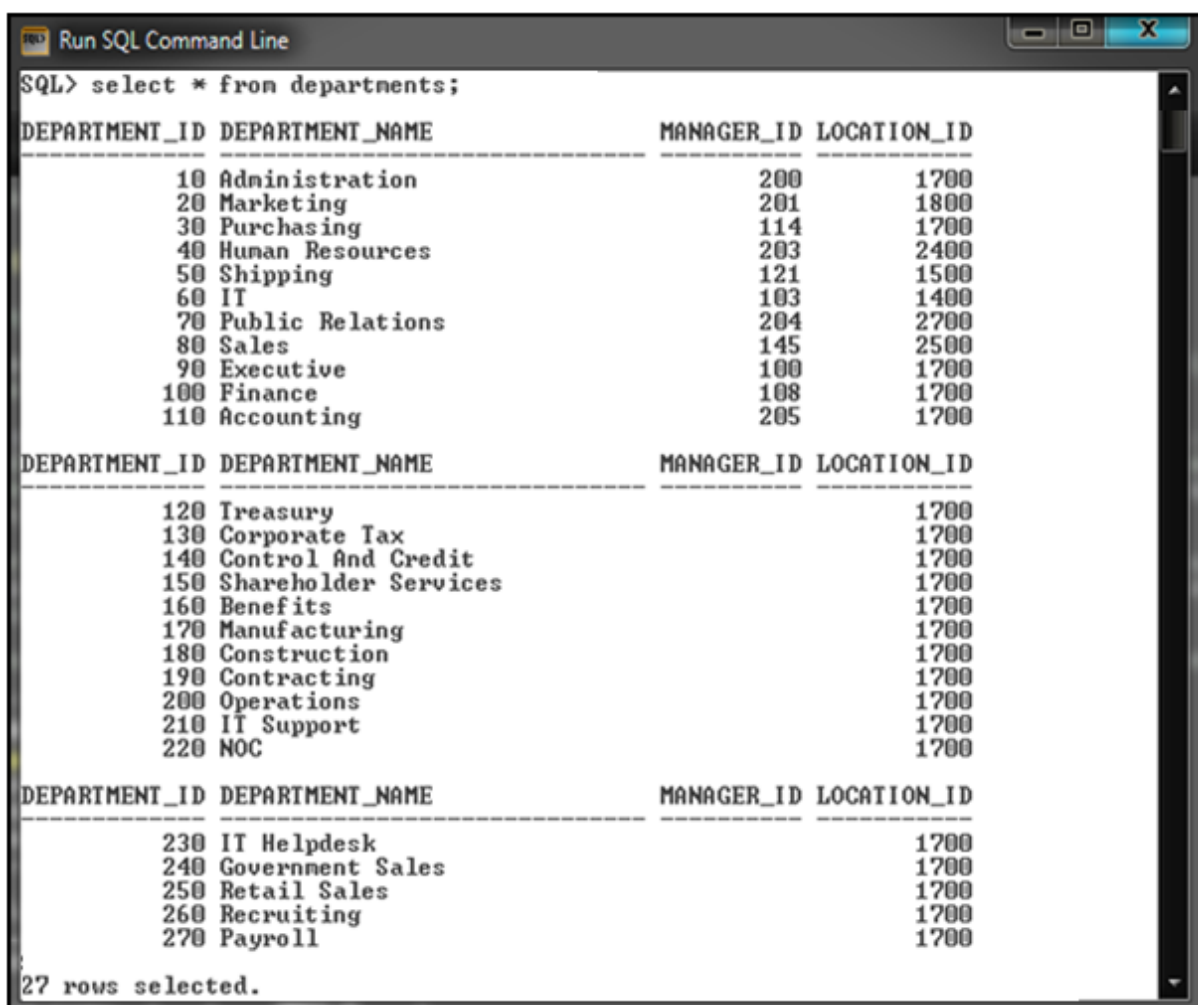


Рис. 14. Вывод записей из таблицы «DEPARTMENTS»

8. Также SQL*Plus позволяет выполнять файлы скриптов SQL и DDL с расширениями .sql и .ddl. Создадим в папке «C:\TEMP\» файл

«EXAMPLE.SQL» и запишем в него все написанные ранее команды. Затем в SQL*Plus напишем:

		SQL
1	@C:\TEMP\EXAMPLE.SQL	

Получим тот же результат (рис. 15).

```

SQL> @C:\TEMP\EXAMPLE.SQL;
Connected.

COUNT(*)
-----
      8

DEPARTMENT_ID DEPARTMENT_NAME          MANAGER_ID LOCATION_ID
-----
      10 Administration             200         1700
      20 Marketing                 201         1800
      30 Purchasing                 114         1700
      40 Human Resources            203         2400
      50 Shipping                   121         1500
      60 IT                         103         1400
      70 Public Relations           204         2700
      80 Sales                      145         2500
      90 Executive                  100         1700
     100 Finance                    108         1700
     110 Accounting                 205         1700

DEPARTMENT_ID DEPARTMENT_NAME          MANAGER_ID LOCATION_ID
-----
     120 Treasury                    1700
     130 Corporate Tax               1700
     140 Control And Credit          1700
     150 Shareholder Services        1700
     160 Benefits                    1700
     170 Manufacturing               1700
     180 Construction                1700
     190 Contracting                 1700
     200 Operations                  1700
     210 IT Support                  1700
     220 NOC                        1700

DEPARTMENT_ID DEPARTMENT_NAME          MANAGER_ID LOCATION_ID
-----
     230 IT Helpdesk                  1700
     240 Government Sales            1700
     250 Retail Sales                1700
     260 Recruiting                  1700
     270 Payroll                     1700

27 rows selected.
  
```

Рис. 15. Вывод записей с помощью файла скрипта «EXAMPLE.SQL»

Для разъединения с Oracle выполним команду «DISCONNECT».

Практическая работа №2

Команды описания данных (DDL)

Data Definition Language (язык описания данных) – это семейство языков, используемых для описания структуры баз данных. В настоящее время наиболее популярным языком DDL является SQL (Structured Query Language - структурированный язык запросов), используемый для получения и манипулирования данными в СУБД, и сочетающий в себе элементы DDL, DML, и DCL. Инструкции DDL предназначены для создания, изменения и удаления объектов баз данных. Функции языка DDL определяются первым словом в предложении. В случае с SQL это «**CREATE**», «**ALTER**» и «**DROP**». Назначение этих команд:

CREATE – Создание новых объектов: таблиц, полей, индексов.

DROP – Удаление объектов.

ALTER – Изменение объектов.

Использование команд DDL

1. Для начала создадим в Oracle нового пользователя «**INSTITUTE**». В Oracle для каждого созданного пользователя автоматически создается схема данных. Запустим SQL*Plus и выполним соединение под администратором «**SYS**». Для этого вводим команду «**CONNECT SYS AS SYSDBA**». Далее необходимо ввести пароль, который вы указывали при установке Oracle (ранее рис.11).

Пуск ⇒ Oracle Database 11g Express Edition ⇒ Run SQL Command Line

2. Для создания пользователя «**INSTITUTE**» вводим команду

		SQL
1		CREATE USER INSTITUTE BY IDENTIFIED INSTITUTE DEFAULT TABLESPACE USERS TEMPORARY TABLESPACE TEMP QUOTA UNLIMITED ON USERS .

Таким образом, мы создадим пользователя «**INSTITUTE**», назначим ему пароль «**INSTITUTE**» и укажем основное табличное пространство «**USERS**», временное «**TEMP**» и разрешаем использование табличного пространства «**USERS**» (рис. 16).

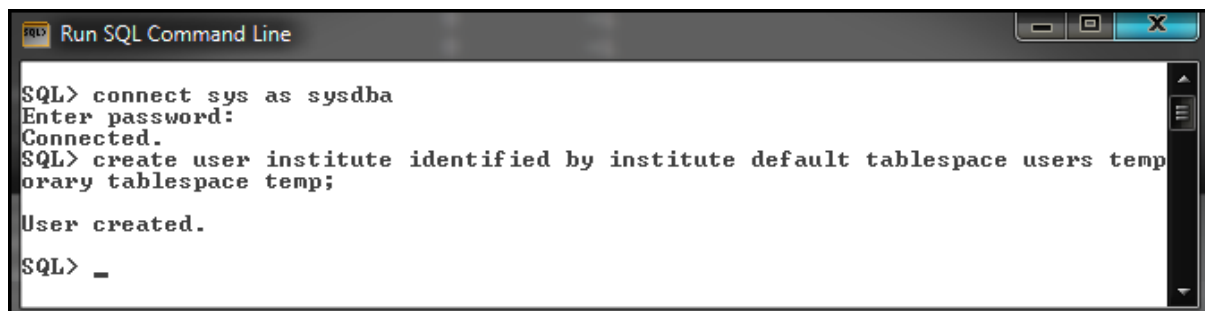


Рис. 16. Создание пользователя «INSTITUTE» в табличном пространстве

3. Теперь нашего пользователя необходимо наделить правами доступа. Для этого используется команда «**GRANT**». Введем последовательно:

	SQL
1	GRANT DELETE ANY TABLE TO INSTITUTE;
2	GRANT CREATE TABLE TO INSTITUTE;
3	GRANT CREATE VIEW TO INSTITUTE;
4	GRANT CREATE SEQUENCE TO INSTITUTE;
5	GRANT CREATE TYPE TO INSTITUTE;
6	GRANT CREATE SESSION TO INSTITUTE;
7	GRANT DROP ANY TABLE TO INSTITUTE;
8	GRANT DROP ANY TYPE TO INSTITUTE;
9	GRANT DROP ANY VIEW TO INSTITUTE;
10	GRANT ALTER ANY TABLE TO INSTITUTE;
11	GRANT ALTER ANY TYPE TO INSTITUTE;

Либо мы можем создать файл «CREATEUSER.SQL» с теми же записями (рис. 17).

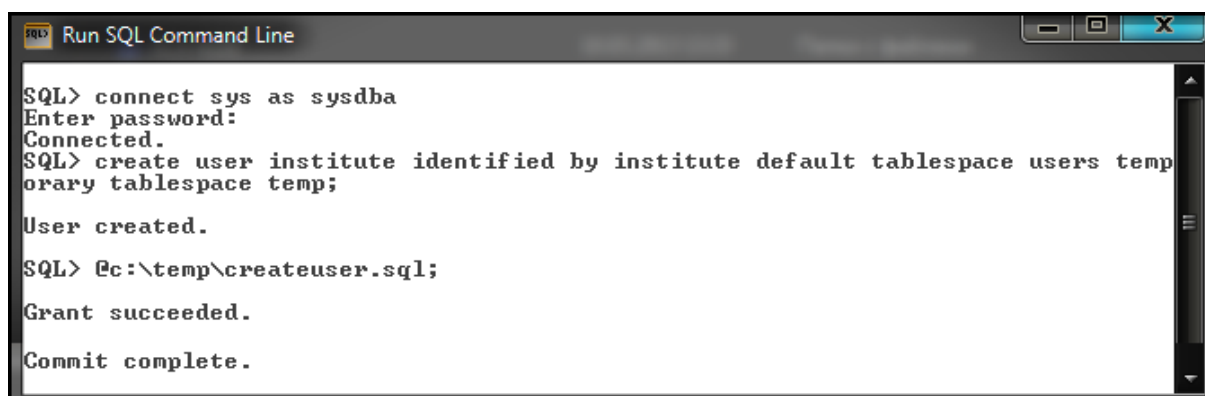
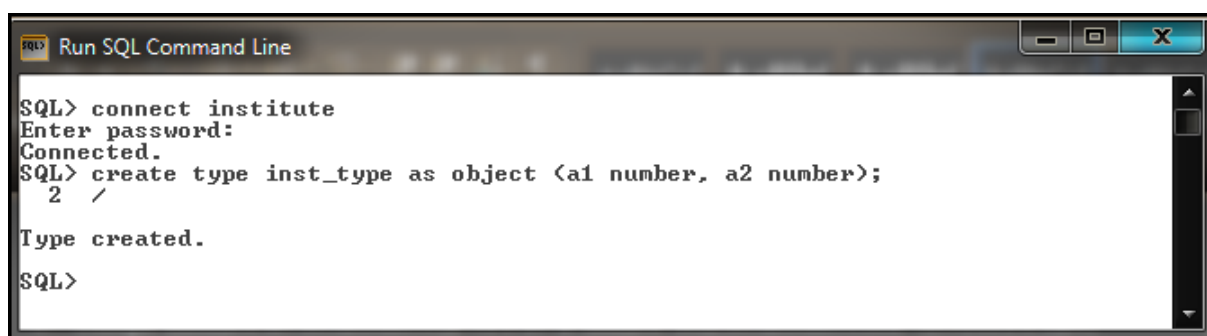


Рис. 17. Наделение пользователя «INSTITUTE» правами доступа

4. Затем выйдем из пользователя «**SYS**» и зайдем под пользователем «**INSTITUTE**» так, как было показано ранее. Теперь давайте создадим тип данных «**INST_TYPE**» для изучения возможностей языка DDL. Oracle позволяет создавать пользовательские типы данных для хранения сложных объектов. Создать новый пользовательский тип можно командой «**CREATE TYPE**»(рис.18):

SQL	
1	CREATE TYPE INST_TYPE AS OBJECT (a1 NUMBER , a2 NUMBER)



```
SQL> connect institute
Enter password:
Connected.
SQL> create type inst_type as object (a1 number, a2 number);
2 /
Type created.
SQL>
```

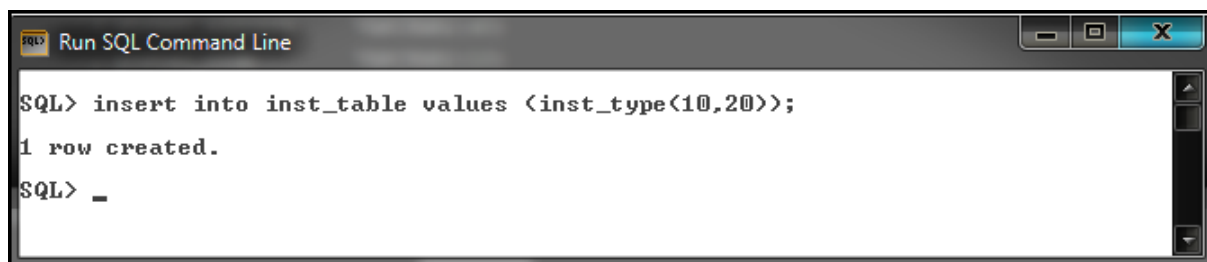
Рис. 18. Создание нового пользовательского типа данных «**INST_TYPE**»

5. Создадим небольшую таблицу для проверки созданного типа. Для этого используем команду языка DDL «**CREATE TABLE**»:

SQL	
1	CREATE TABLE INST_TABLE (COL1 INST_TYPE);

Внесем в нее запись (рис. 19)

SQL	
1	INSERT INTO INST_TABLE VALUES (INST_TYPE(10, 20)).

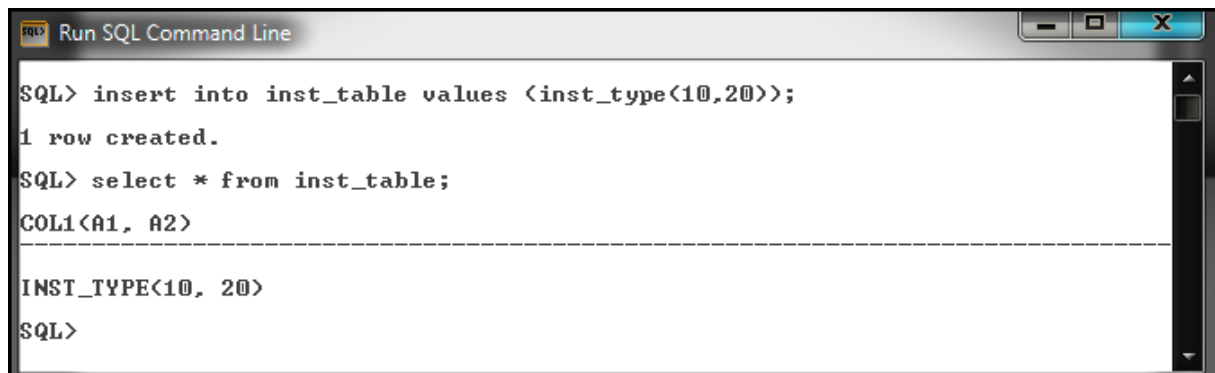


```
SQL> insert into inst_table values (inst_type(10,20));
1 row created.
SQL> _
```

Рис. 19. Проверка использования созданного типа «**INST_TYPE**»

6. Посмотрим результат наших действий (рис. 20):

		SQL
1	SELECT *	FROM INST_TABLE.



```
SQL> insert into inst_table values (inst_type(10,20));
1 row created.
SQL> select * from inst_table;
COL1(A1, A2)
-----
INST_TYPE(10, 20)
SQL>
```

Рис. 20. Проверка результата добавления новой записи

7. Изменим структуру таблицы командой «ALTER TABLE», добавив столбец (рис.21):

		SQL
1	ALTER TABLE INST_TABLE	ADD (COL2 VARCHAR (10)).



```
SQL> alter table inst_table add (col2 varchar(10));
Table altered.
SQL> _
```

Рис. 21. Изменение структуры таблицы командой «**ALTER TABLE**»

8. Командами «**DROP INST_TABLE**» и «**DROP INST_TYPE**» можно удалить нашу таблицу и тип данных.

9. Есть еще одна команда, полезная при проверке созданных таблиц. Это команда **DESCRIBE** (описать) – показать описание столбцов.

Синтаксис команды:

		SQL
1	DESC[RIBE]	[польз.] объект [имя_связи_с_БД]

Команда **DESCRIBE** показывает описания столбцов для указанной таблицы, представления данных, или синонимы.

Команда **польз** задает пользователя, который владеет объектом. Если **польз** опущен, SQL предполагает, что вы владелец данного объекта.

Команда **объект** задает таблицу, обзор или синоним (чье описание вы хотите вывести).

Команда **имя_связи_с_БД** состоит из имени связи БД, где существует данный объект. За информацией о связях с БД обращайтесь к «Руководство пользователя по языку SQL» и «Справочное руководство по языку SQL». Для каждого столбца описание содержит:

- имя столбца
- разрешено ли нулевое значение (**NULL** или **NOT NULL**)
- тип данных
- точность столбца (и количество знаков после запятой, для числовых столбцов)

Пример: Чтобы получить описание таблицы EMP, введите:

		SQL
1	SQL> DESCRIBE EMP	

DESCRIBE покажет следующую информацию:

Имя	Пусто?	Тип данных
-----	-----	-----
EMPNO	NOT NULL	NUMBER(4)
ENAME		CHAR(10)
JOB		CHAR(9)
MGR		NUMBER(4)
		DATE
HIREDATE		
SAL		NUMBER(7,2)
COMM		NUMBER(7,2)
DEPTNO		NUMBER(2)

10. Для продолжения работы с SQL PLUS создайте себя как пользователя базы данных (используйте свою фамилию на латинице) и присвойте необходимые права доступа к БД командой **GRANT**.

Практическая работа №3

Команды манипулирования данными (DML)

Data Manipulation Language (язык манипулирования данными) – это семейство языков, используемых для изменения данных в БД. Так же как и DDL, наиболее популярным языком DML сейчас является SQL (Structured Query Language – структурированный язык запросов), в который DML входит как составная часть. Инструкции DML позволяют выбирать данные из таблиц БД, а также добавлять, удалять и изменять их. В случае с SQL DML это: **SELECT**, **INSERT**, **UPDATE**, **DELETE**.

Использование команд DML

1. Соединимся с Oracle под пользователем «INSTITUTE». Для этого вводим команду «**CONNECT INSTITUTE**». Далее необходимо ввести пароль «INSTITUTE» (рис. 22).

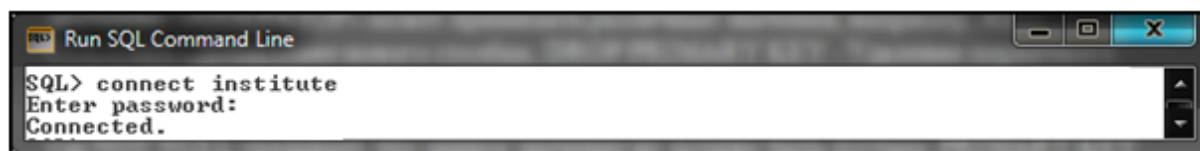


Рис. 22. Соединение с Oracle под пользователем «INSTITUTE»

2. Вставим еще одну запись в созданную на занятии по изучению DDL (практическая работа 2) таблицу (рис. 23):

		SQL
1	INSERT INTO INST_TABLE VALUES (INST_TYPE (15, 25) 'FIRST')	

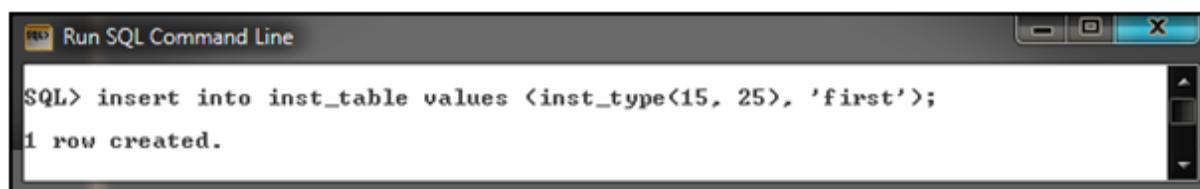


Рис. 23. Добавление еще одной записи в таблицу «INST_TABLE»

Теперь для добавленного столбца «COL2» внесем данные, используя команду «**UPDATE**»

		SQL
1	UPDATE INST_TABLE SET COL2='INST'	

Данные будут внесены в обе записи, так как мы не использовали условие «**WHERE**» (рис. 24).



Рис. 24. Команда обновление данных в таблице командой «**UPDATE**»

3. Проверим наши изменения командой DML «**SELECT**» (рис. 25):

		SQL
1	SELECT * FROM INST_TABLE.	



Рис. 25. Проверка изменений в таблице «INST_TABLE»

4. Удалим одну запись из таблицы. Для этого используем команду «**DELETE**» (рис.26):

		SQL
1	DELETE FROM INST_TABLE WHERE COL1=INST_TYPE(10,20)	

Проверим наши изменения:

		SQL
1	SELECT * FROM INST_TABLE	

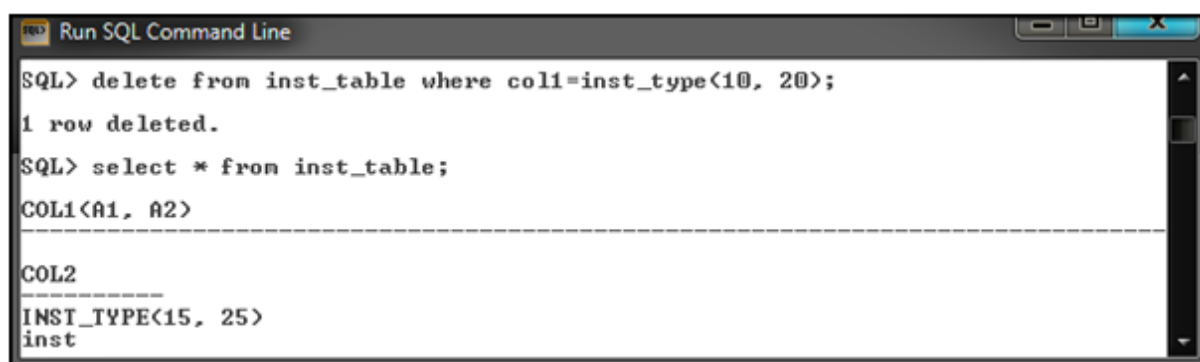


Рис. 26. Удаление одной записи из таблицы «INST_TABLE»

Практическая работа №4

Настройка и использование Data Modeler

Oracle SQL Developer Data Modeler – это утилита для многоуровневого проектирования и построения концептуальных диаграмм сущностей и связей и их превращения в реляционные модели. Data Modeler предлагает множество функциональных возможностей для моделирования данных и баз данных, в том числе и визуальное моделирование взаимосвязей между сущностями, которое хорошо интегрируется с Oracle SQL Developer – графическим инструментом Oracle для разработки баз данных, чтобы предоставить разработчикам возможность открывать и просматривать созданные ранее структуры. Oracle Data Modeler доступен для скачивания с официального сайта Oracle.

Установка Oracle Data Modeler

Oracle SQL Developer Data Modeler для Windows можно скачать в двух вариантах: архив, включающий в себя JRE (Java Runtime Environment), и не включающий, если у вас на компьютере уже установлена Java-машина (рис. 27).

1. Скачиваем архив, включающий JRE с официального сайта Oracle «<http://www.oracle.com/>». Скачать Data Modeler могут только зарегистрированные пользователи. Если у вас нет учетной записи Oracle, вам придется зарегистрироваться на сайте (ранее на рис. 1).

2. Распакуем архив и запустим файл `datamodeler.exe`.

3. После запуска перед вами появится главное окно программы. Data Modeler позволяет создавать логические модели (Logical Models), реляционные модели (Relational Models), физическую реализацию (Physical Models), новые типы данных (Data Types Models), домены (Domains) и многое другое.

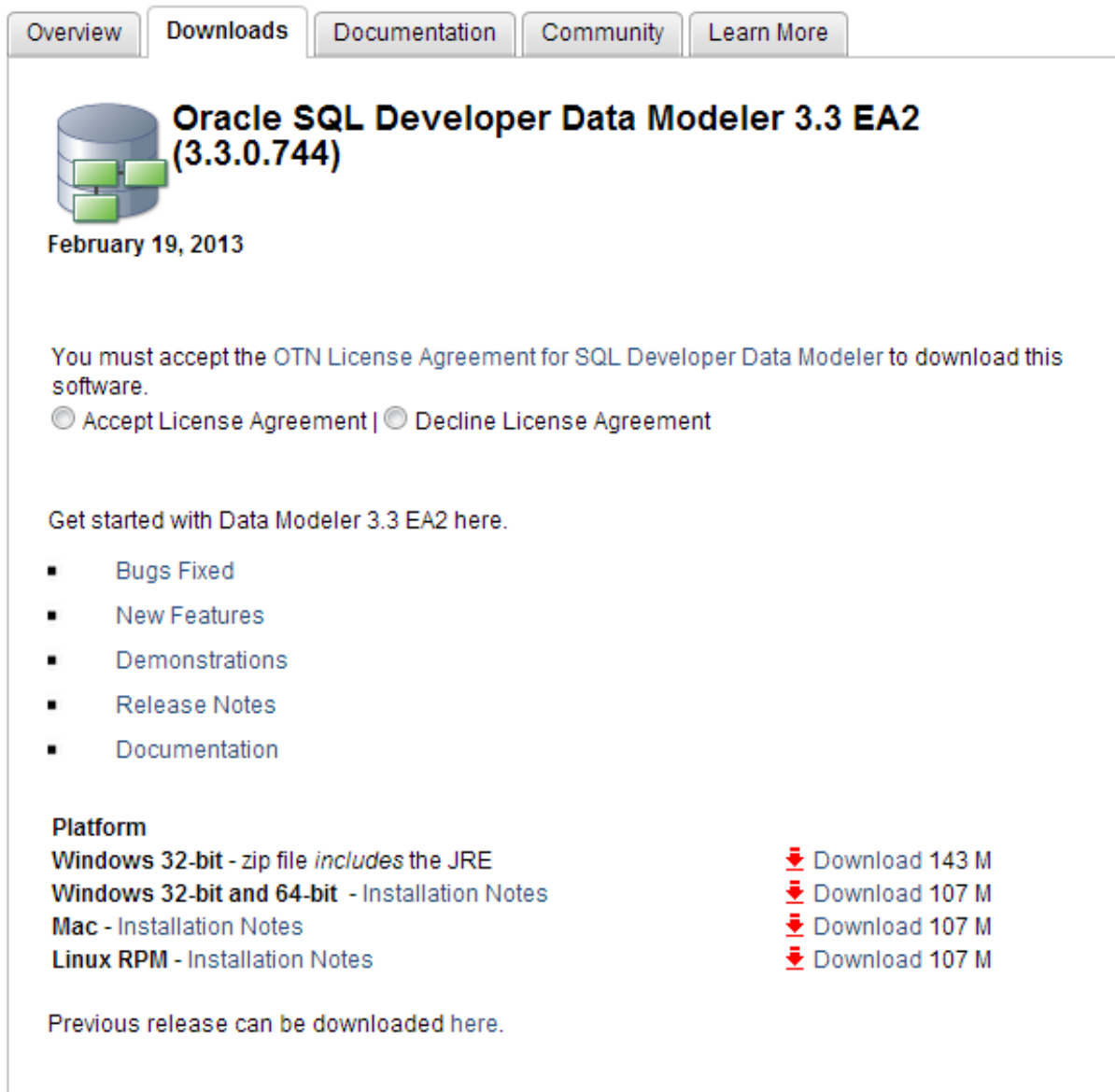


Рис. 27. Страница загрузки Oracle SQL Developer Data Modeler

Каждая логическая модель имеет одну или несколько реализаций в виде реляционных моделей, которые в свою очередь могут воплотиться в одну или несколько физических моделей. В главном окне слева находится «Browser», справа рабочее графическое окно, снизу окно сделанных изменений «Messages - Log» (рис. 28).

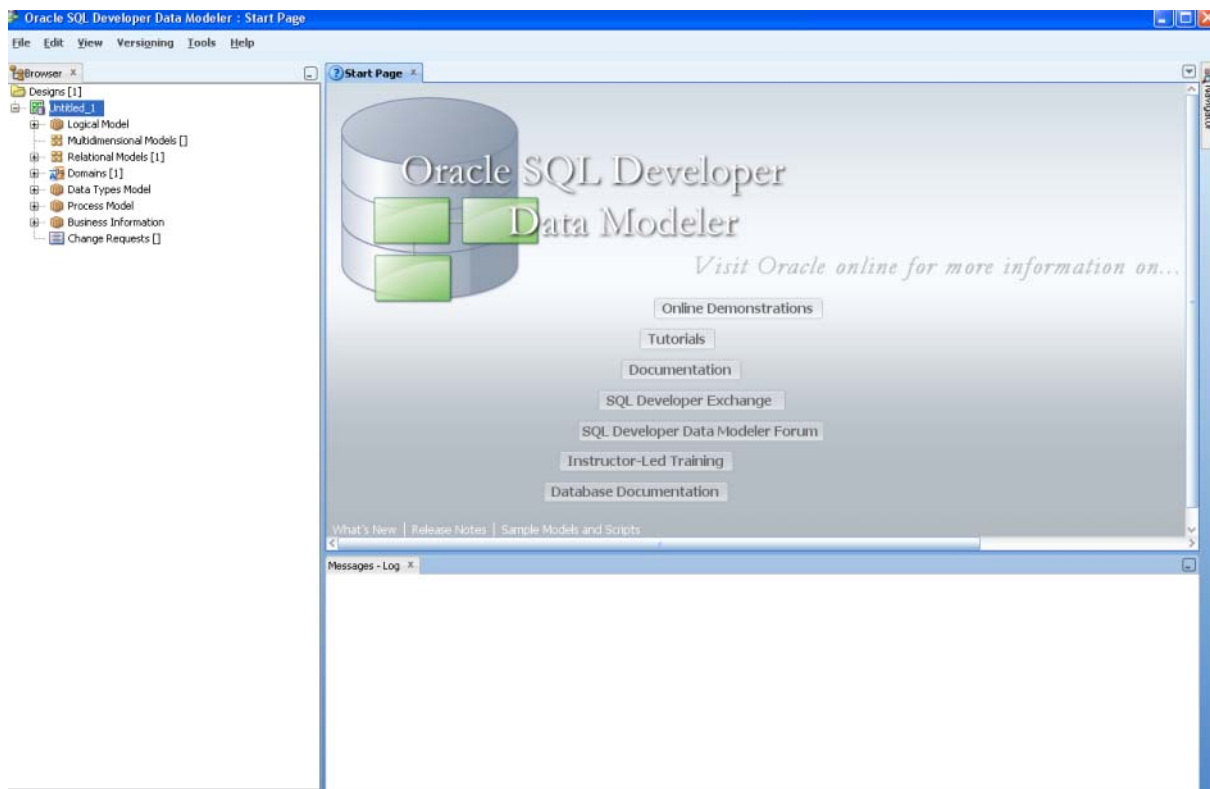


Рис. 28. Главное окно Oracle SQL Developer Data Modeler

4. Для создания логической модели щелкните правой кнопкой мыши по пиктограмме «Logical Models» в левом окне «Browser» и нажмите «Show» (рис. 29).

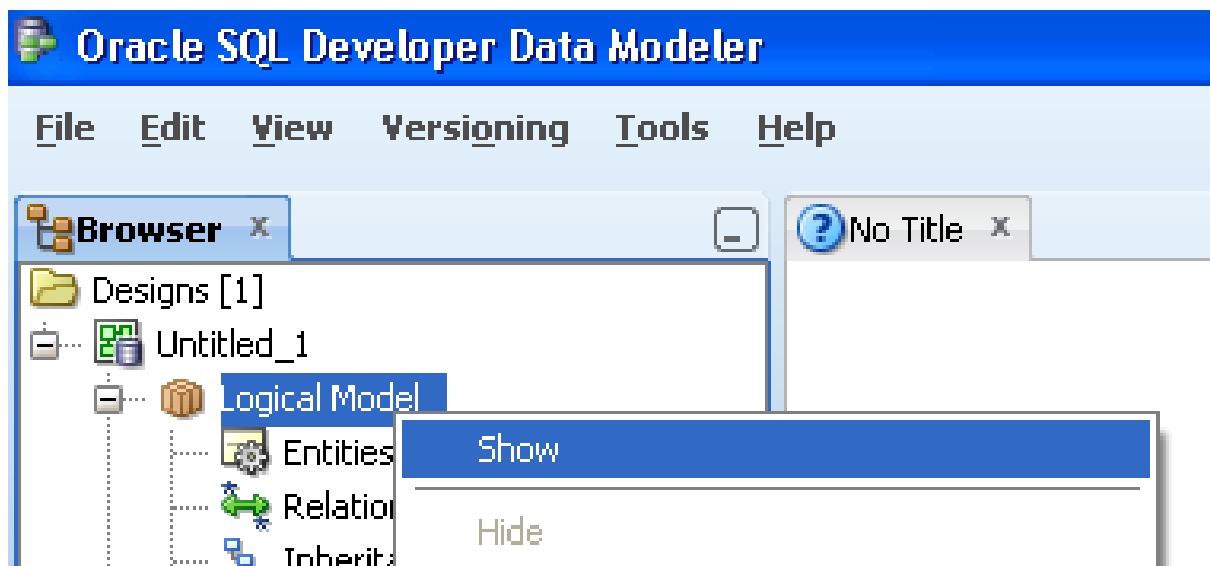


Рис. 29. Создание логической модели в Oracle SQL Developer Data Modeler

5. В правом рабочем окне появится вкладка «Logical (Untitled_1)», а сверху панель инструментов «Logical Models» (рис. 30).

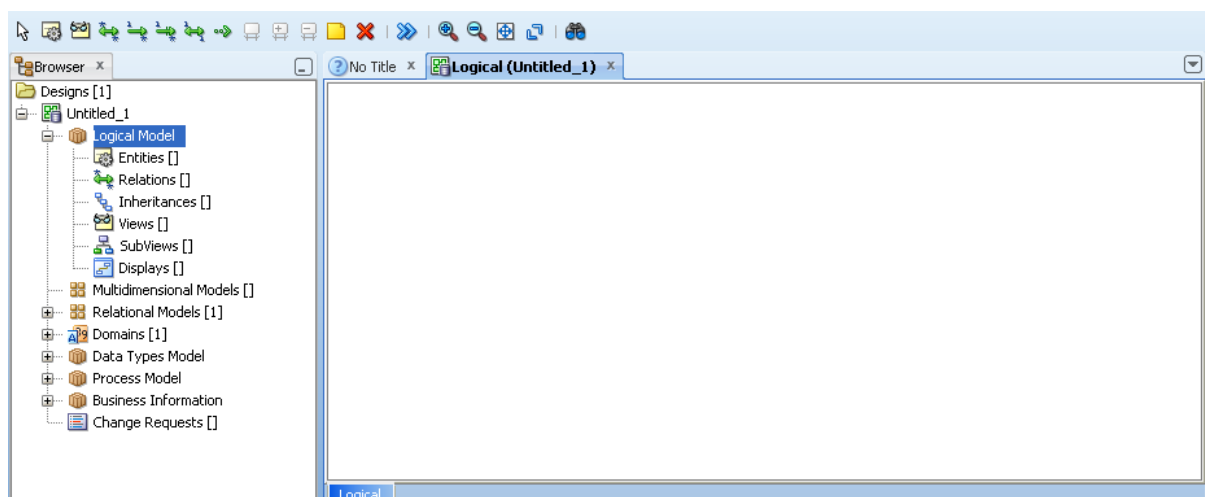


Рис. 30. Работа в новой вкладке «Logical (Untitled_1)»

6. Логическая модель оперирует понятиями «сущность», «атрибут» и «связь». Для создания сущностей и связей между ними в панели инструментов есть кнопки: «New Entity» (новая сущность) и ряд кнопок «New 1:1(1:N; M:N) Relation» (для создания связей разной кратности). Созданную здесь ER-диаграмму потом можно будет конвертировать в реляционную модель, щелкнув правой кнопкой мыши по пиктограмме «Logical Models» и выбрав «Engineer to Relational Model» (рис. 31).

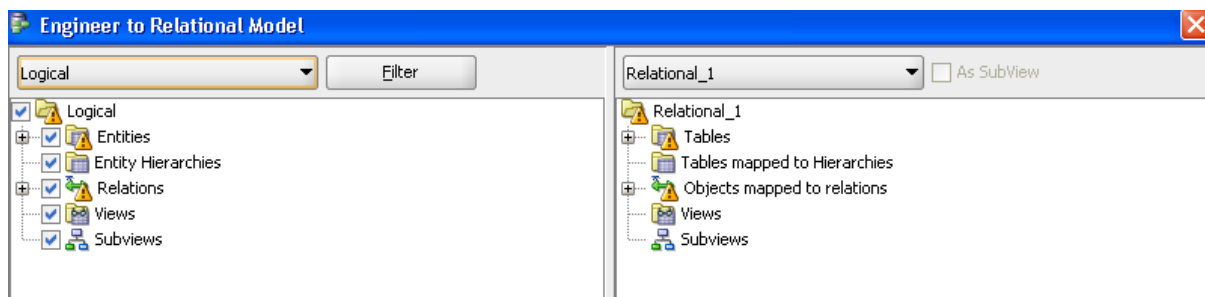


Рис. 31. Создание ER-диаграммы в Oracle SQL Developer Data Modeler

7. Для создания реляционной модели щелкните правой кнопкой мыши по пиктограмме «Relational Models» в левой панели «Browser» и нажмите «New Relation Model», если у вас нет ни одной реляционной модели, или «Show», если она есть (рис. 32).

8. В правом рабочем окне появится вкладка «Relational_1 (Untitled_1)», а сверху панель инструментов «Relational Models».

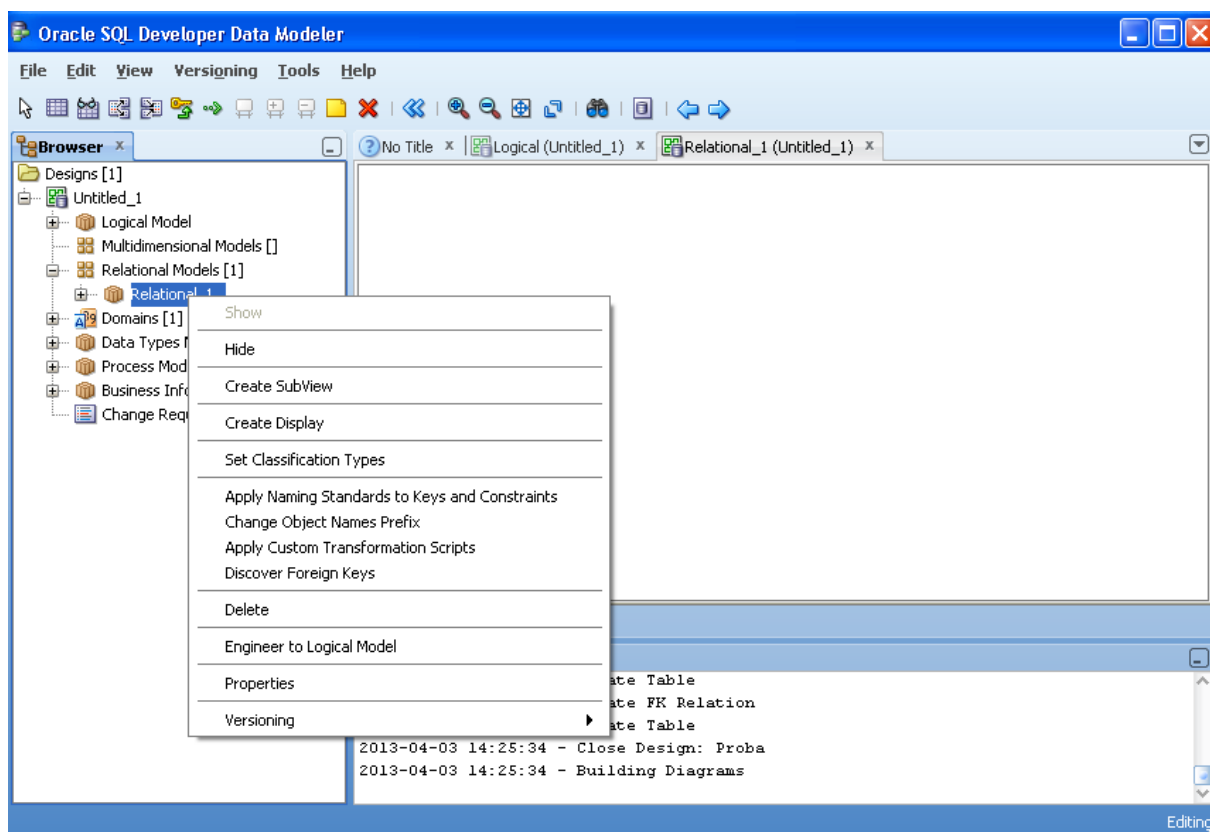


Рис. 32. Реляционная модель в Oracle SQL Developer Data Modeler

9. Реляционная модель оперирует понятиями «таблица», «представление», «столбец» и «ключ». Для создания таблиц и связей между ними в панели инструментов есть кнопки: «New Table» (новая таблица), «New View» (новое представление) и несколько кнопок для создания дополнительных параметров. Созданную здесь модель данных потом можно будет конвертировать в физическую модель для кон-

кретной СУБД, щелкнув правой кнопкой мыши по пиктограмме «Physical Models» в списке «Relational_1» и выбрав «New» (рис. 33).

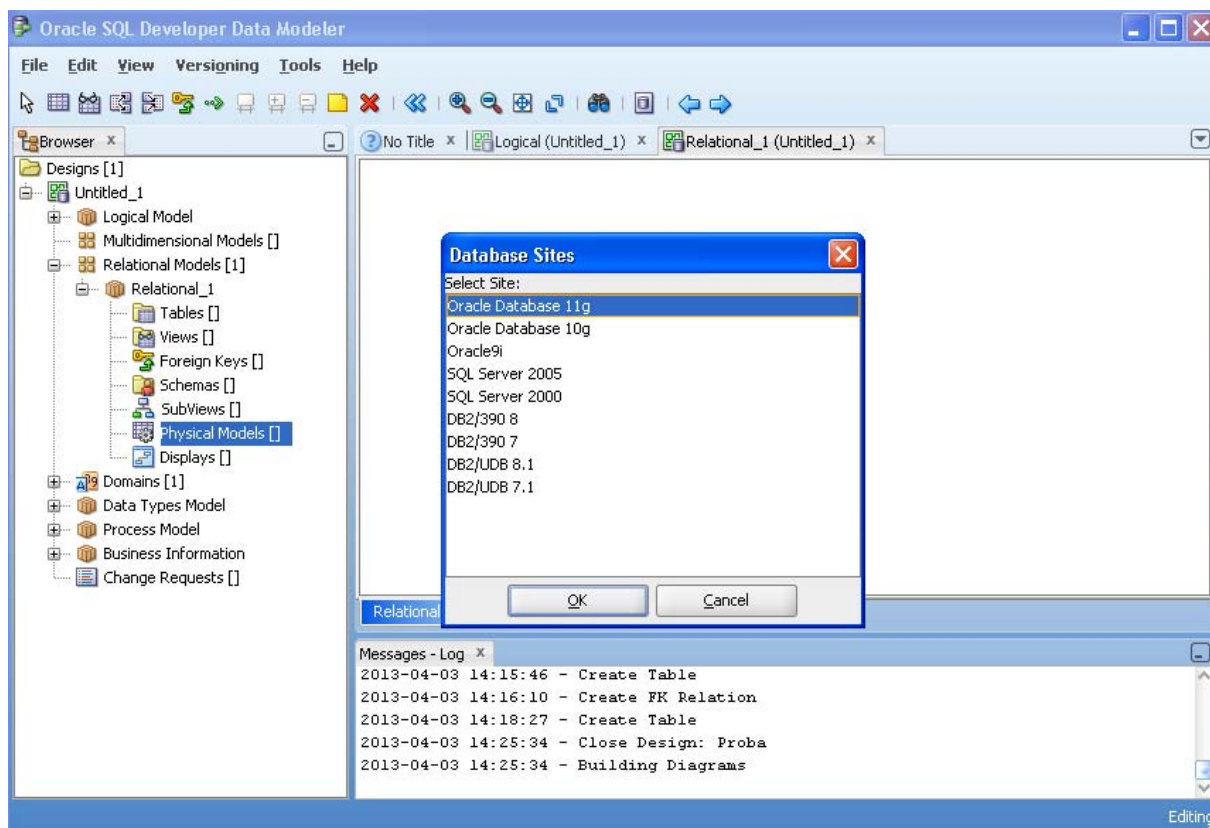


Рис. 33. Физическая модель в Oracle SQL Developer Data Modeler

Oracle SQL Developer Data Modeler предлагает еще множество возможностей, узнать о которых можно из документации. В следующей лабораторной работе мы рассмотрим практическое применение Data Modeler при разработке ER-диаграммы.

Практическая работа №5

Разработка ER-диаграмм

Построение ER-диаграмм обычно начинается с общего описания исследуемого объекта, его процедур и операций. Затем создается графическое представление базовой ER-модели. При исследовании ER-модели, скорее всего, появятся дополнительные сущности, атрибуты и связи. Поэтому базовая ER-модель будет изменяться до тех пор, пока пользователи или разработчики не посчитают, что разработанная ER-диаграмма правильно отражает деятельность исследуемого объекта. Oracle SQL Developer Data Modeler позволяет создавать наглядную ER-диаграмму, с последующим созданием на ее основе реляционной и физической моделей.

Пример создания ER-диаграммы

1. Создавать ER-диаграмму будем в среде Data Modeler. Запустить Data Modeler можно при помощи меню «Пуск ⇒ Oracle Database 11g Express Edition ⇒ Data Modeler». После запуска перед вами появится главное окно Oracle SQL Developer Data Modeler (ранее рис. 28).

2. Для создания логической модели щелкните правой кнопкой мыши по пиктограмме «Logical Models» в левой панели «Browser» и нажмите «Show» (ранее рис.29).

3. В правой части экрана появится рабочее окно, а сверху панель инструментов «Logical Models» (ранее рис. 30).

4. Логическая модель оперирует понятиями «сущность», «атрибут» и «связь». Создадим сущность «Student» с атрибутами «Stud_Id», «Stud_FIO», «Stud_Date_Birth», «Stud_Phone» и сущность «Groupe» с атрибутами «Group_Id», «Group_Name», «Group_Course», «Group_Faculty».

Для этого в панели инструментов нажмем «NewEntity» и щелкнем на рабочем окне «Logical (Untitled_1)». Появится окно создания новой сущности. В поле «Name» пишем имя сущности «Student» (рис. 34).

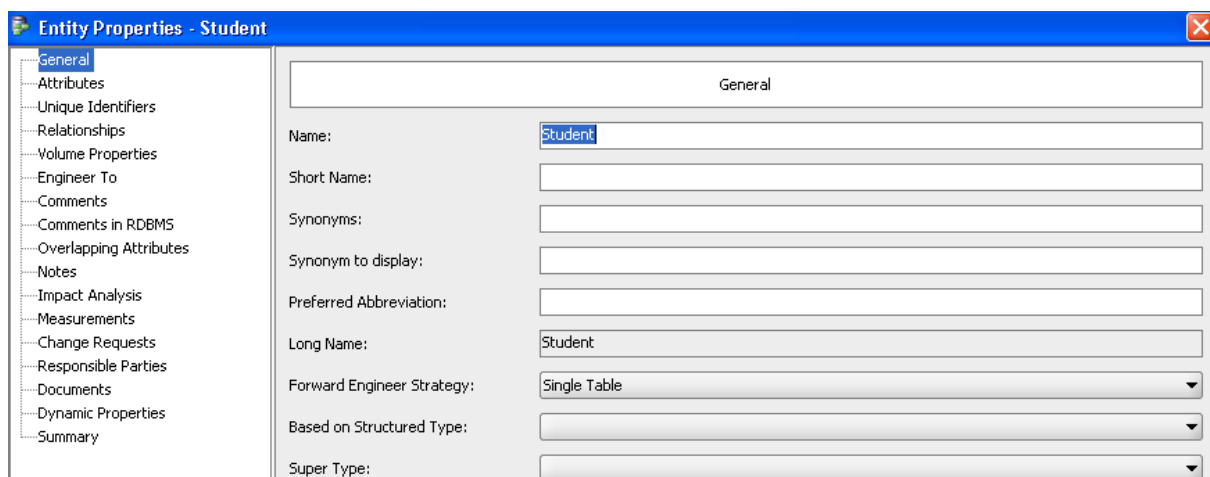


Рис. 34. Вкладка «Logical» в Oracle SQL Developer Data Modeler

5. В списке свойств выбираем «Attributes» и добавляем необходимые атрибуты (рис. 35).

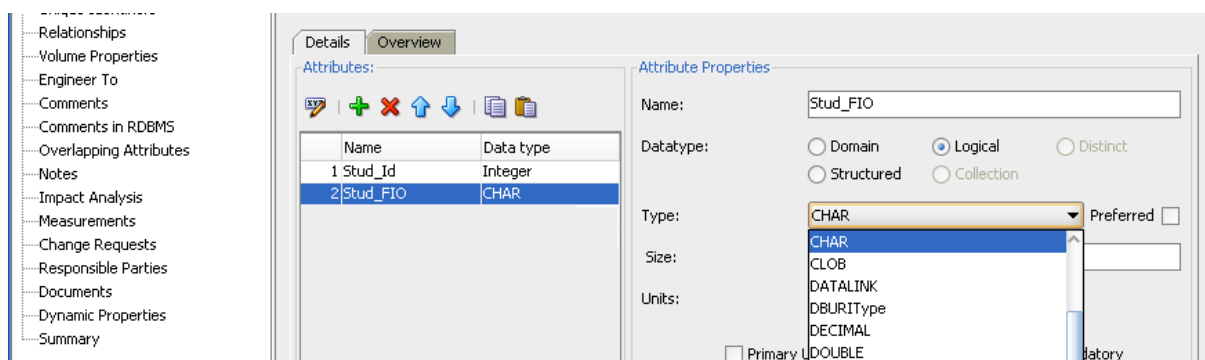


Рис. 35. Список свойств «Attributes» в Oracle SQL Developer Data Modeler

Для первого атрибута в поле «Name» указываем имя атрибута «Stud_Id», «Data type» - «Logical», в списке «Type» выбираем «Integer», ставим галочку «PrimaryUID» для указания уникального атрибута; для второго атрибута в поле «Name» указываем имя атрибута «Stud_FIO» и так далее.

6. Нажимаем «ОК» и получаем сущность. Аналогично добавляем вторую сущность (рис. 36).

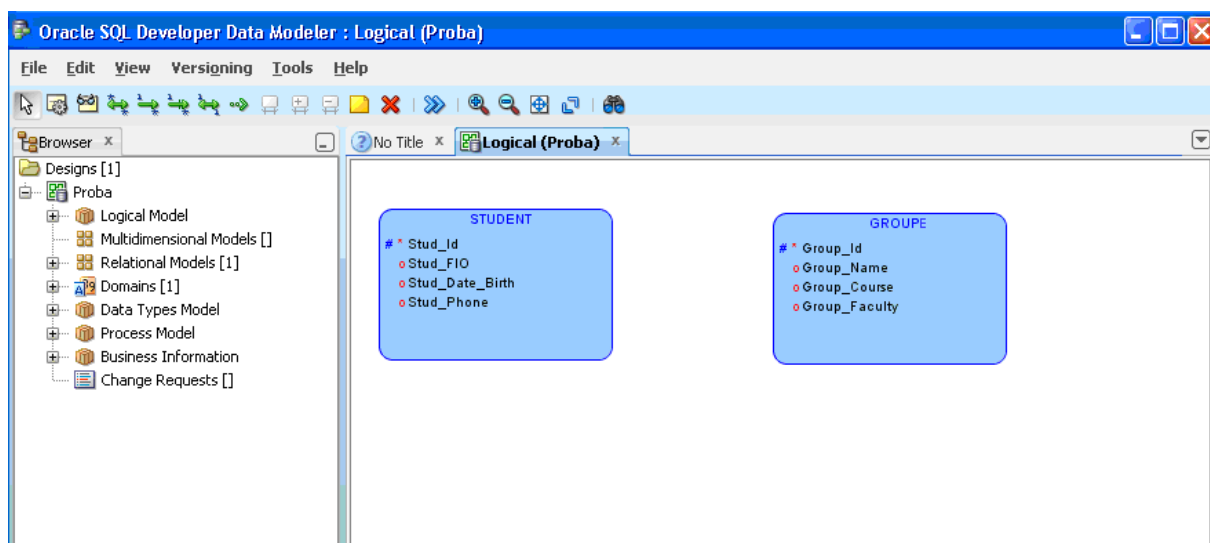


Рис. 36. Отображение сущностей в Oracle SQL Developer Data Modeler

7. Теперь нам необходимо связать эти сущности. Для этого в панели инструментов выбираем «New 1:N Relation» и выбираем сущности, которые необходимо связать. Появится окно «Relation Properties» (рис.37).

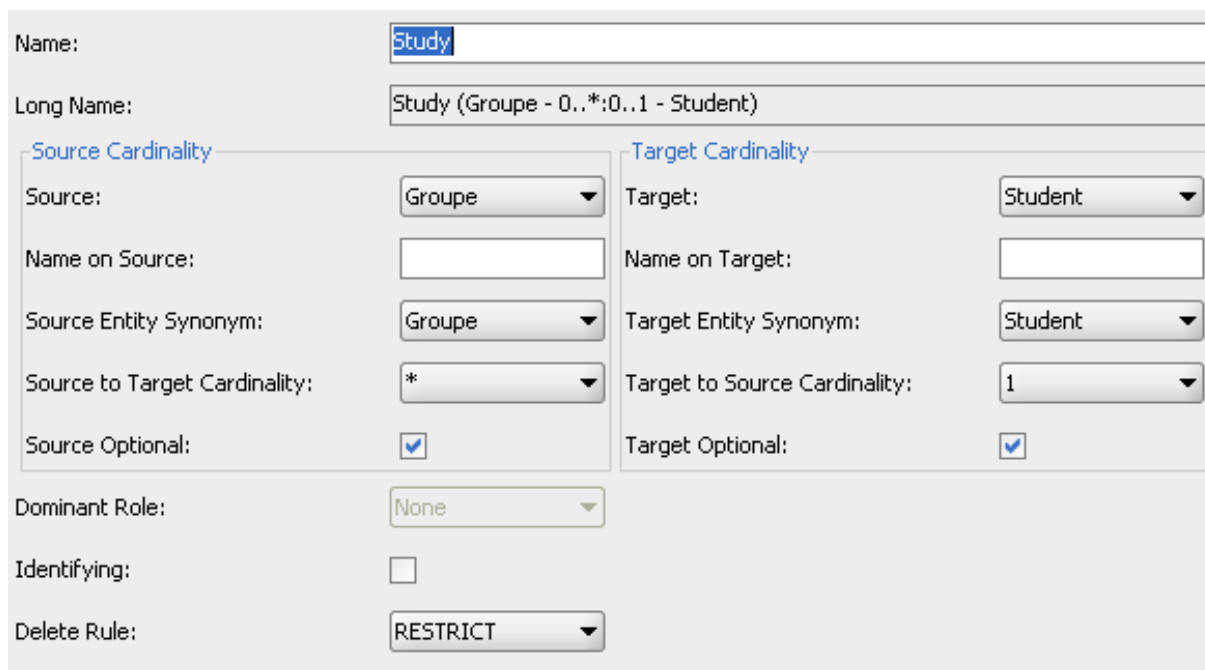


Рис. 37. Свойства связей в Oracle SQL Developer Data Modeler

8. Здесь указываем имя отношения («Name»), и кратность у сущности «Groupe» – «*», у сущности «Student» – «1» (Для Data Modeler кратность отношения ставятся для противоположной сущности, то есть «Source to Target Cardinality» кратность ставится для «Student»). Жмем «OK».

9. Мы получили ER-диаграмму. Теперь сохраним ее «File ⇒ Save As» (рис. 38).

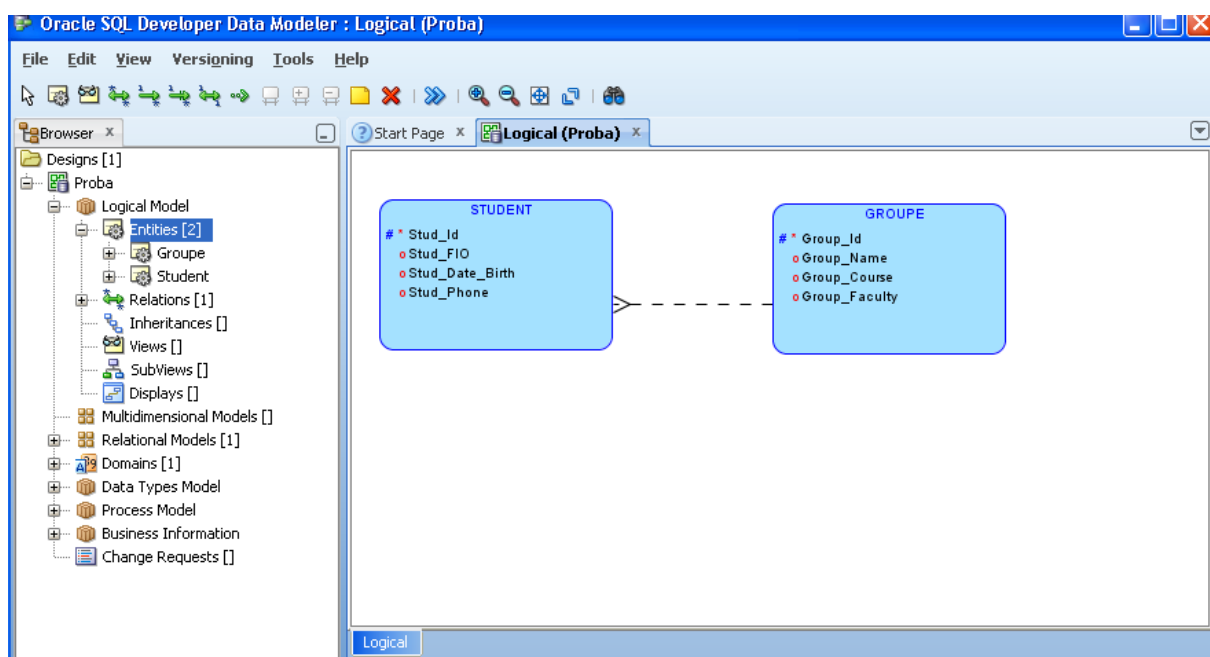


Рис. 38. ER-диаграмма в Oracle SQL Developer Data Modeler

Разработка ER-диаграммы по индивидуальному заданию

1. Прочитать описание предметной области по своему варианту.
2. Выявить сущности и их атрибуты, необходимые для построения диаграммы.
3. Построить необходимое количество вариантов ER-диаграммы, последовательно сохраняя их в личной папке.
4. Подготовить отчет по проделанной работе.

Практическая работа №6

Формирование таблиц БД в среде Data modeler

Oracle SQL Developer Data Modeler – мощное средство для создания реляционных моделей. Data Modeler позволяет создавать наглядную реляционную модель как с «нуля», так и как продолжение созданной ранее концептуальной модели. Data Modeler позволяет трансформировать все правила и решения, сделанные на концептуальном уровне, в реляционную модель, в которой детали уточняются и обновляются.

Формирование таблиц БД

1. Запустим Data Modeler при помощи меню «Пуск->Oracle Database 11g Express Edition-> Data Modeler». После запуска перед вами появится главное окно Oracle SQL Developer Data Modeler (ранее рис. 28).

2. Откроем сохраненную на предыдущем занятии (практическая работа №5) логическую модель «File->Open» (ранее рис. 38).

3. Реляционная модель оперирует понятиями «таблица», «столбец» и «ключ». Создать реляционную модель можно несколькими способами. Для начала создадим нашу модель вручную. Создадим таблицы «Student» и «Groupe» со столбцами, которые мы определили в прошлой работе. При создании логической модели автоматически создается и реляционная модель. Щелкните правой кнопкой мыши по пиктограмме «Relational_1» в левой панели «Browser» и нажмите «Show». Если ее нет, то щелкните правой кнопкой мыши по пиктограмме «Relational Models» в левой панели «Browser» и нажмите «New Relational Model», а затем «Show». В правом рабочем окне появится вкладка «Relational_1», а сверху панель инструментов «Relational Models» (рис. 39).

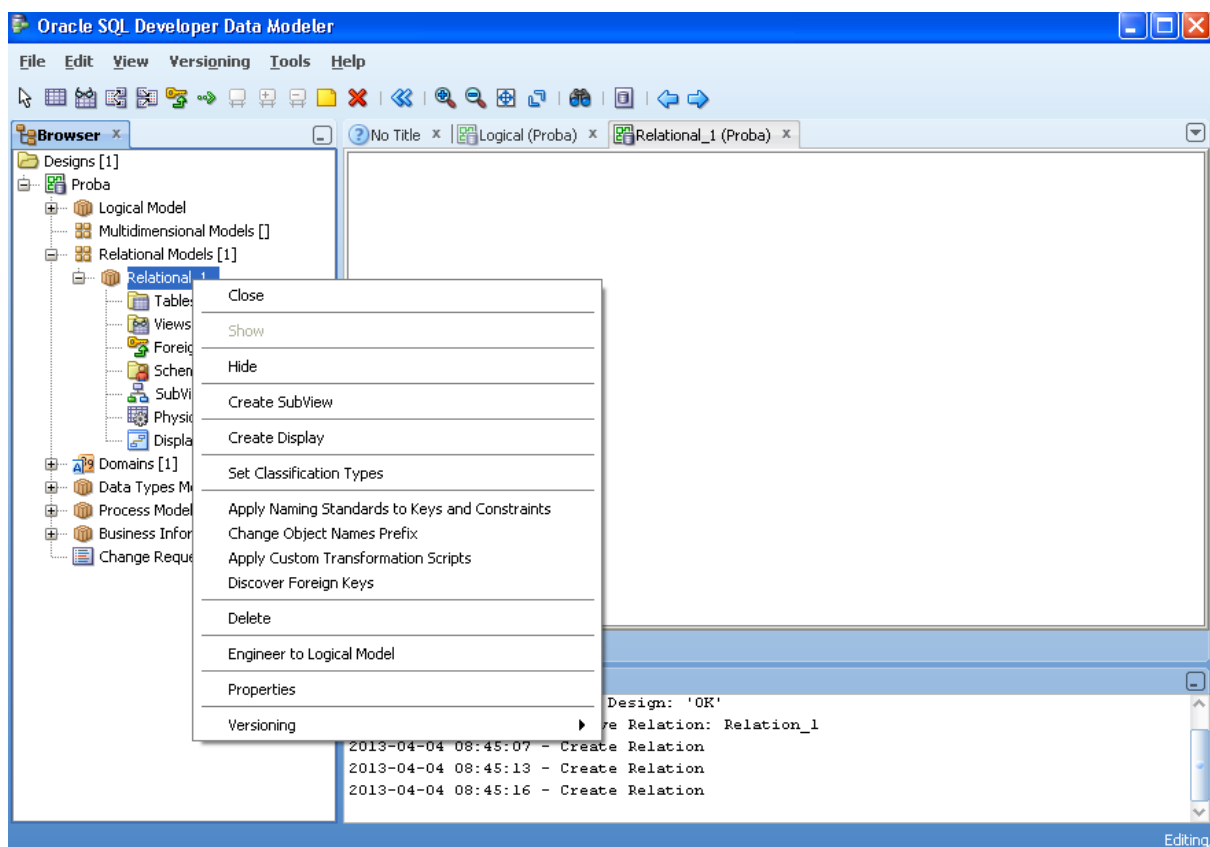


Рис. 39. Создание реляционной модели вручную

4. Для создания таблицы в панели инструментов нажмем кнопку: «New Table» и щелкнем на рабочем окне «Relational_1». Появится окно создания новой таблицы. В поле «Name» пишем имя таблицы «Student» (рис. 40).

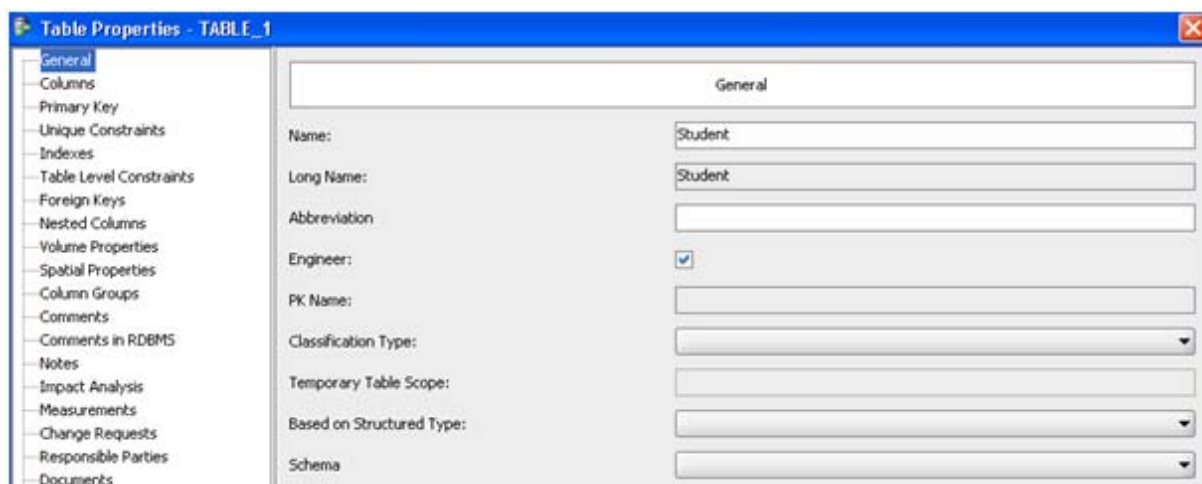


Рис. 40. Создание таблицы в Oracle SQL Developer Data Modeler

5. В списке свойств выбираем «Columns» и добавляем необходимые столбцы. Для первого столбца в поле «Name» указываем имя «Stud_Id», «Datatype» – «Logical», в списке «Type» выбираем «Integer», ставим галочку «PK» для указания первичного ключа; для второго столбца в поле «Name» указываем имя атрибута «Stud_FIO» и так далее (рис. 41).

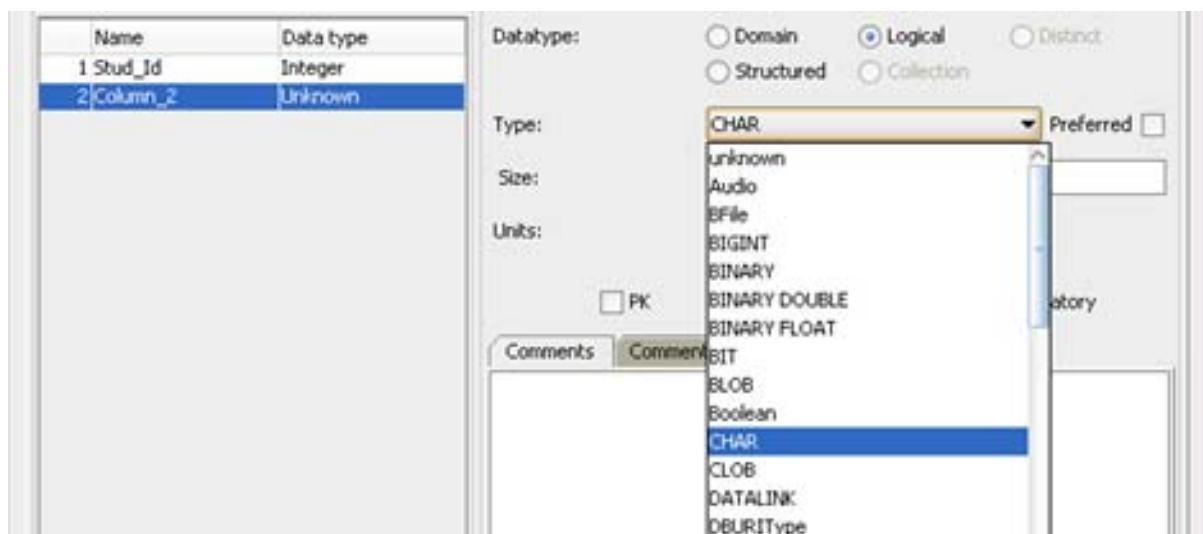


Рис. 41. Добавление колонок в Oracle SQL Developer Data Modeler

6. Нажимаем «ОК» и получаем таблицу. Аналогично добавляем вторую таблицу(рис. 42).

Student		
P *	Stud_Id	INTEGER
	Stud_FIO	CHAR
	Stud_Date_Birth	DATE
	Stud_Phone	UNKNOWN
Student_PK (Stud_Id)		

Group		
P *	Group_Id	INTEGER
	Group_Name	CHAR
	Group_Course	INTEGER
	Group_Faculty	CHAR
Group_PK (Group_Id)		

Рис. 42. Вторая таблица в Oracle SQL Developer Data Modeler

7. Теперь нам необходимо связать эти таблицы. Для этого можно воспользоваться окном создания/редактирования таблиц и свойст-

вом «Foreign Keys». Мы же будем использовать панель инструментов и кнопку «New FK Relation». Нажимаем кнопку и выделяем наши таблицы. Появится окно создания нового внешнего ключа (рис. 43).

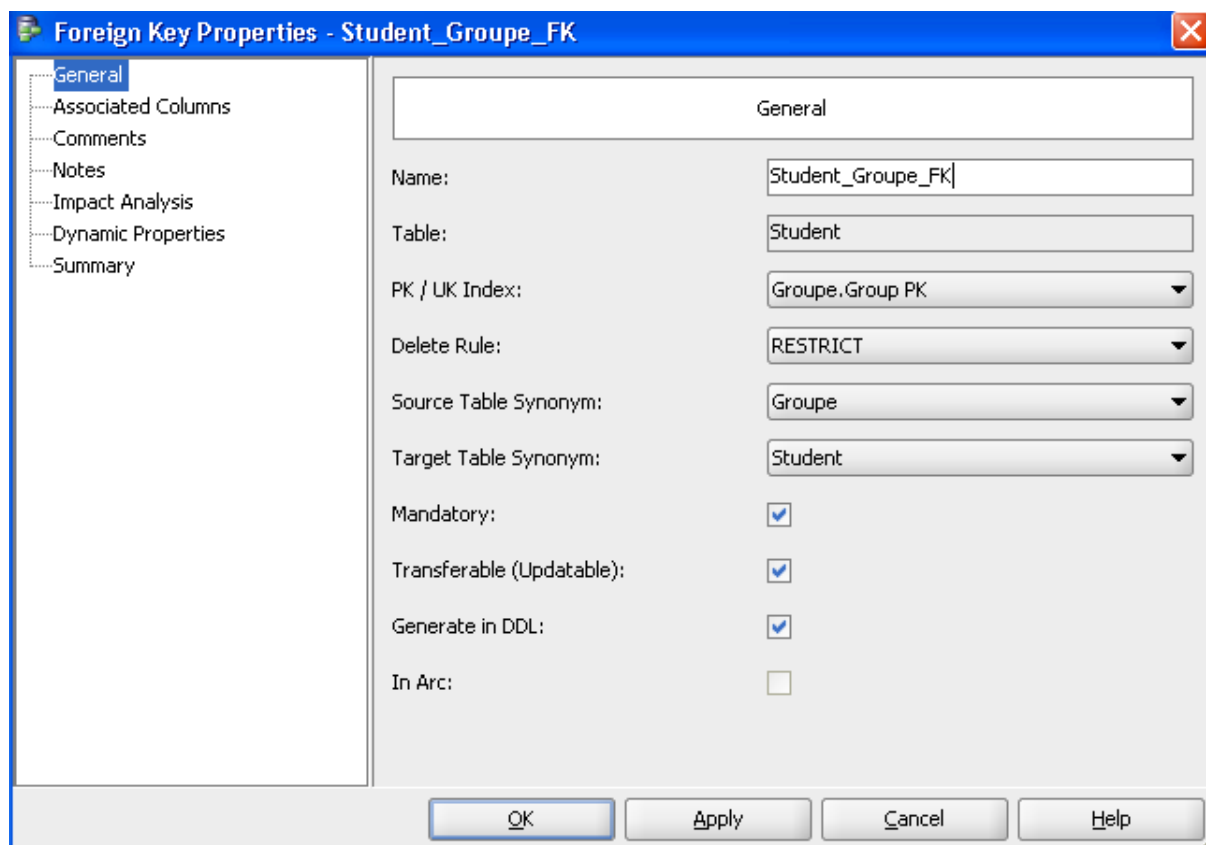


Рис. 43. Создание внешнего ключа в Oracle SQL Developer Data Modeler

8. Нажимаем «ОК» и получаем связанные таблицы (рис. 44).

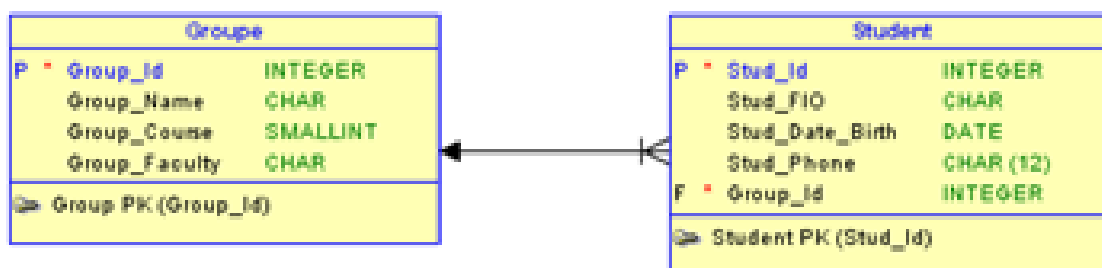


Рис. 44. Результат связывания таблиц в Oracle SQL Developer Data Modeler

9. Теперь давайте попробуем получить наши таблицы из готовой логической модели. Удалим созданные таблицы и щелкнем правой кнопкой мыши по пиктограмме «Logical Models» в левой панели «Browser». Выберем пункт «Engineer to Relational Model». Появится окно конвертирования логической модели в реляционную (рис. 45).

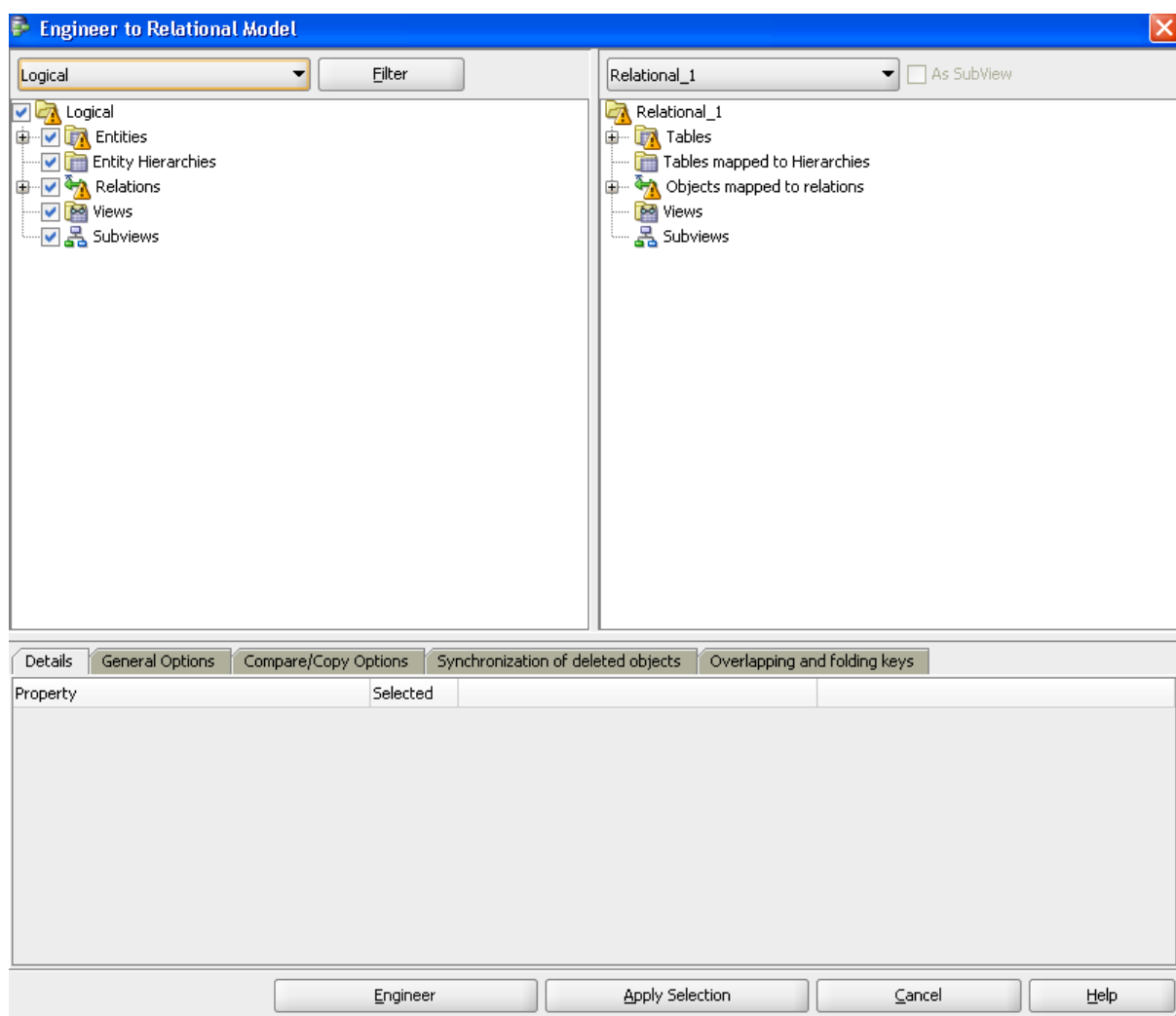


Рис. 45. Конвертирование логической модели в реляционную

10. Сверху в списках указываются объекты для конвертирования, в нижних вкладках – параметры. Нажимаем кнопку «Engineer» и получаем готовые таблицы (рис. 46).

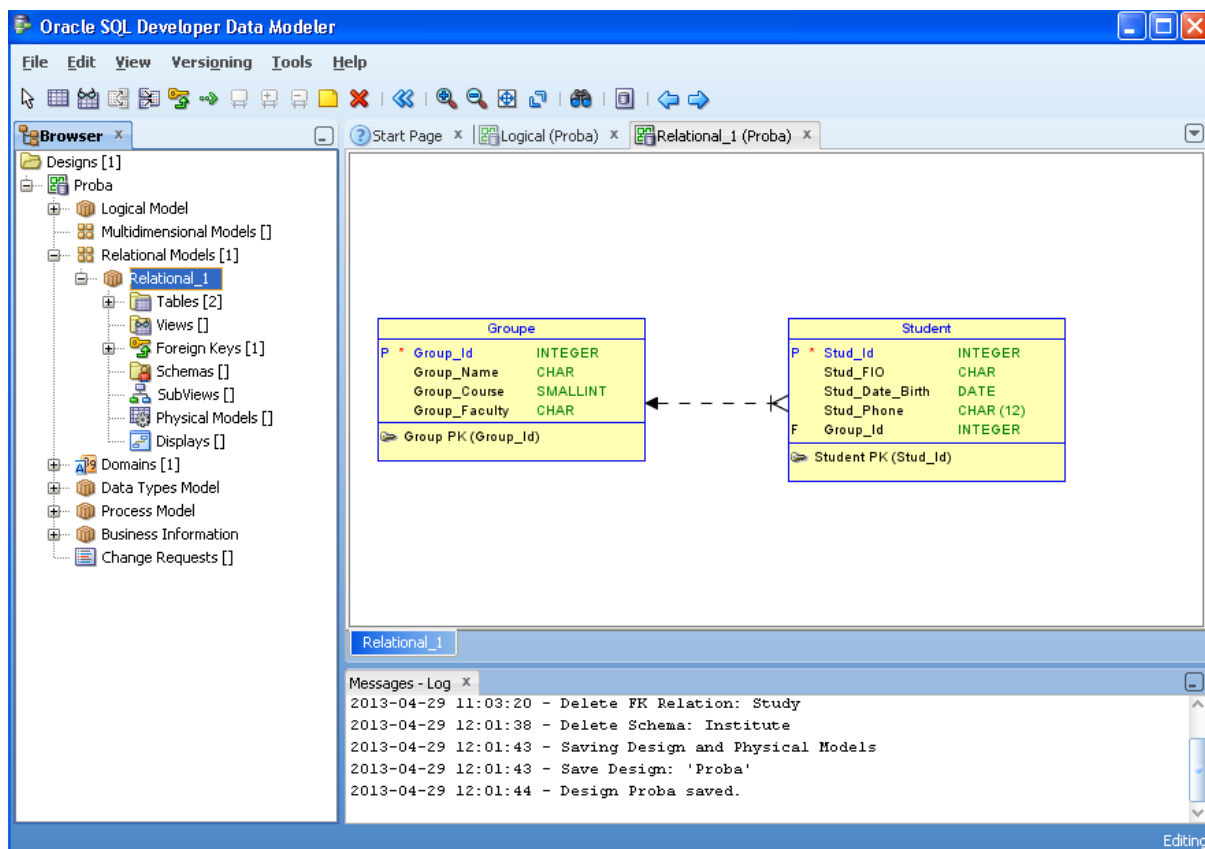


Рис. 46. Таблицы, полученные при конвертировании модели

11. Мы изучили различные способы создания реляционной модели. Теперь сохраним ее «File ⇒ Save As».

Формирование таблиц по индивидуальному заданию

1. Откройте последнюю сохраненную на прошлом занятии индивидуальную логическую модель «File->Open».
2. Трансформируйте ее в реляционную модель, как показано в примере.
3. Сохраните ее «File->Save As».
4. Подготовьте отчет по проделанной работе.

Практическая работа №7

Загрузка сформированных таблиц в БД Oracle

Oracle SQL Developer Data Modeler дает возможность загружать сформированные в реляционной модели таблицы в Oracle наиболее удобными способами. На сегодняшнем занятии мы рассмотрим некоторые из них.

Загрузка таблиц в БД

1. Запустим Data Modeler при помощи меню «Пуск->Oracle Database 11g Express Edition->Data Modeler». После запуска перед вами появится главное окно Oracle SQL Developer Data Modeler (ранее рис. 28).

2. Откроем сохраненную на прошлом занятии реляционную модель «File->Open» (ранее рис. 48).

3. Сформируем DDL файл, который потом можно будет выполнить в SQL *Plus. Для этого в панели инструментов нажмем кнопку «Generate DDL». Появится окно создания DDL (рис. 47).

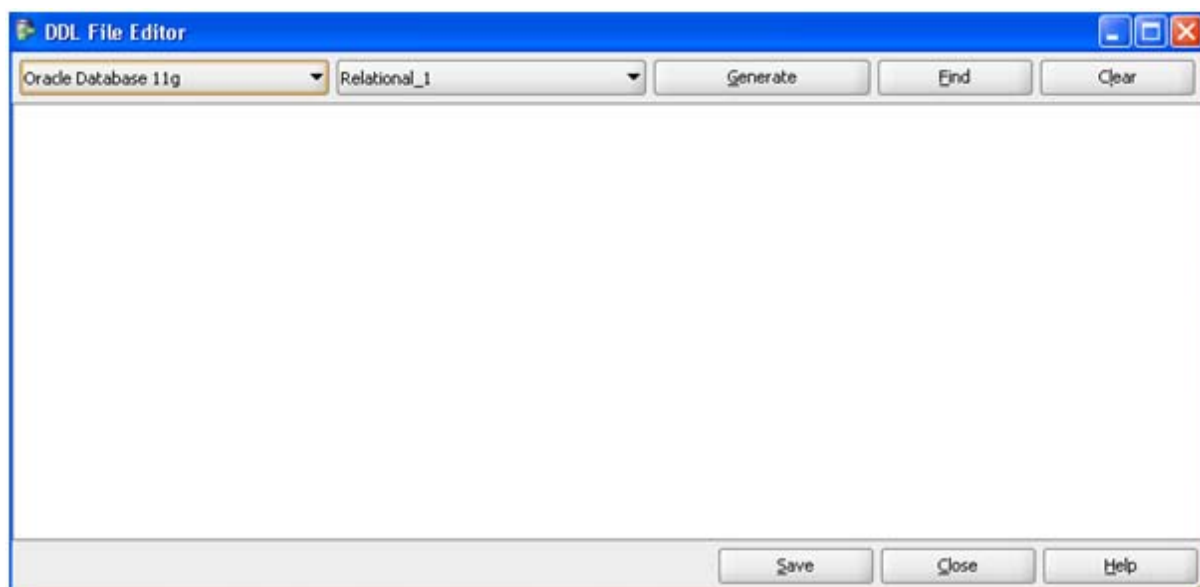


Рис. 47. Окно создания DDL файла в Oracle SQL Developer Data Modeler

4. В выпадающем списке укажите: для какой СУБД генерируется DDL и имя реляционной модели. Нажимаем «Generate». Появится окно, в котором можно указать объекты для создания и некоторые дополнительные опции (рис. 48).



Рис. 48. Выбор опций при создании DDL файла

5. Нажимаем «OK» и получаем созданный DDL-скрипт. Сохраняем его «Save» под именем «CREATETABLES.DDL» (рис. 51).

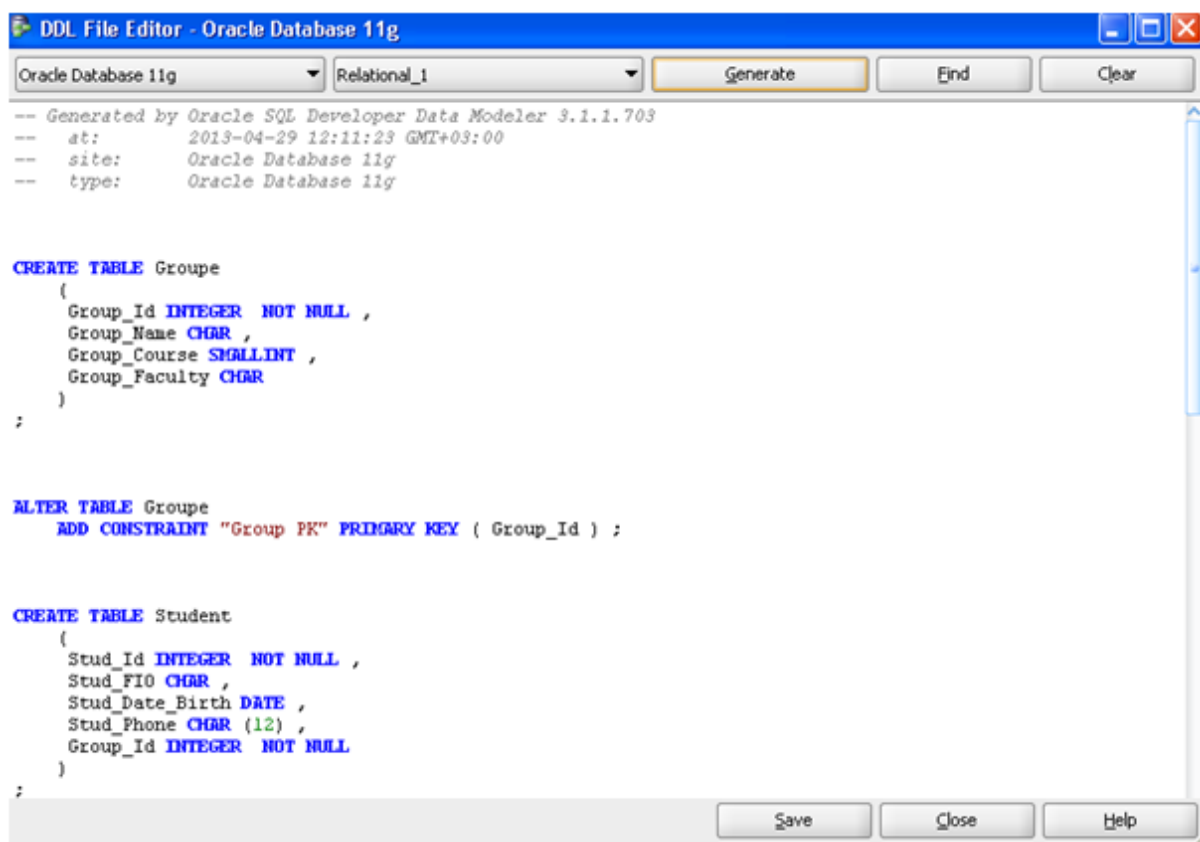


Рис. 49. Текст DDL-скрипта в Oracle SQL Developer Data Modeler

6. Далее запускаем утилиту, соединяемся с БД под пользователем «INSTITUTE» и запускаем DDL на выполнение (рис. 50).



Рис. 50. Выполнения DDL скрипта в среде SQL *Plus

7. Таблицы, созданные при помощи описанного выше DDL, также можно сгенерировать и в программе SQL Developer. Для создания таблиц необходимо создать соединение со схемой данных и выполнить сформированный ранее DDL скрипт. Для связи с БД нам необходимо создать новое соединение «New Connection...» в правом окне «Connections». Откроется окно «New / Select Database Connection». В этом окне указываем настройки соединения (рис. 51).

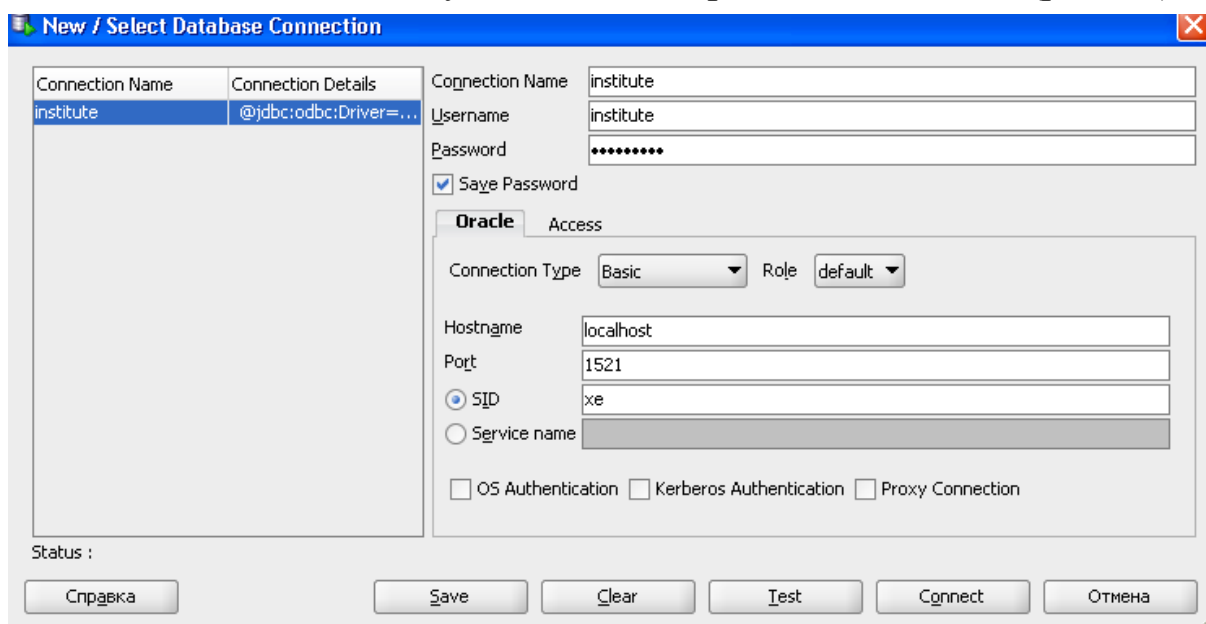


Рис. 51. Создание соединения со схемой данных из SQL Developer

8. Нажимаем «Test» для проверки соединения и, если ошибок нет, «Connect» (рис. 52).

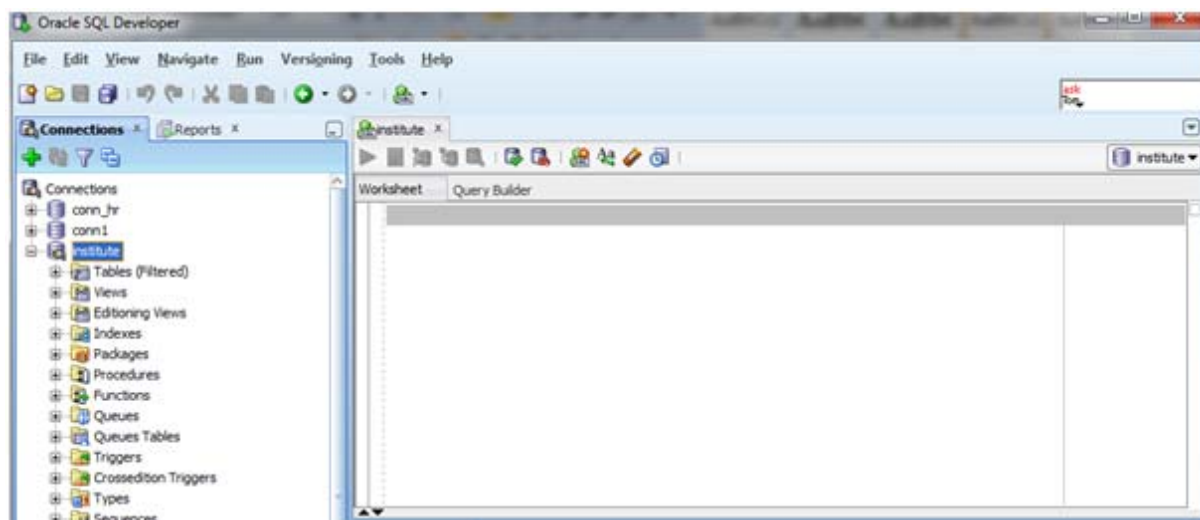


Рис. 52. Проверка соединения со схемой данных из SQL Developer

9. Открывается соединение, в панели «Connections» которого отображаются все объекты нашей схемы. Здесь можно создавать таблицы, представления, типы, вносить данные, исправлять структуру БД и многое другое. Мы же будем формировать созданные в Data Modeler таблицы. Для этого скопируем наш DDL код в окно «Worksheet», которое открылось справа автоматически при создании подключения (рис.53).

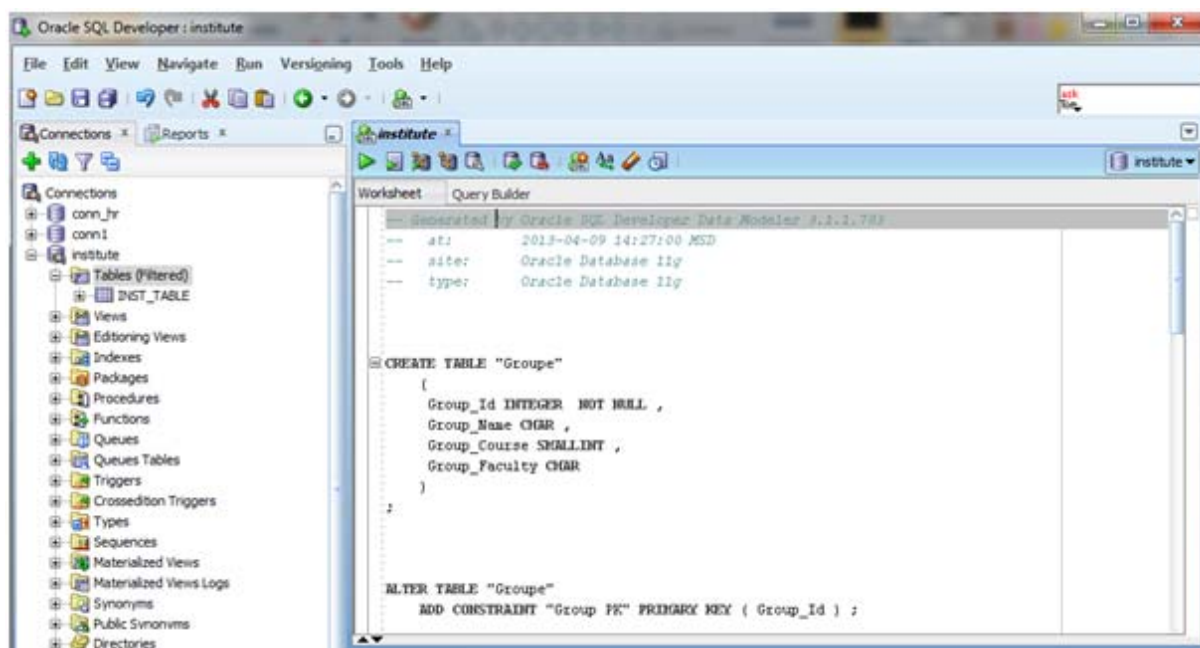


Рис. 53. Добавления кода для создания таблиц в SQL Developer

10. Для проверки нажимаем «Run Statement» в панели инструментов над окном «Worksheet» для выполнения «Run Script» и для подтверждения изменений «Commit» (рис. 54).

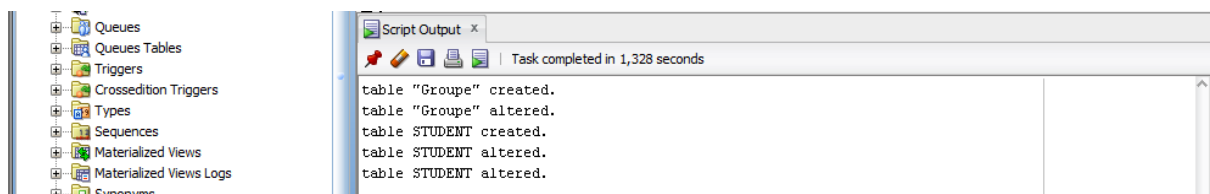


Рис. 54. Успешное создание таблиц в SQL Developer

11. Открыв список «Tables (Filtered)», мы можем убедиться в создании наших таблиц (рис.55).

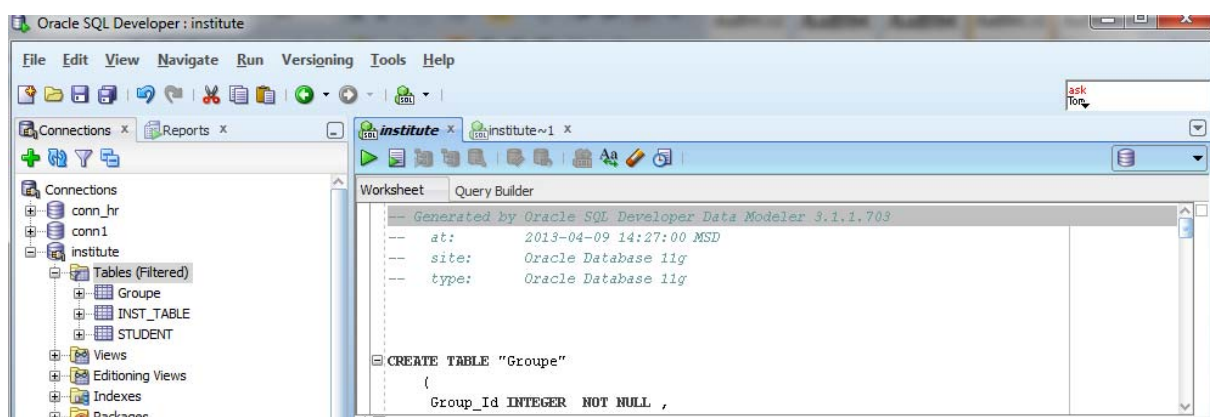


Рис. 55. Демонстрация созданных таблиц в SQL Developer

12. Мы изучили различные способы загрузки сформированных таблиц в БД Oracle.

Загрузка таблиц индивидуального задания в БД Oracle

1. Загрузите сохраненные таблицы вашего индивидуального задания в БД Oracle любым удобным вам способом.
2. Сохраните состояние базы данных.
3. Подготовьте отчет по проделанной работе.

ЛАБОРАТОРНЫЕ РАБОТЫ

Лабораторная работа №1

Создание таблиц средствами SQL*Plus. Оператор Create Table

В данной лабораторной работе мы изучим создание таблиц в Oracle посредством языка SQL. Для создания, изменения и удаления таблиц в SQL используются операторы «**CREATE TABLE**», «**ALTER TABLE**» и «**DROP TABLE**».

Синтаксис оператора «**CREATE TABLE**»:

		SQL
1	CREATE TABLE имя_таблицы имя_столбца тип_данных [NOT NULL] [DEFAULT значение_по_умолчанию] [PRIMARY KEY]	

Синтаксис оператора «**ALTER TABLE**»:

		SQL
1	ALTER TABLE имя_таблицы « ОПЕРАТОР » значение	

«**ОПЕРАТОР**» может принимать различные значения, например, **ADD COLUMN** – Добавление нового столбца;

DROP PRIMARY KEY – Удаление первичного ключа таблицы и так далее.

Синтаксис оператора «**DROP TABLE**»:

		SQL
1	DROP TABLE имя_таблицы	

Создание таблиц

1. Соединимся с Oracle под своим именем. Для этого вводим команду

		SQL
1	CONNECT СВОЕ_ИМЯ	

Далее необходимо ввести пароль, который вы назначили при создании себя как пользователя базы данных (ранее рис. 22).

2. Далее создадим две учебные таблицы:

«Student» со столбцами

«Stud_Id»,
«Stud_FIO»,
«Stud_Date_Birth»,
«Stud_Phone»

и «Groupe» со столбцами

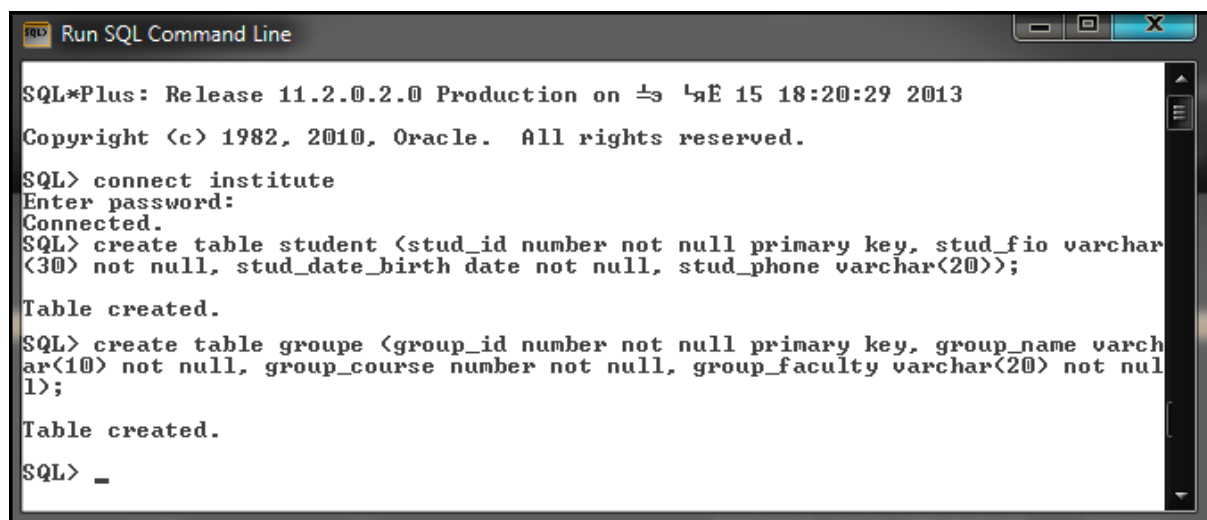
«Group_Id»,
«Group_Name»,
«Group_Course»,
«Group_Faculty».

Для этого напомним SQL команду:

		SQL
1	CREATE TABLE	STUDENT (STUD_ID NUMBER NOT NULL PRIMARY KEY, STUD_FIO VARCHAR(30) NOT NULL, STUD_DATE_BIRTH DATE NOT NULL, STUD_PHONE VARCHAR(20))

где **NOT NULL** указывает, что данное значение не должно быть пустым, а **PRIMARY KEY** указывает первичный ключ.

Аналогично создаем вторую таблицу (рис. 56).



```
Run SQL Command Line
SQL*Plus: Release 11.2.0.2.0 Production on 15 18:20:29 2013
Copyright (c) 1982, 2010, Oracle. All rights reserved.
SQL> connect institute
Enter password:
Connected.
SQL> create table student (stud_id number not null primary key, stud_fio varchar
(30) not null, stud_date_birth date not null, stud_phone varchar(20));
Table created.
SQL> create table groupe (group_id number not null primary key, group_name varchar
(10) not null, group_course number not null, group_faculty varchar(20) not nul
l);
Table created.
SQL> _
```

Рис. 56. Создание учебных таблиц

3. Далее нам необходимо связать таблицы. Для этого в таблицу «Student» добавим столбец «Group_Id» и сделаем его внешним ключом. Напишем SQL команды (рис. 57):

		SQL
1	ALTER TABLE STUDENT ADD (GROUP_ID NUMBER NOT NULL); ALTER TABLE STUDENT ADD CONSTRAINT STUD_FK1 FOREIGN KEY (GROUP_ID) REFERENCES GROUPE (GROUP_ID)	

```

SQL> alter table student add <group_id number not null>;
Table altered.
SQL> alter table student add constraint stud_fk1 foreign key <group_id> references groupe<group_id>;
Table altered.
SQL>

```

Рис. 57. Добавление внешнего ключа в таблицу «GROUPE»

4. Для удаления таблиц используется команда

		SQL
1	DROP TABLE STUDENT	
2	DROP TABLE GROUPE	

!	<i>Однако созданные в данной работе таблицы удалять не следует – они потребуются в следующих работах.</i>
---	---

Лабораторная работа №2

Ввод данных в таблицы средствами SQL*Plus.

Оператор Insert

В данной лабораторной работе мы изучим ввод данных в Oracle посредством языка SQL. Для внесения данных в SQL используют оператор «**INSERT INTO**», он имеет несколько вариантов.

Синтаксис оператора «**INSERT INTO**»:

SQL	
1	INSERT INTO имя_таблицы VALUES (значение1, ...).

Значения, заносимые после оператора «**VALUES**», должны соответствовать порядку перечисления спецификаций столбцов таблицы в операторе «**CREATE TABLE**».

SQL	
1	INSERT INTO имя_таблицы (имя_столбца, ...) VALUES (значение, ...).

Здесь списки имен столбцов и значений ячеек добавляемой строки должны быть согласованы, хотя нет никаких требований к их порядку.

SQL	
1	INSERT INTO имя_таблицы [(имя_столбца, ...)] SELECT

Такой оператор дает возможность добавить в таблицу сразу несколько новых строк, полученных в результате запроса к базе данных, реализуемого оператором «**SELECT**».

Ввод данных в таблицы

1. Соединимся с Oracle под своим именем.

SQL	
1	CONNECT СВОЕ_ИМЯ

Далее необходимо ввести пароль, который вы назначили при создании себя как пользователя базы данных (ранее рис. 22).

2. Для изучения работы оператора мы воспользуемся таблицами, созданными на прошлом занятии. Вставим несколько записей в таблицу «**GROUPE**» (рис. 58). Напишем

SQL	
1	INSERT INTO GROUPE VALUES (1, 'AISTBV-11', 1, 'SAMOLETOSTROENIE')

Самостоятельно добавьте еще несколько записей, в том числе и свою группу.

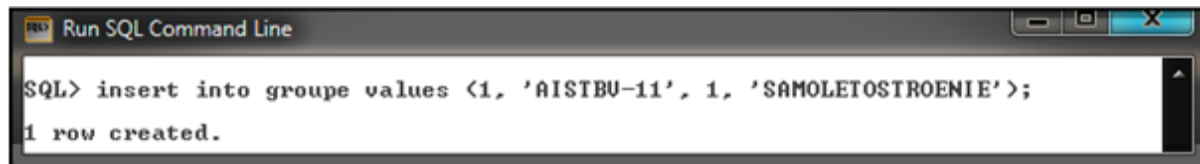


Рис. 58. Добавление записи в таблицу «GROUPE»

3. Теперь заполним таблицу «STUDENT». Напишем

SQL	
1	INSERT INTO STUDENT VALUES (1, 'Semenov Igor Olegovich', '23.05.1994', '26-33-75', 1)

Последнее значение должно соответствовать номеру необходимой группы в таблице «GROUPE» (рис. 59).

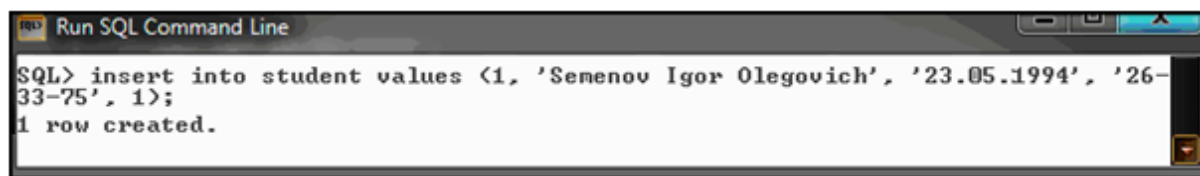


Рис. 59. Добавление записи в таблицу «GROUPE»

4. Теперь используем другой синтаксис, перечисляя столбцы в произвольном порядке (рис. 60):

SQL	
1	INSERT INTO STUDENT (STUD_ID, GROUP_ID, STUD_FIO, STUD_PHONE, STUD_DATE_BIRTH) VALUES (2, 3, 'Petrov Sergey Viktorovich', '36-48-24', '14.07.1993');

Самостоятельно добавьте еще несколько записей для каждой группы, в том числе и студентов своей группы.

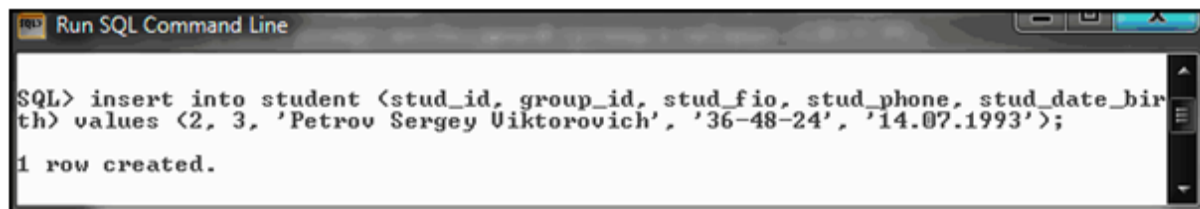


Рис. 60. Добавление записей в таблицу «STUDENT»

5. Для облегчения работы можно использовать возможности SQL plus.

Лабораторная работа №3

Виртуальное объединение таблиц баз данных (создание представлений)

В данной лабораторной работе мы изучим создание представлений Oracle посредством языка SQL. Представление (VIEW) – объект данных, который не содержит никаких данных его владельца. Это – тип таблицы, чье содержание выбирается из других таблиц с помощью выполнения запроса. Результат запроса представляется в виде обычной таблицы. Представление, которое предоставляет доступ к данным одной или нескольких таблиц, является основным схемным объектом в базе данных Oracle. Представления обычно используют для создания дополнительного уровня защиты табличных данных, а также для упрощения разработки приложений. Представления могут быть как изменяемыми, так и неизменяемыми. Представления создаются оператором «**CREATE VIEW**».

Синтаксис оператора «**CREATE VIEW**»:

		SQL
1	CREATE VIEW имя_представления AS SELECT [ALL DISTINCT] в_выражение, ... FROM имя_таблицы [синоним_таблицы], ... [WHERE сложное_условие],	

где «в_выражение» – количество и название столбцов;

«синоним_таблицы» – необязательный синоним имени таблицы.

Создание представлений

1. Соединимся с Oracle под пользователем «INSTITUTE». Для этого вводим команду (ранее рис. 22):

		SQL
1	CONNECT INSTITUTE	

Далее необходимо ввести пароль «INSTITUTE».

2. Создадим представление, которое выводит все записи из таблицы «GROUPE» (рис. 61):

		SQL
1	CREATE VIEW GROUP_ALL AS SELECT * FROM GROUPE	

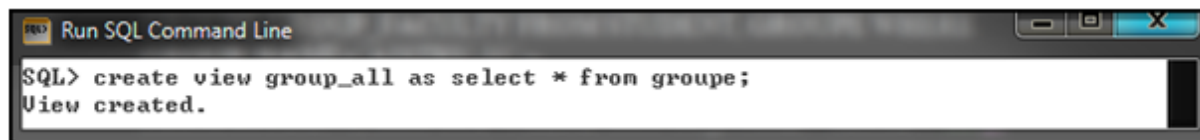


Рис. 61. Создание представления «GROUP_ALL»

3. Представление может быть создано как на основании одной таблицы, так и на основе нескольких. Создадим такое представление (рис. 62).

Наберем:

		SQL
1	CREATE VIEW STUDENT_GROUP AS SELECT STUDENT.STUD_FIO, STUDENT.STUD_DATE_BIRTH, STUDENT.PHONE, GROUPE.GROUP_NAME, GROUPE.GROUP_COURSE, GROUPE.GROUP_FACULTY FROM STUDENT, GROUPE WHERE STUDENT.GROUP_ID=GROUPE.GROUP_ID AND GROUP_NAME = 'AISTBV-31'	

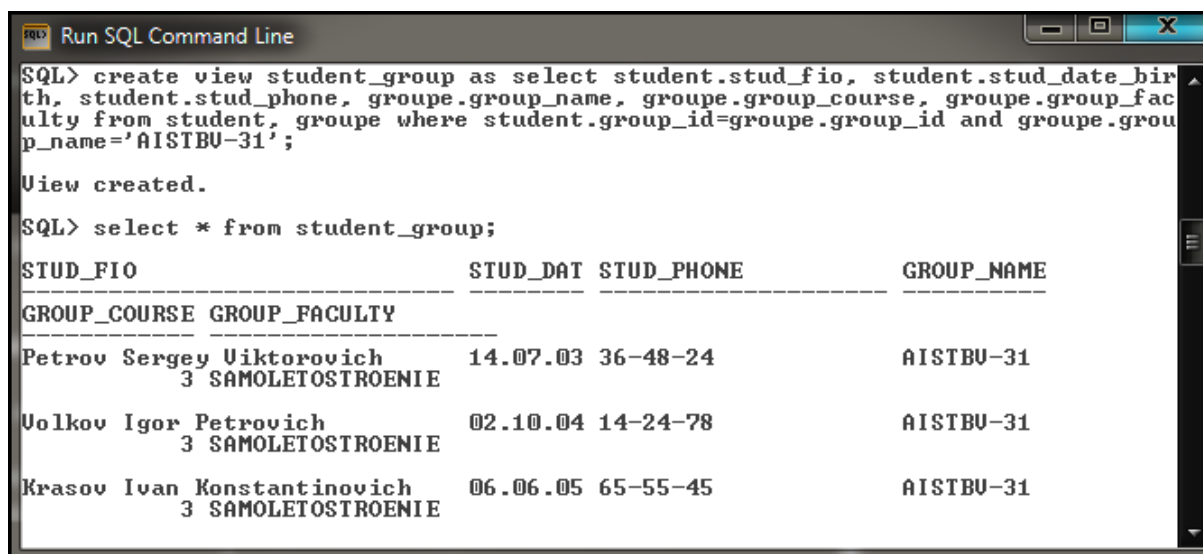


Рис. 62. Обновление структуры представления «STUDENT_GROUP»

4. Если явно не указывать, что создаваемое представление является представлением только для чтения (WITH READ ONLY), то Oracle создаст представление, которые можно использовать для вставки, обновления или удаления строк базовой таблицы. Представление позволяет вставлять и обновлять только такие строки, которые это представление может, в свою очередь, выбрать (рис. 63). Изменим представление «STUDENT_GROUP»:

		SQL
1	UPDATE STUDENT_GROUP SET STUD_PHONE='55-55-55'	
	WHERE STUD_FIO='Petrov Sergey Viktorovich'	

```

SQL> update student_group set stud_phone='55-55-55' where stud_fio='Petrov Sergey Viktorovich';
1 row updated.
SQL> select * from student_group;
STUD_FIO          STUD_DAT  STUD_PHONE          GROUP_NAME
-----
GROUP_COURSE  GROUP_FACULTY
-----
Petrov Sergey Viktorovich  14.07.03  55-55-55          AISTBU-31
      3 SAMOLETOSTROENIE
Volkov Igor Petrovich      02.10.04  14-24-78          AISTBU-31
      3 SAMOLETOSTROENIE
Krasov Ivan Konstantinovich 06.06.05  65-55-45          AISTBU-31
      3 SAMOLETOSTROENIE
SQL> _
  
```

Рис. 63. Изменение данных в представлении «STUDENT_GROUP»

5. Для замены представления можно переопределить представление оператором CREATE VIEW с параметром OR REPLACE. Параметр OR REPLACE заменяет текущее определение представления (рис. 64):

		SQL
1	CREATE OR REPLACE VIEW STUDENT_GROUP AS SELECT	
	STUDENT.STUD_FIO, STUDENT.STUD_DATE_BIRTH, STUDENT.PHONE, GROUPE.GROUP_NAME,	
	GROUPE.GROUP_COURSE, GROUPE.GROUP_FACULTY FROM	
	STUDENT, GROUPE WHERE STUDENT.GROUP_ID=	
	GROUPE.GROUP_ID AND GROUP_NAME='AISTBV-11'	

```

SQL> create or replace view student_group as select student.stud_fio, student.stud_date_birth, student.stud_phone, groupe.group_name, groupe.group_course, groupe.group_faculty from student, groupe where student.group_id=groupe.group_id and groupe.group_name='AISTBU-11';

View created.

SQL> select * from student_group;

```

STUD_FIO	STUD_DAT	STUD_PHONE	GROUP_NAME
Semenov Igor Olegovich 1 SAMOLETOSTROENIE	23.05.04	26-33-75	AISTBU-11
Ivanov Petr Semenovich 1 SAMOLETOSTROENIE	05.02.03	21-45-45	AISTBU-11
Novikov Ivan Ivanovich 1 SAMOLETOSTROENIE	12.01.03	14-98-25	AISTBU-11

```

SQL>

```

Рис. 64. Замена представления «STUDENT_GROUP»

6. Представление удаляется использованием оператора «DROP VIEW» (рис. 65):

		SQL
1	DROP VIEW GROUP_ALL	

```

e.group_faculty from student, groupe where student.group_id=groupe.group_id and groupe.group_name='AISTBU-11';

View created.

SQL> select * from student_group;

```

STUD_FIO	STUD_DAT	STUD_PHONE	GROUP_NAME
Semenov Igor Olegovich 1 SAMOLETOSTROENIE	23.05.04	26-33-75	AISTBU-11
Ivanov Petr Semenovich 1 SAMOLETOSTROENIE	05.02.03	21-45-45	AISTBU-11
Novikov Ivan Ivanovich 1 SAMOLETOSTROENIE	12.01.03	14-98-25	AISTBU-11

```

SQL> drop view group_all;

View dropped.

SQL>

```

Рис. 65. Удаление представления «STUDENT_GROUP»

Лабораторная работа № 4

Построение запросов в SQL*Plus. Оператор Select

В данной лабораторной работе мы изучим получение данных из Oracle посредством языка SQL. Для выборки данных в SQL используются оператор «SELECT», он имеет сложный и многовариантный синтаксис. Оператор «SELECT» является фактически самым важным для пользователя и самым сложным оператором SQL. Результатом выполнения оператора «SELECT» всегда является таблица. По результатам действий оператор «SELECT» похож на операторы реляционной алгебры. Любой оператор реляционной алгебры может быть выражен подходящим образом сформулированным оператором «SELECT». Сложность оператора «SELECT» определяется тем, что он содержит в себе все возможности реляционной алгебры, а также дополнительные возможности, которых в реляционной алгебре нет.

В общем случае синтаксис оператора «SELECT» выглядит так:

SQL	
1	SELECT [ALL DISTINCT] <i>в_выражение</i> , ... FROM <i>имя_таблицы</i> [<i>синоним_таблицы</i>], ... [WHERE <i>сложное_условие</i>] [GROUP BY <i>полное_имя_столбца номер_столбца</i> , ...] [ORDER BY <i>полное_имя_столбца номер_столбца</i> [ASC DESC], ...] [HAVING <i>сложное_условие</i>];

где «*в_выражение*» – количество и название столбцов;

«*синоним_таблицы*» – необязательный синоним имени таблицы;

«*номер_столбца*» – номер столбца результирующей таблицы.

Построение запросов

1. Соединимся с Oracle под пользователем «INSTITUTE». Для этого вводим команду «**CONNECT** INSTITUTE». Далее необходимо ввести пароль «INSTITUTE» (ранее рис. 22).

2. Выберем из таблицы «GROUPE» все записи (рис. 66):

		SQL
1	SELECT * FROM GROUPE	

Символ «*» имеет значение «все столбцы таблиц из списка **FROM**».

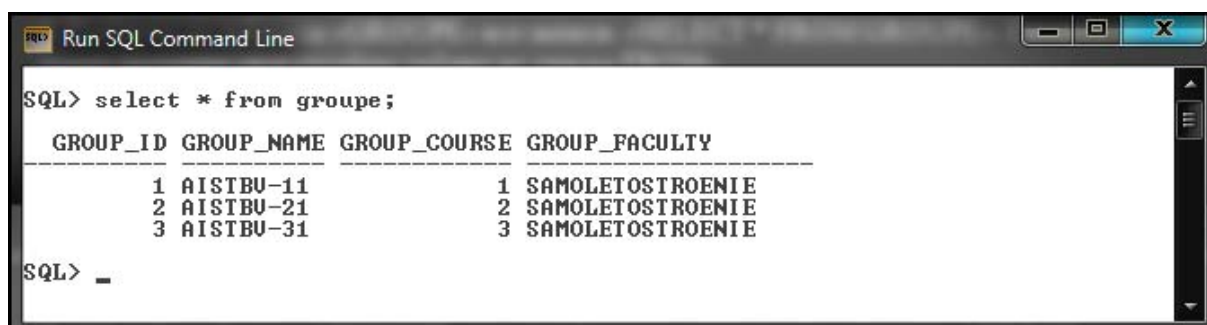


Рис. 66. Запрос данных из таблицы «GROUPE» с помощью «**SELECT**»

3. Простым (и также часто используемым) случаем в выражении является полное имя столбца одной из таблиц списка **FROM**. Выберем названия групп и курс из таблицы «GROUPE» (рис. 67):

		SQL
1	SELECT GROUP_NAME, GROUP_COURSE FROM GROUPE	

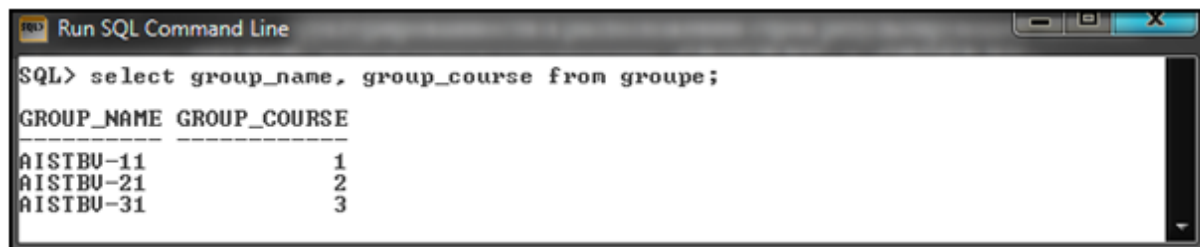


Рис. 67. Выбор названия групп и курс из таблицы «GROUPE»

4. В качестве критерия выбора информации из таблиц списка «**FROM**» оператора «**SELECT**» выступает сложное условие, записываемое после ключевого слова «**WHERE**». Выберем из таблицы «STUDENT» всех, чье имя «Ivan» (рис. 68):

		SQL
1	SELECT STUD_FIO, STUD_DATE_BIRTH, STUD_PHONE FROM STUDENT WHERE STUD_FIO LIKE '%IVAN%';	

Оператор **LIKE** выбирает все строки, которые соответствуют имени «IVAN», а метасимвол «%» указывает, что до и после условия могут стоять любые символы.

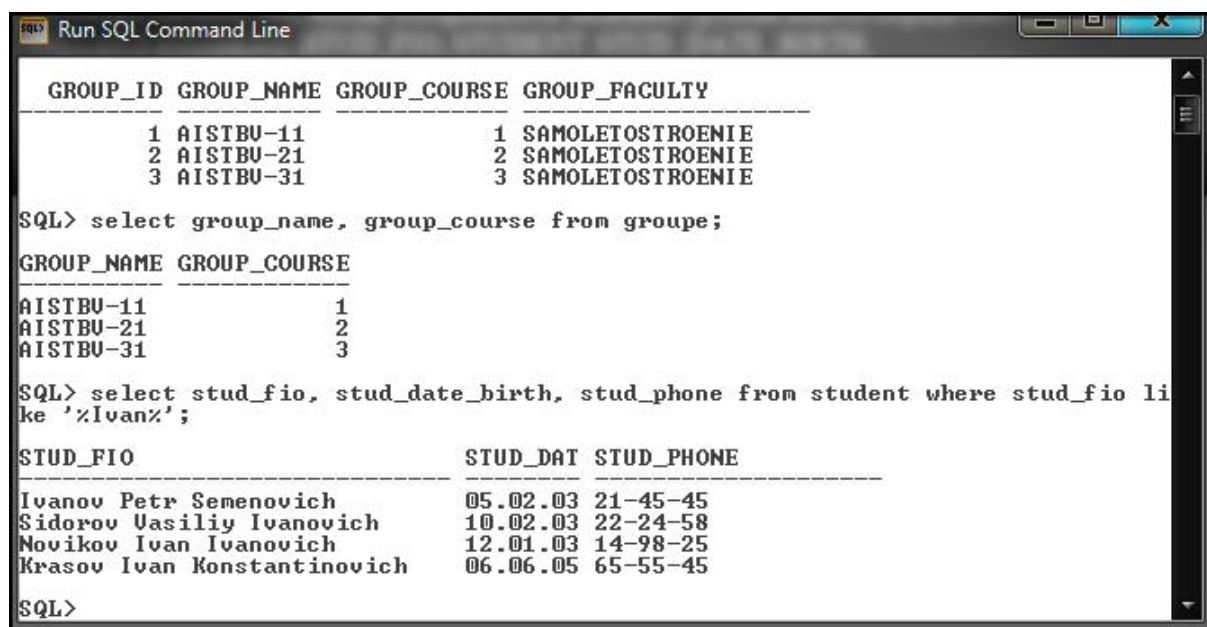


Рис. 68. Выбор по неполному условию из таблицы «STUDENT»

5. Для обеспечения структурированности в расположении строк результирующей таблицы в операторе «**SELECT**» используются конструкции «**GROUP BY**» и «**ORDER BY**». Выберем из таблицы «STUDENT» все записи и упорядочим их по группе (рис. 69):

		SQL
1	SELECT STUD_FIO, STUD_DATE_BIRTH, STUD_PHONE FROM STUDENT ORDER BY STUD_FIO;	

```

SQL> select stud_fio, stud_date_birth, stud_phone from student order by stud_fio
;

```

STUD_FIO	STUD_DAT	STUD_PHONE
Ivanov Petr Semenovich	05.02.03	21-45-45
Krasov Ivan Konstantinovich	06.06.05	65-55-45
Kravcov Mihail Sergeevich	13.03.04	45-58-41
Novikov Ivan Ivanovich	12.01.03	14-98-25
Petrov Sergey Viktorovich	14.07.03	55-55-55
Semenov Igor Olegovich	23.05.04	26-33-75
Sidorov Vasiliy Ivanovich	10.02.03	22-24-58
Volkov Igor Petrovich	02.10.04	14-24-78

```

8 rows selected.
SQL>

```

Рис. 69. Упорядоченная выборка из таблицы «STUDENT»

6. Теперь сделаем так, чтобы отображалось название группы вместо порядкового номера (рис. 70):

```

1 SELECT STUDENT.STUD_FIO, STUDENT.STUD_PHONE,
  GROUPE.GROUP_NAME, GROUPE.GROUP_COURSE
FROM STUDENT, GROUPE
WHERE STUDENT.GROUP_ID=GROUPE.GROUP_ID
ORDER BY GROUPE.GROUP_NAME;

```

```

8 rows selected.
SQL> select student.stud_fio, student.stud_phone, groupe.group_name, groupe.grou
p_course from student, groupe where student.group_id=groupe.group_id order by gr
oupe.group_name;

```

STUD_FIO	STUD_PHONE	GROUP_NAME	GROUP_COURSE
Semenov Igor Olegovich	26-33-75	AISTBU-11	1
Novikov Ivan Ivanovich	14-98-25	AISTBU-11	1
Ivanov Petr Semenovich	21-45-45	AISTBU-11	1
Sidorov Vasiliy Ivanovich	22-24-58	AISTBU-21	2
Kravcov Mihail Sergeevich	45-58-41	AISTBU-21	2
Petrov Sergey Viktorovich	55-55-55	AISTBU-31	3
Volkov Igor Petrovich	14-24-78	AISTBU-31	3
Krasov Ivan Konstantinovich	65-55-45	AISTBU-31	3

```

8 rows selected.
SQL>

```

Рис. 70. Перекрестный запрос к таблицам «STUDENT» и «GROUPE»

Лабораторная работа №5

Дополнительные команды управления данными

Заканчиваем работать с утилитой SQL*Plus. В данной лабораторной работе мы рассмотрим дополнительные команды манипуляции данными посредством языка SQL. Для изменения структур данных в SQL используются оператор «ALTER», который имеет несколько вариантов в зависимости от необходимых изменений. Оператор «UPDATE» изменяет данные в таблицах, оператор «DELETE» удаляет данные из таблиц.

1. Синтаксис оператора «ALTER»:

SQL	
1	ALTER TABLE имя_таблицы ОПЕРАТОР значение

«ОПЕРАТОР» может принимать различные значения, например, **ADD COLUMN** – Добавление нового столбца.
DROP PRIMARY KEY – Удаление первичного ключа таблицы.

2. Синтаксис оператора «UPDATE»:

SQL	
1	UPDATE имя_таблицы SET имя_столбца=выражение, ... [WHERE сложное_условие]

Если конструкция **WHERE** в операторе отсутствует, то обновляются все строки таблицы.

3. Синтаксис оператора «DELETE»:

SQL	
1	DELETE FROM имя_таблицы [WHERE сложное_условие]

Если конструкция **WHERE** в операторе отсутствует, то удаляются все строки таблицы.

Ввод данных в таблицы

1. Соединимся с Oracle под пользователем «INSTITUTE». Для этого вводим команду «**CONNECT** INSTITUTE». Далее необходимо ввести пароль «INSTITUTE» (ранее рис. 22).

2. Добавим в таблицу «GROUPE» столбец (рис.71):

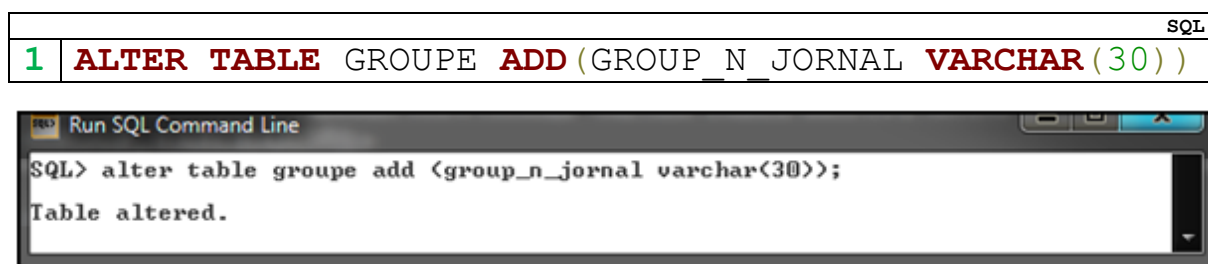


Рис. 71. Редактирование структуры таблицы «GROUPE»

3. Теперь изменим тип данных этого столбца (рис.72):

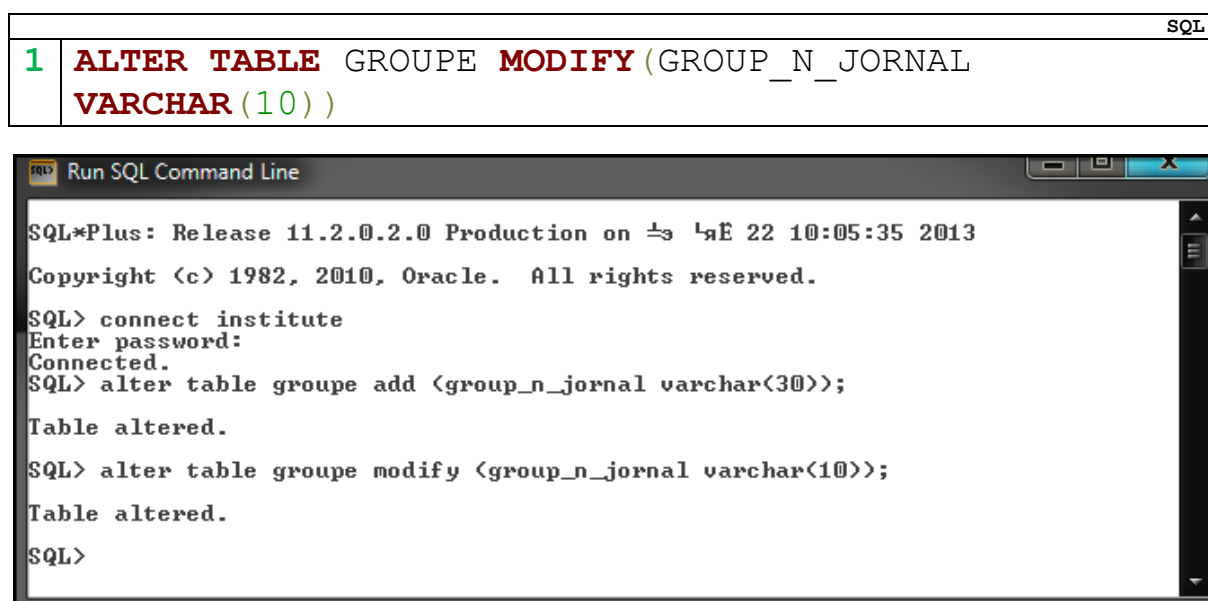
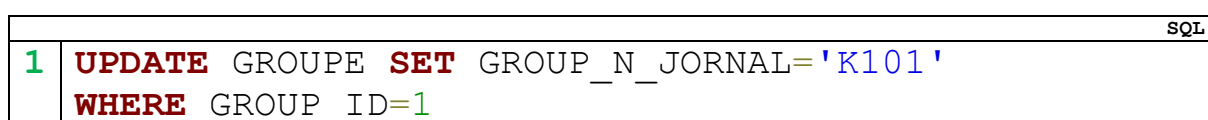


Рис. 72. Изменение типа данных поля в таблице «GROUPE»

4. Для добавления информации используем оператор «**UPDATE**»:



Аналогично добавим данные для остальных записей (рис.73).

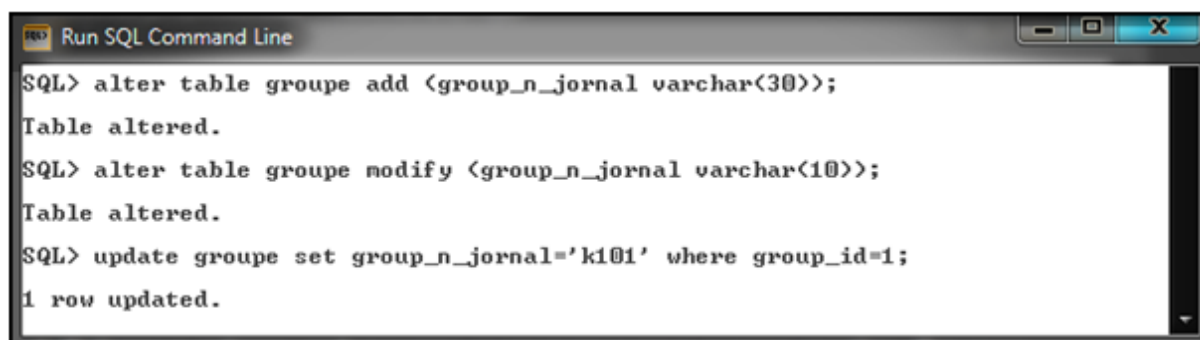


Рис. 73. Добавление данных в новый столбец таблицы «GROUPE»

1. Теперь удалим запись из таблицы «STUDENT»:

		SQL
1	DELETE FROM STUDENT WHERE STUD_ID=3	

и проверим результат (рис. 74):

		SQL
1	SELECT * FROM STUDENT	

```

SQL> update groupe set group_n_jornal='k101' where group_id=1;
1 row updated.
SQL> delete from student where stud_id=3;
1 row deleted.
SQL> select * from student;

```

STUD_ID	STUD_FIO	STUD_DAT	STUD_PHONE
1	Semenov Igor Olegovich	23.05.04	26-33-75
2	Petrov Sergey Viktorovich	14.07.03	55-55-55
4	Sidorov Vasiliy Ivanovich	10.02.03	22-24-58

STUD_ID	STUD_FIO	STUD_DAT	STUD_PHONE
5	Volkov Igor Petrovich	02.10.04	14-24-78
6	Novikov Ivan Ivanovich	12.01.03	14-98-25
7	Kravcov Mihail Sergeevich	13.03.04	45-58-41

STUD_ID	STUD_FIO	STUD_DAT	STUD_PHONE
8	Krasov Ivan Konstantinovich	06.06.05	65-55-45

```

7 rows selected.
SQL>

```

Рис. 74. Вывод данных в таблице «GROUPE»

Лабораторная работа № 6

Способы подключения к БД Oracle

Oracle – универсальная СУБД, которая позволяет подключать к себе множеством различных способов. Среди них уже известные SQL*Plus и SQL Developer и не рассмотренные еще Oracle Application Express. Осуществить работу с БД можно также и с помощью приложений, работающих на базе различных механизмов доступа к данным, например Microsoft ADO. Давайте изучим некоторые из них.

Способы подключения к БД Oracle

Запуск SQL*Plus при помощи меню

1. Самый доступный и простой способ – это использование утилиты SQL*Plus. Запустить SQL*Plus можно при помощи меню (ранее рис. 9):

После запуска появится окно работы с утилитой (рис. 75).

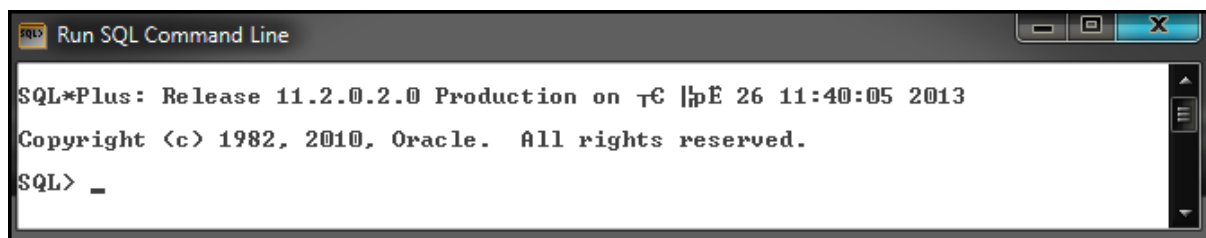


Рис. 75. Окно работы с утилитой SQL*Plus

Подключиться к БД можно как локально, так и с удаленной машины. Формат подключения:

		SQL
1	« пользователь/пароль @[/]сервер[:порт][/имя_сервиса]», где «/» - не обязательно;	

«сервер» – имя сервера или IP-адрес компьютера, на котором работает Oracle Database XE;

«порт» (не обязательно) – номер TCP-порта, по которому принимает соединения слушатель Oracle Net. Если порт не указан, то подразумевается стандартный порт номер 1521;

«имя_сервиса» (не обязательно) – это имя сервиса базы данных, к которому надо подключиться. Для Oracle Database XE имя сервиса – XE. Если «имя_сервиса» не указано, то клиент Oracle Database XE добавляет запрос к сервису по умолчанию базы данных, который настраивается в процессе установки под именем XE.

Соединимся с нашей БД «INSTITUTE». Для этого вводим команду «**CONNECT** INSTITUTE». Далее необходимо ввести пароль «INSTITUTE» (ранее рис. 22).

Для разъединения пишем «**DISCONNECT**».

2. Для соединения с БД с помощью SQL Developer необходимо создать соединение, которое также может быть как локальным, так и удаленным. Для запуска SQL Developer используем меню (рис. 76).

Пуск ⇒ Oracle Database 11g Express Edition ⇒ SQL Developer

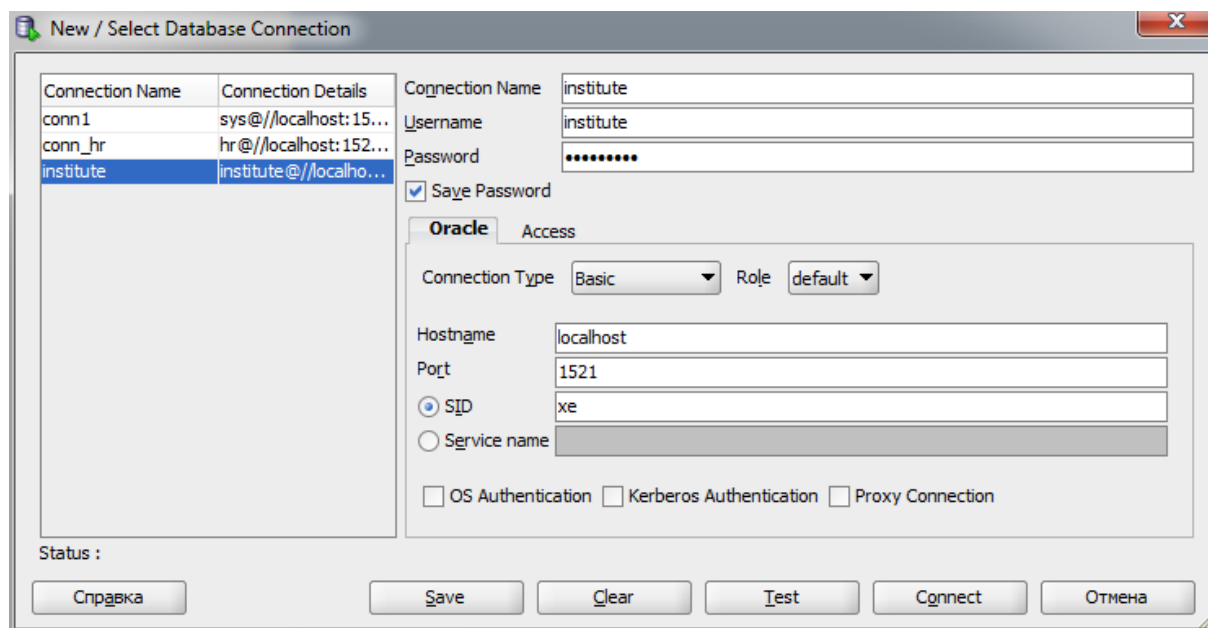


Рис. 76. Панель «Connections» утилиты SQL Developer

3. В поле «Connection name» указывается имя подключения, далее имя пользователя и пароль, который мы указывали при создании пользователя. Если мы нажимаем галочку «Save Password», SQL Developer не будет каждый раз при открытии этого подключения спрашивать пароль.

В поле «Hostname» мы указываем IP-адрес компьютера, где «localhost» – это адрес локального компьютера, при удаленном подключении указывается IP-адрес удаленной машины. «SID» – имя экземпляра Oracle, по умолчанию при установке присваивается значение «XE».

Далее нажимаем «Test» и, если нет ошибок, «Connect».

В правой панели «Connections» мы можем создать либо изменить необходимые объекты БД. В окне «Worksheet» можно выполнять любые запросы к БД и просмотреть результат (рис. 77).

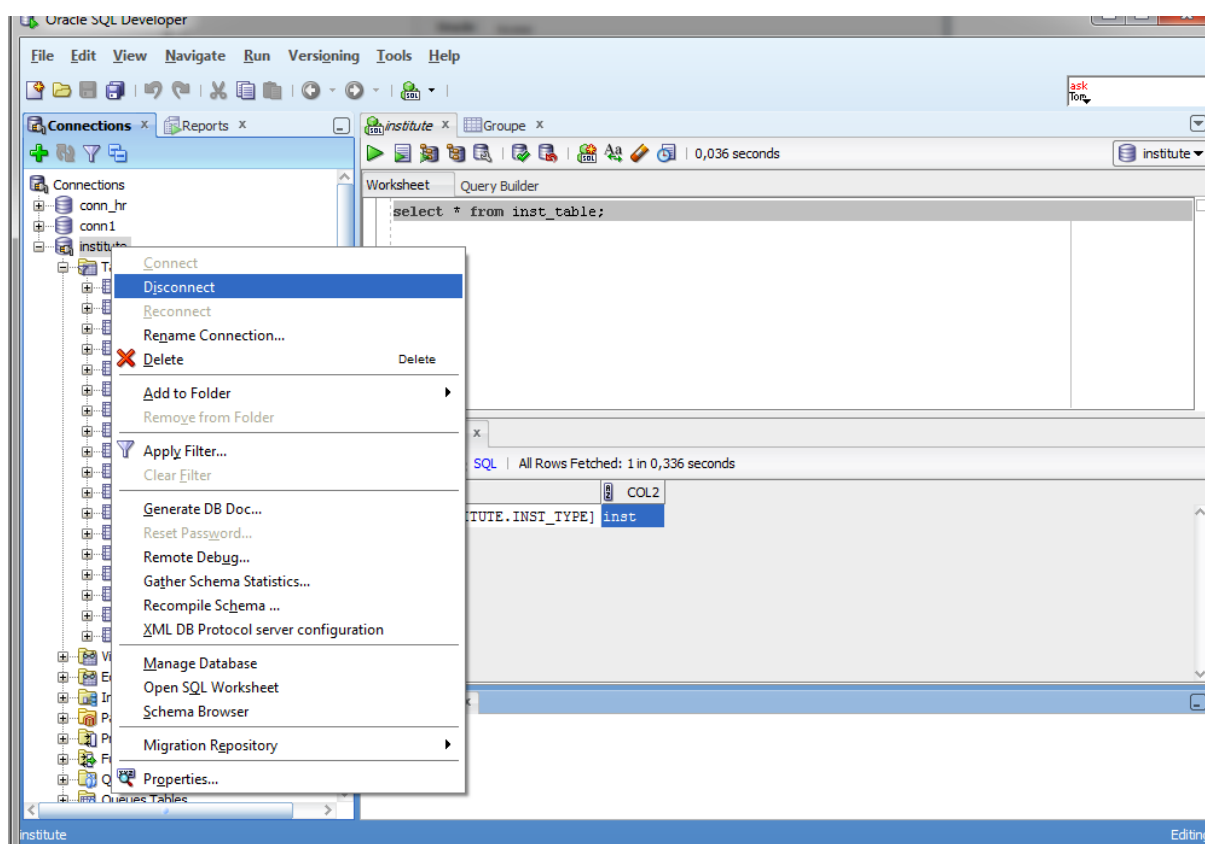


Рис. 77. Окно «Worksheet» SQL Developer

Для отключения от БД нажимаем правой кнопкой мыши в панели «Connections» на имени нашего соединения «Disconnect».

Запуск с помощью Oracle Application Express (Oracle APEX)

1. Еще один популярный способ соединения с Oracle – это Oracle Application Express. Oracle Application Express (Oracle APEX), ранее известный как HTML DB, представляет собой среду для быстрого создания и разворачивания веб-приложений, которая позволяет разрабатывать более быстрые и надежные профессиональные приложения, используя для этого лишь только веб-браузер и минимальный опыт программирования. Oracle Application Express сочетает в себе высокую производительность, безопасность, интегрируемость и масштабируемость корпоративных БД, разработанных на основе веб-технологий, с простотой использования, доступностью и гибкостью персональных БД. Разработка и запуск программ на Application Express осуществляется посредством стандартного веб-браузера и не требует установки и обслуживания какого-либо клиентского ПО. Для подключения с помощью APEX мы используем пункт меню

«Пуск ⇒ Oracle Database 11g Express Edition ⇒ Get Started».

Откроется окно работы с Oracle через веб-браузер (рис. 78).

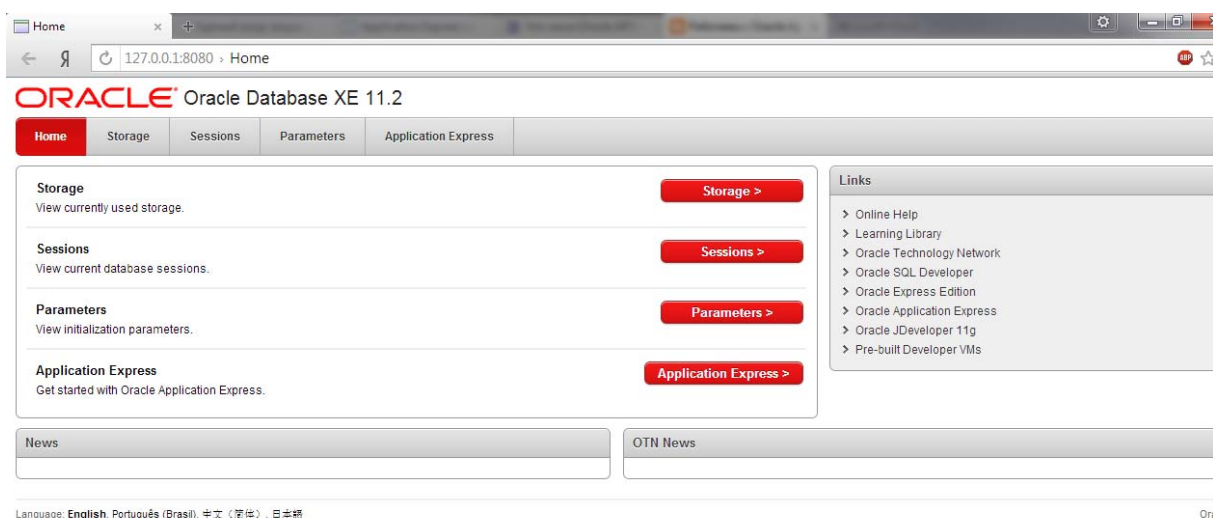


Рис. 78. Окно управления ORACLE через веб-интерфейс

2. Нажимаем кнопку «Application Express», откроется окно авторизации, в котором необходимо указать пользователя, обладающего правами «DBA». Мы зайдем под пользователем «SYSTEM», пароль которого мы указывали при установке Oracle (рис. 79).

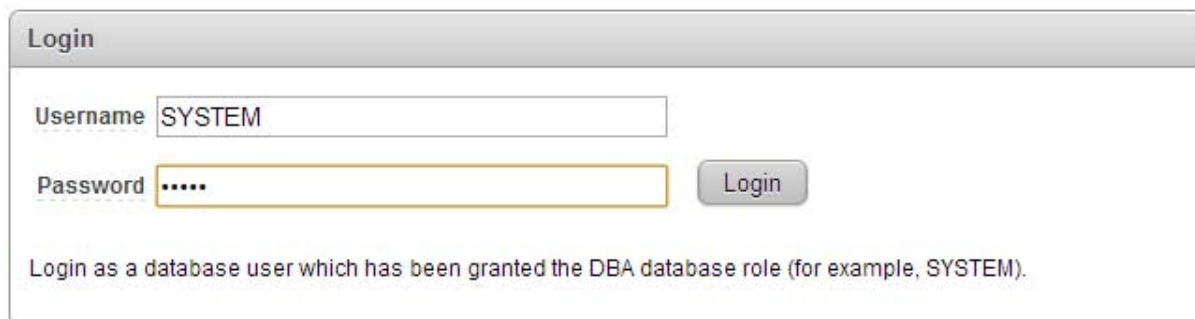


Рис. 79. Окно авторизации «Application Express»

3. Для работы с нашей БД нам необходимо создать рабочее пространство (рис. 80). Для этого в открывшемся окне указываем «Database user – Use exiting», в поле «Database Username» созданного нами ранее пользователя «INSTITUTE» «Application Express Username» указываем имя, которое будет использоваться в APEX, указываем пароль «INSTITUTE» и подтверждаем его. Далее нажимаем кнопку «Create Workspace».

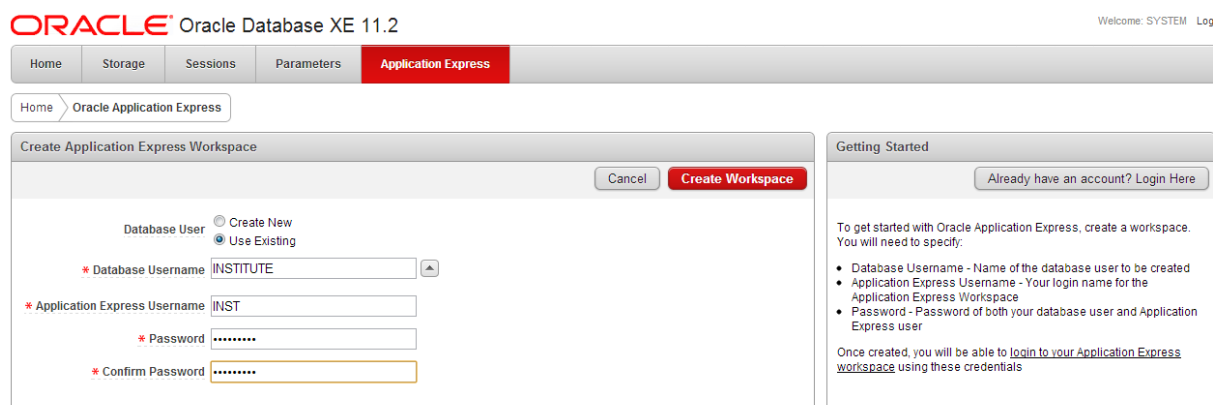


Рис. 80. Создание нового рабочего пространства

4. В появившемся окне авторизации указываем имя нашего рабочего пространства, имя пользователя и пароль (рис. 81). Нажимаем «Login».

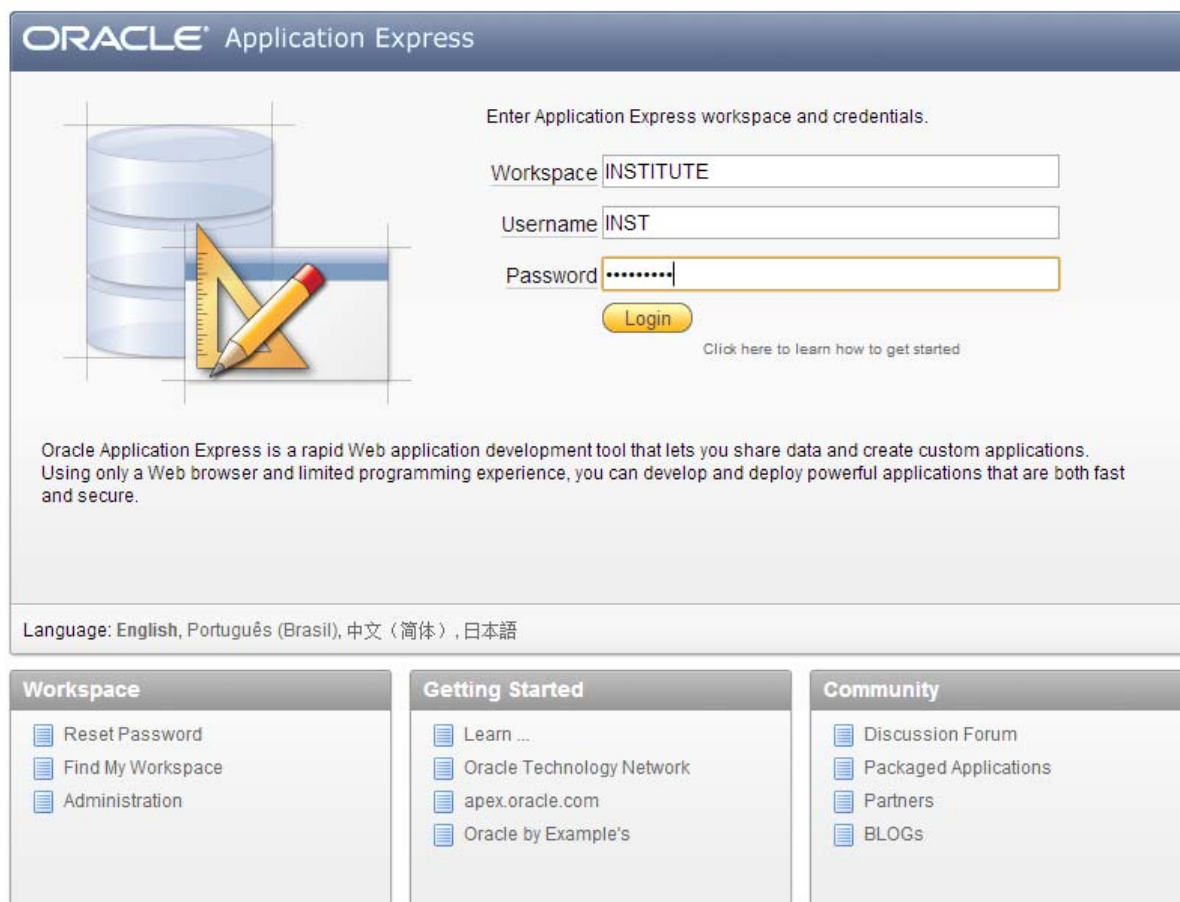


Рис. 81. Окно авторизации «Application Express»

5. Открывается окно работы с БД (рис. 82). Основные инструменты Application Express:

Application Builder – инструмент, позволяющий разрабатывать динамические веб-приложения, управляемые БД;

SQL Workshop – инструмент для просмотра содержимого БД, визуального построения запросов, в том числе и сложных SQL запросов;

Team Development – инструмент, позволяющий совместную разработку приложений;

Administration – средства администрирования нашего рабочего пространства.



Рис. 82. Окно работы с БД в «Application Express»

6. Для проверки работы внесем запись в таблицу «GROUPE». Для этого нажимаем кнопку «SQL Workshop» (рис. 83). Откроется окно, в котором можно создавать различные SQL-запросы вручную, использовать утилиту автоматического создания запросов, конвертировать данные и так далее. Мы воспользуемся кнопкой «Object Browser».

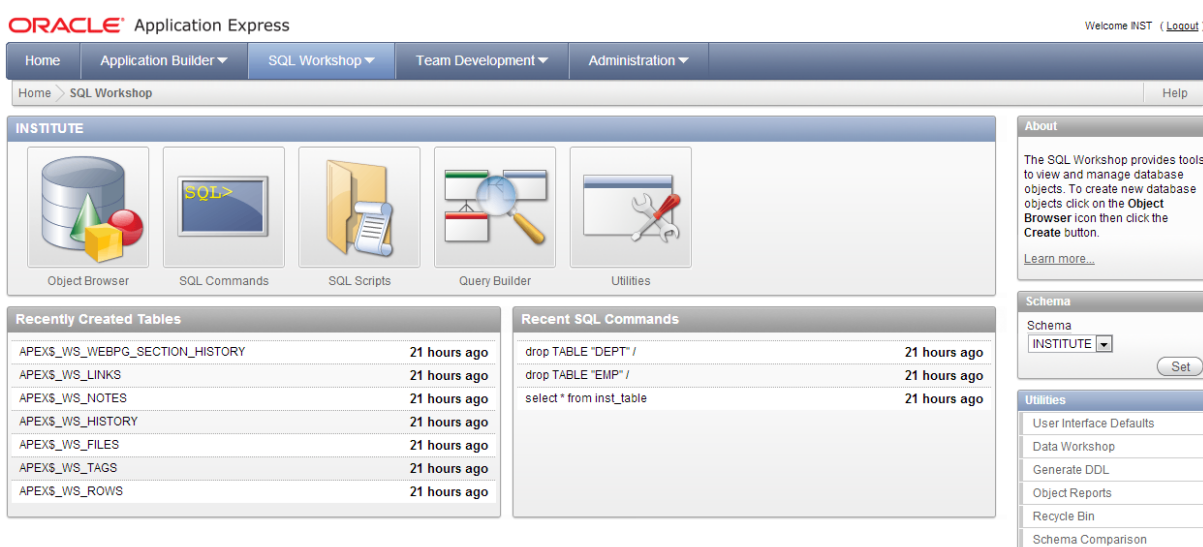


Рис. 83. Рабочее пространство SQL в «Application Express»

7. В левом списке выбираем таблицу «GROUPE», в появившемся справа меню «Data» и «Insert row» (рис. 84).

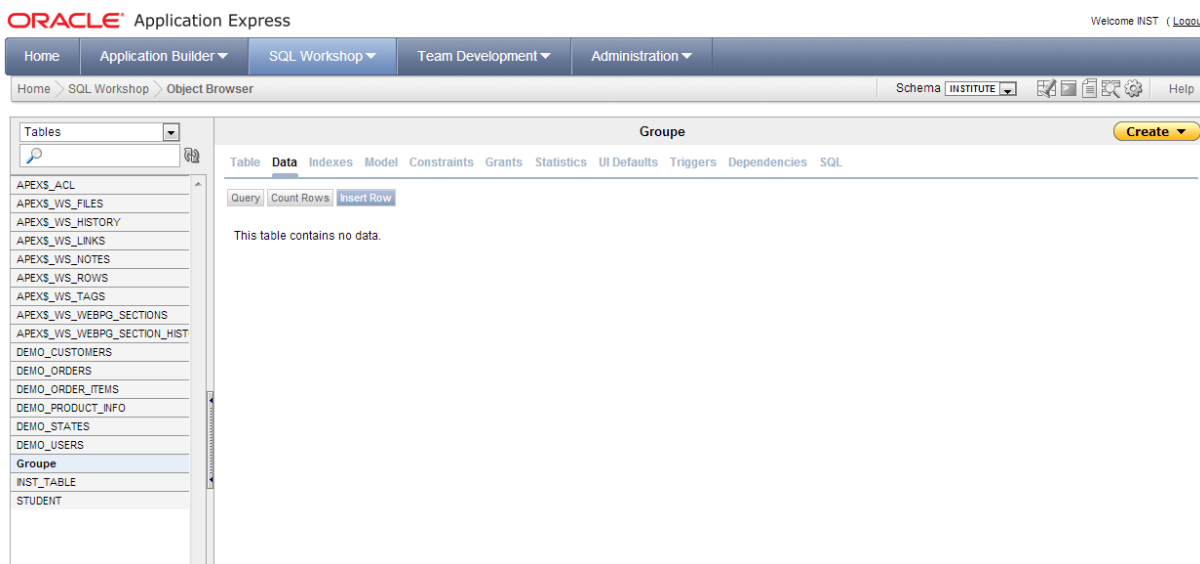


Рис. 84. Представление таблицы «GROUPE» в «Application Express»

8. Вводим в поля данные и нажимаем «Create», наша информация будет внесена в БД (рис. 85).

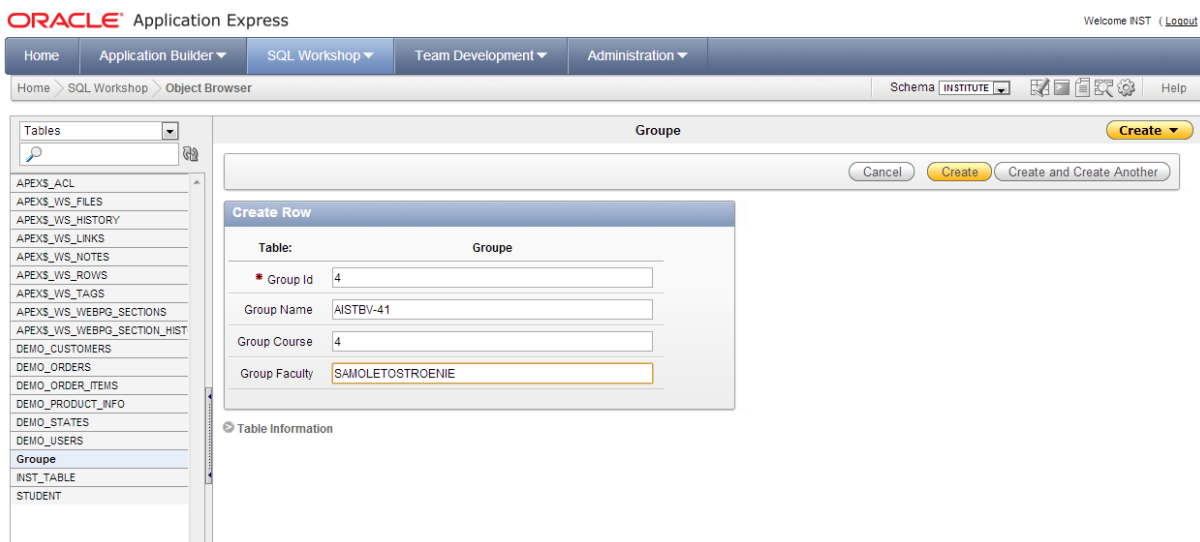


Рис. 85. Заполнение таблицы «GROUPE» в «Application Express»

9. Для выхода из АРЕХ в правом верхнем углу нажимаем «Logout».

Лабораторная работа № 7

Создание приложений

На прошлом занятии (лабораторная работа № 6) мы изучили некоторые способы подключения к Oracle. На этом занятии рассмотрим еще один – подключение к БД с помощью приложений, разрабатываемых прикладными программистами. Хорошо спроектированная БД еще не все для нормального функционирования БД. Важным шагом является создание понятного пользователям и удобного интерфейса. Для изучения данного вопроса используем пробную версию Delphi XE 4.

Создание приложений

1. Запустим Delphi. В появившемся окне нажимаем «*Project ⇒ New Project ⇒ VCL Form Application*».

Появится форма нашей программы. Сохраним программу «Save as» с именем «First_prog» (рис. 86).

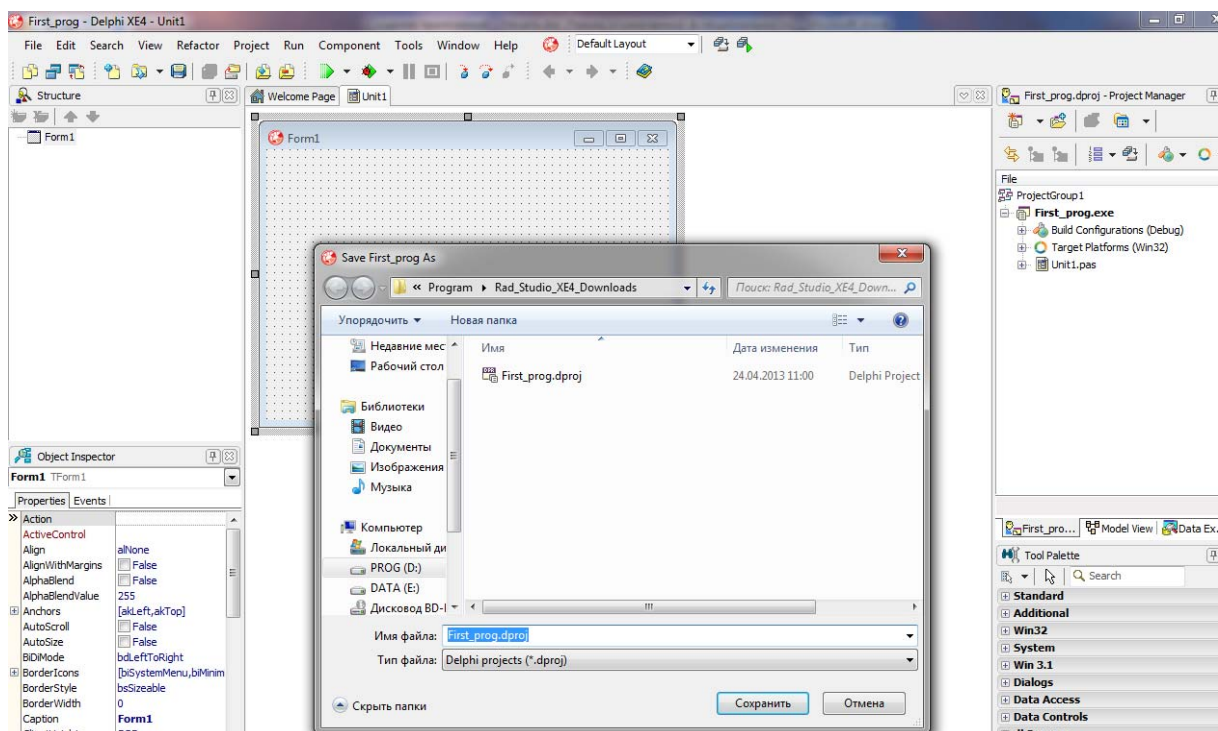


Рис. 86. Меню сохранения проекта «Save as» в Delphi XE 4

2. Положим на нашу форму с закладки «Data Controls» палитры компонентов визуальный компонент для обзора данных «DB Grid», с закладки «Data Access» – компонент «Data Source», затем с закладки «db Go» надо добавить компоненты «ADO Connection» и «ADO Query» (рис. 87).

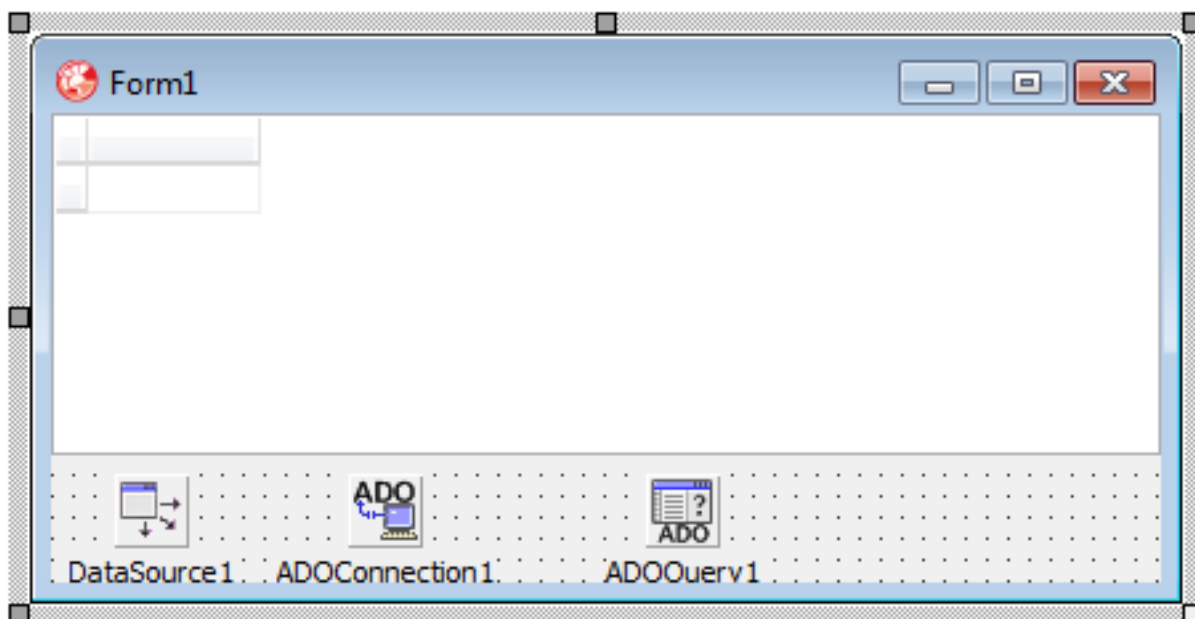


Рис. 87. Дизайн формы приложения в Delphi XE 4

3. Настроим «ADO Connection1» (рис. 88). Этот компонент служит для задания пути к базе данных и способа открытия таблицы (задания драйвера базы данных).

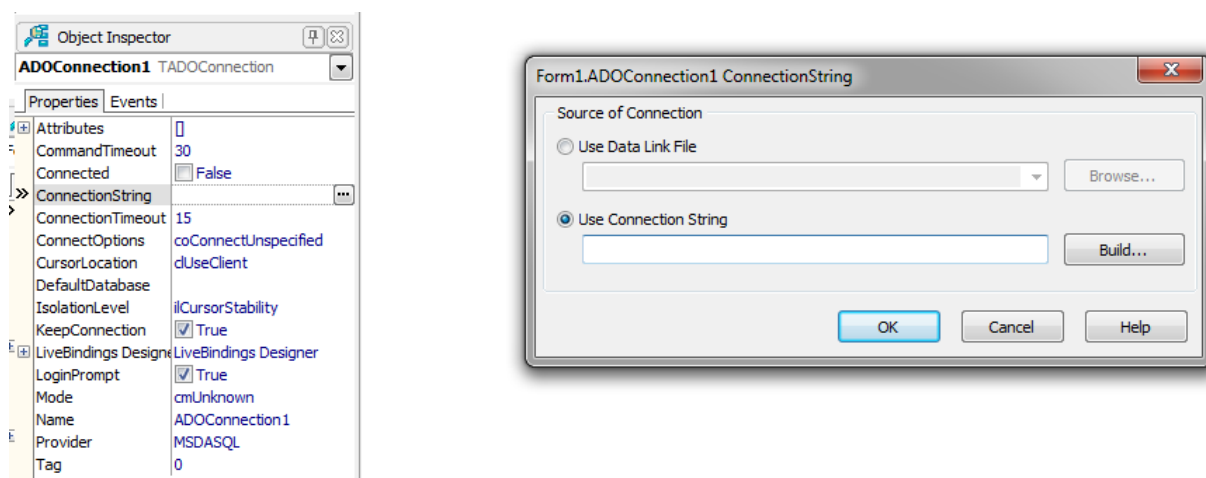


Рис. 88. Свойства подключения к Oracle в Delphi XE 4

В принципе, его можно было не создавать, а воспользоваться предоставленным для этого свойством компонента «ADO Query». Но корректнее воспользоваться именно компонентом «ADO Connection», потому что из одной базы данных надо будет открыть несколько таблиц.

Настроим путь к базе данных. Для этого в свойстве «Connection String» компонента «ADO Connection1» надо щелкнуть на кнопке «Build...» строки «Use Connection String» ().

4. Откроется окно выбора поставщика данных. Выберем драйвер «Oracle Provider for OLE DB» и нажимаем «Далее» (рис. 89).

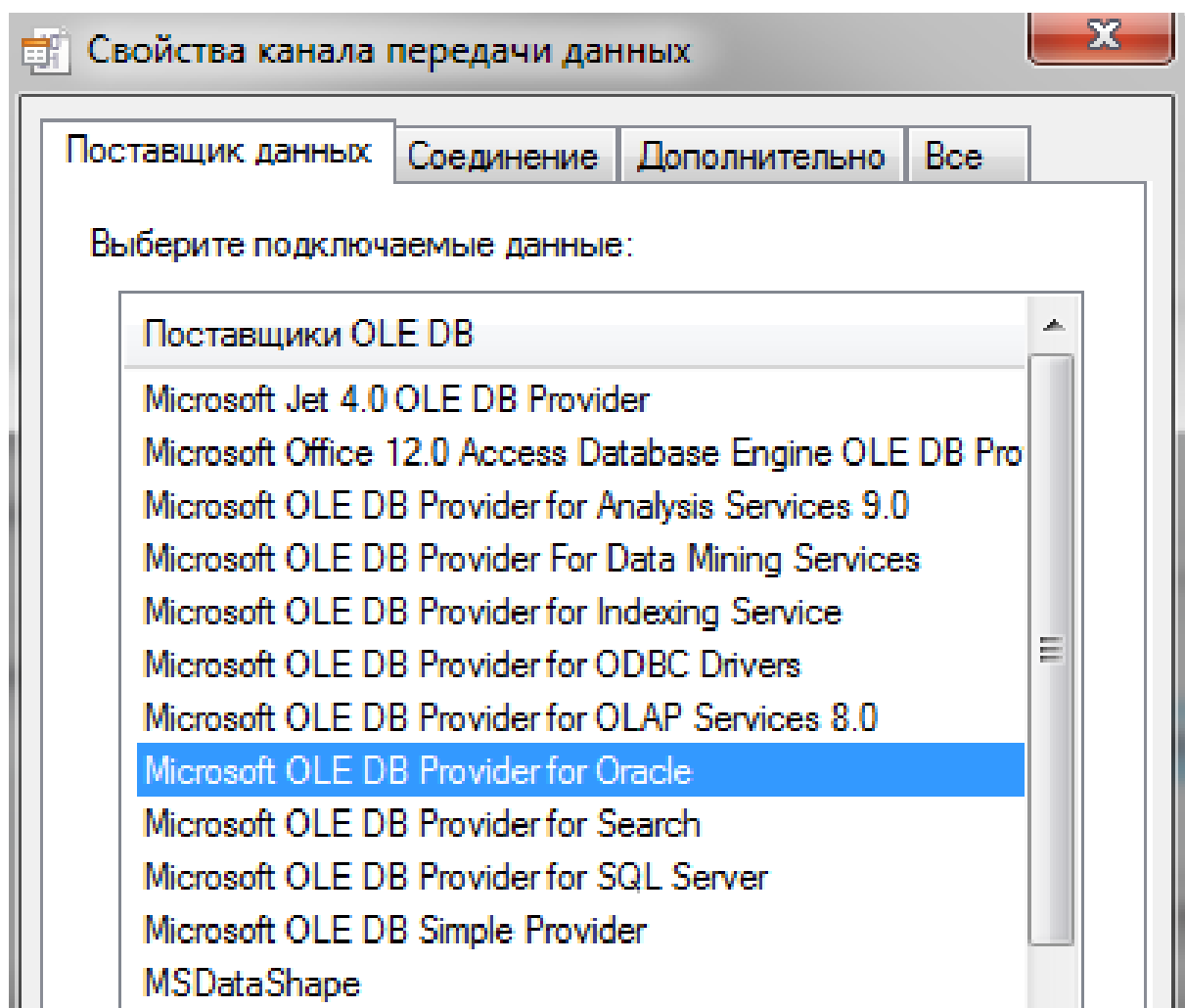


Рис. 89. Выбор драйвера «Oracle Provider for OLE DB» из списка

5. Откроется окно свойств соединения. Здесь указываем имя сервера «localhost», имя пользователя и пароль нашего пользователя «INSTITUTE». Нажимаем «Проверить соединение». Далее «ОК» (рис. 90).

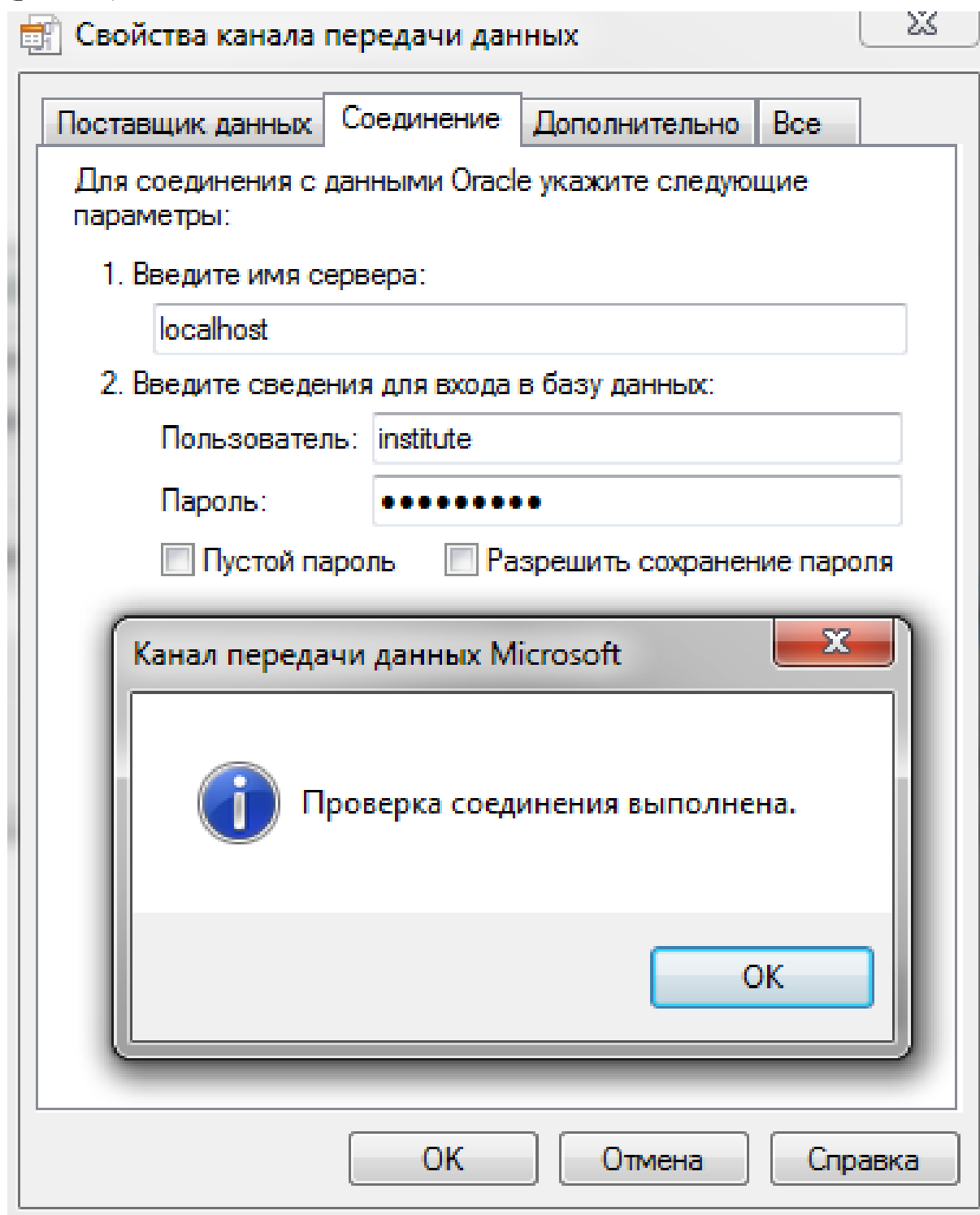


Рис. 90. Проверка успешности соединения приложения с Oracle

6. В свойстве «Connection» компонента «ADO Query1» указываем «ADO Connection1». В свойстве «DataSet» компонента «DataSource1» указываем «ADO Query1». В свойстве «Data Source» компонента «DB Grid1» выбираем «Data Source1».

Все компоненты связаны, теперь можно писать запрос к БД (рис. 91). В свойстве «SQL» компонента «ADO Query1» пишем:

	SQL
1	SELECT group_name, group_course, group_faculty FROM Groupe ORDER BY group_id

Нажимаем «True» в свойстве «Active».

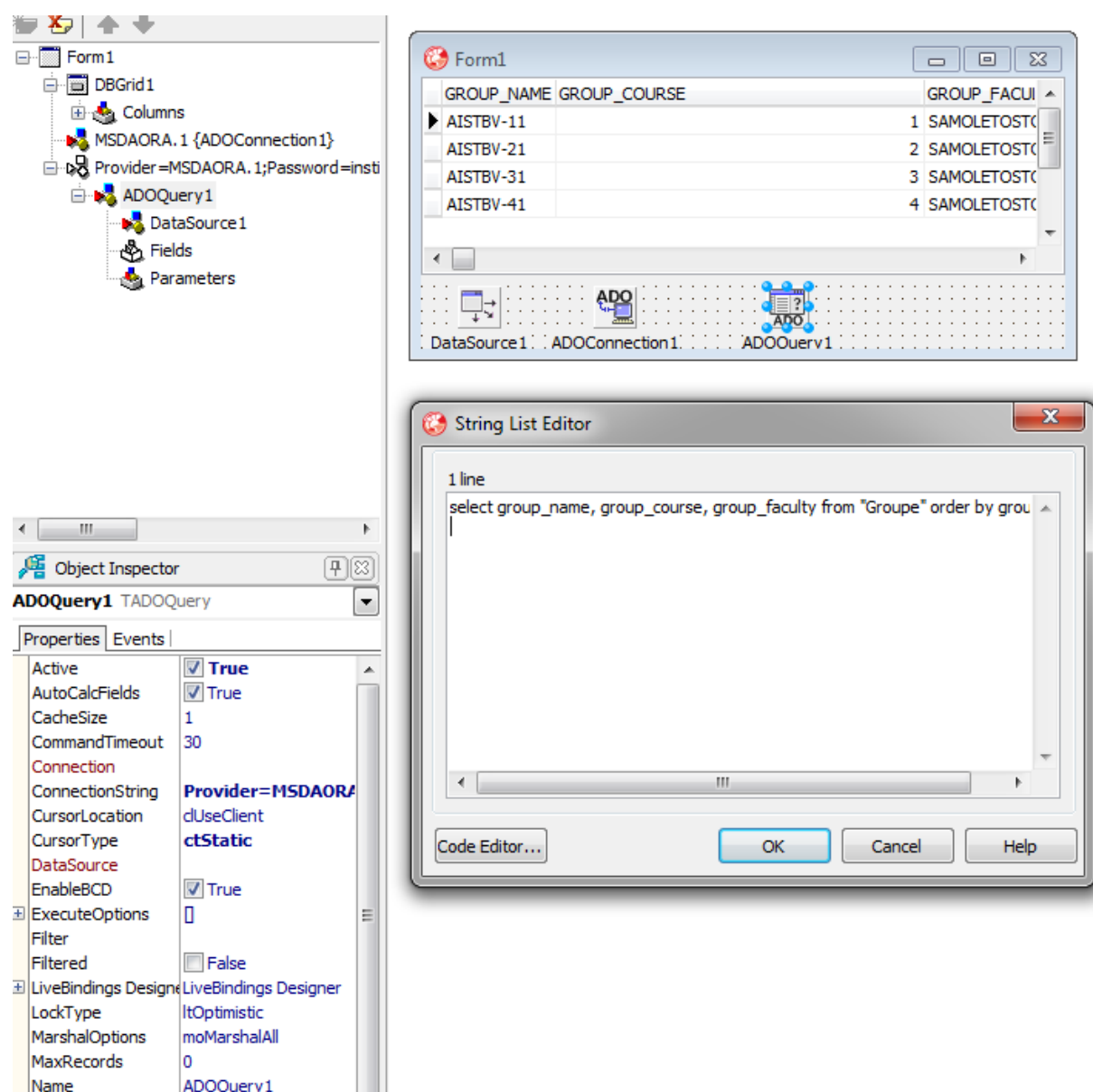


Рис. 91. Выстраивание связи компонентов для работы с БД в Delphi XE 4

7. Для проверки работоспособности программы нажимаем «Run», и наша программа компилируется и запускается (рис. 92).

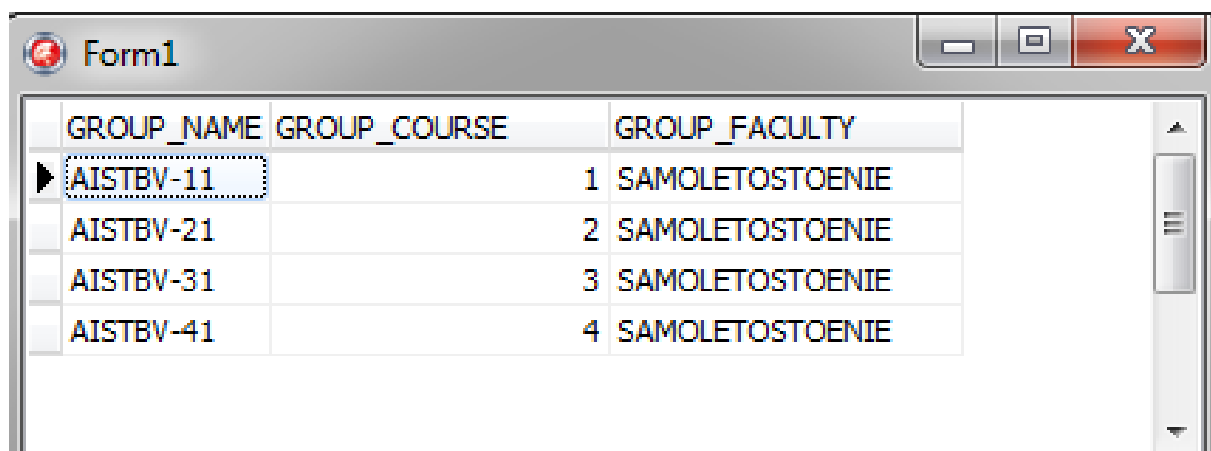


Рис. 92. Запуск основной формы приложения для работы с БД

8. Здесь мы можем вносить изменения в данные, так как в свойствах компонента «DB Grid1» не указывали «Read Only». Напишем в поле «GROUP_FACULTY» группы «AISTBV-41» – «EDIT» (рис. 93). Сделанные изменения сохранятся в БД.

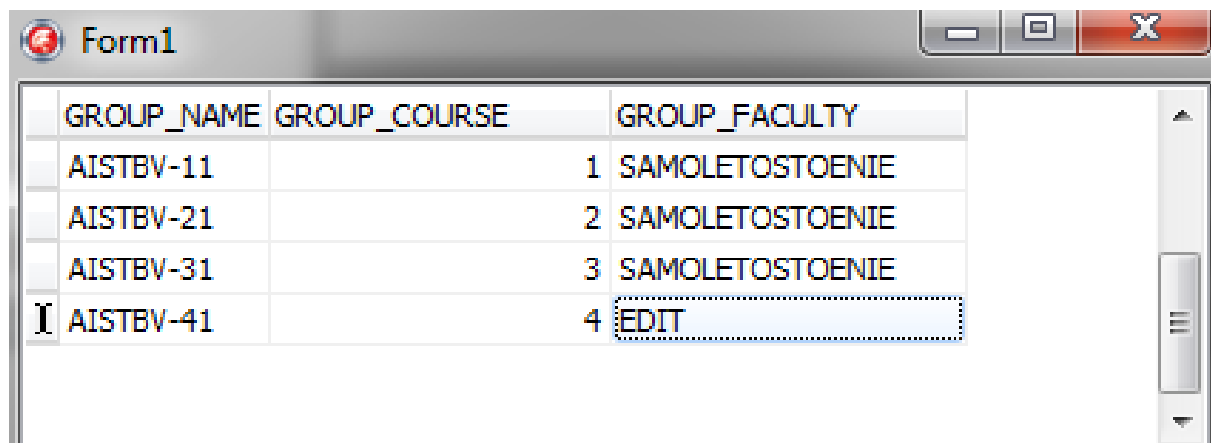


Рис. 93. Доступ к изменению данных в БД

Мы создали простой пример программы, которая взаимодействует с Oracle.

Лабораторная работа № 8

Oracle SQL Developer для работы с запросами Oracle: преимущества и недостатки, автоматическое построение запросов в Query Builder

Одно из наиболее рекомендованных средств для работы с запросами SQL и программными единицами PL/SQL – это Oracle SQL Developer. Это – «родное» программное средство от корпорации Oracle, которое появилось относительно недавно (в 2005 году).

Отметим его самые важные отличия:

- Oracle SQL Developer – полностью бесплатное приложение. Несмотря на это, по своим возможностям и удобству оно не уступает «платным» средствам (PL/SQL Developer, Toad, SQLNavigator);
- Oracle SQL Developer написан на языке Java и может использоваться (с одним и тем же графическим интерфейсом) как под Windows, так и под другими операционными системами, например, Linux;
- Oracle SQL Developer не требует установки на компьютер. Все пользовательские настройки в нем хранятся в файлах XML;
- Oracle SQL Developer может использоваться для создания запросов к другим системам управления базами данных, например, Microsoft SQL Server, Microsoft Access, MySQL и Times Ten.

На многих предприятиях возможность использования единообразного программного средства для создания запросов к разным источникам может быть очень удобной. Правда, отметим, что Oracle SQL Developer знает про базы данных Microsoft далеко не все, и лучше использовать его при создании простых запросов к базам данных, отличных от Oracle.

К недостаткам SQL Developer можно отнести несколько непривычный интерфейс (основанный на графических компонентах Java) и большую ресурсоемкость (особенно с точки зрения оперативной памяти). Кроме того, в Windows все настройки SQL Developer (включая,

например, определения пользовательских отчетов и сниппетов) хранятся не в реестре, а в файлах XML в профиле пользователя, что также необходимо учитывать.

Первое, что нужно сделать при работе с Oracle SQL Developer – создать новое соединение с источником данных. Это можно сделать как из меню File | New, так и из окна Connection (нужно щелкнуть правой кнопкой мыши по контейнеру Connections и в контекстном меню выбрать кнопку New). Откроется окно создания нового соединения (рис. 94).

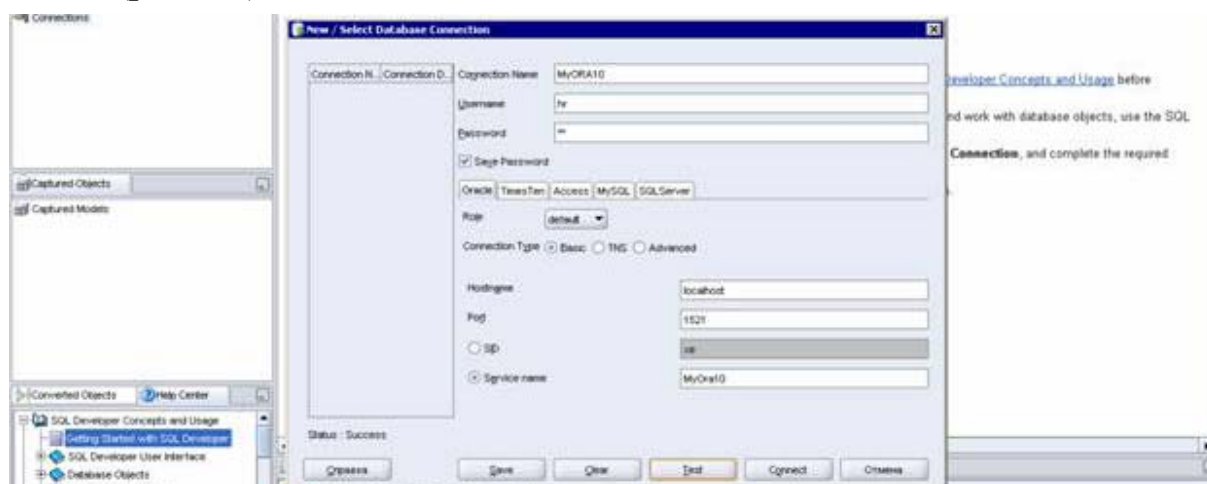


Рис. 94. Создание нового соединения в Oracle SQL Developer

После занесения параметров нового соединения в этом окне можно воспользоваться кнопкой Test, чтобы убедиться, что все параметры введены правильно. После этого параметры подключения можно сохранить для последующего использования (кнопка Save) и выполнить вход на сервер (кнопка Connect). Откроется окно редактирования кода SQL, в котором можно набирать и выполнять код на SQL и PL/SQL (официально оно называется SQL Worksheet). Предусмотрен также графический построитель запросов, который можно вызвать, если щелкнуть правой кнопкой мыши по пустому месту в редакторе кода и в контекстном меню выбрать Query Builder (рис. 95).

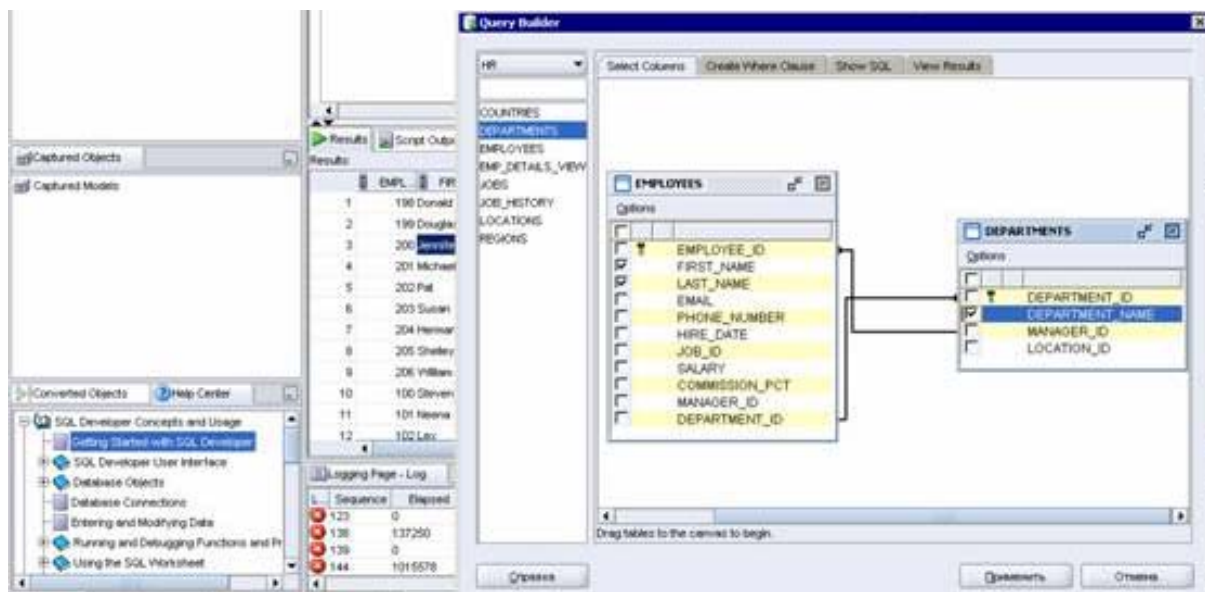


Рис. 95. Графический построитель запросов Query Builder
Отметим наиболее важные возможности Oracle SQL Developer.

1. При помощи контекстного меню для объектов в окне Connections можно выполнить различные административные операции, например, по внесению изменений. При этом для таких операций можно сгенерировать код команды SQL (просмотреть его можно на вкладке SQL соответствующего окна). Отдельно отметим встроенные возможности импорта и экспорта данных для таблиц. При импорте поддерживается единственный формат – Excel, а при экспорте – множество форматов, в том числе Excel (xls), CSV, XML, SQL (в виде набора команд INSERT).

2. В Oracle SQL Developer реализованы возможности прямого редактирования записей в таблицах баз данных. Для этого нужно в контекстном меню для таблицы в окне Connections выбрать команду Edit, а затем перейти на вкладку Data.

3. В Oracle SQL Developer предусмотрены средства для отладки кода программных модулей PL/SQL (включая точки останова, пошаговое выполнение, просмотр значений переменных и т.п.). Эти средства в основном доступны из меню Debug.

4. Заранее заготовленные блоки программного кода доступны из окна Snippets (это окно можно открыть при помощи меню View). Такие блоки заготовлены как для встроенных функций Oracle SQL, так и для других синтаксических конструкций (операторов условного перехода, циклов, курсоров и т.п.). Набор сниппетов является расширяемым: вы вполне можете добавлять в него блоки программного кода.

5. Для того чтобы было проще переносить программный код, написанный в расчете на выполнение в SQL Plus, в Oracle SQL Developer предусмотрена поддержка многих синтаксических конструкций SQL Plus (например, обращение к внешним файлам скриптов при помощи символов @ и @@). Однако поддерживаются далеко не все специальные команды SQL Plus.

Если объектов в базе данных много (обычная ситуация для реальных баз данных), и просматривать их все в окне Connection трудно, можно воспользоваться специальным окном поиска объектов. Для этого нужно в меню View воспользоваться командой Find DB Object, затем в открывшемся окне выбрать соединение, ввести имя объекта (можно использовать подстановочные символы) и нажать на Enter. Для объектов в списке результатов из контекстного меню доступны те же возможности, что и для объектов в окне Connections.

6. Во второй вкладке окна Connections предусмотрены средства работы с отчетами. Правда, эти отчеты предназначены в основном для администраторов баз данных. Можно использовать как готовые отчеты, так и создавать свои. В отчетах предусмотрено использование параметров (например, для отчета, который возвращает все отчеты, принадлежащие какому-либо пользователю, нужно указать его имя).

В Oracle SQL Developer предусмотрены средства переноса информации между различными источниками данных, например, из базы данных SQL Server в Oracle. Средства для выполнения такого переноса сосредоточены в меню Migration.

Лабораторная работа № 9

Проектирование таблиц

Таблицы

База данных – это набор файлов. Несмотря на то, что они хранятся на жестком диске, напрямую обратиться к ним нельзя. Вернее сказать, можно, но это приведет к порче информации. Доступ к информации в этих файлах обеспечивает программа-сервер базы данных.

Каждый файл определенным образом размечен. Это делается с одной стороны для более эффективной работы программы-сервера, а с другой – для того, чтобы хранить разные данные отдельно друг от друга. Каждому логически обоснованному массиву информации выделяется участок памяти. В свою очередь он разбивается на еще более мелкие участки – записи. Поскольку информация в массиве однородна, то записи имеют одинаковую структуру.

Логически обоснованный массив информации наглядно можно представлять в форме таблицы (рис. 96). Строки таблицы – это записи. Несколько таблиц – это база данных.

Имя таблицы: **ПРИМЕР**

Строка	Колонка 1	Колонка 2	Колонка 3	Колонка 4
	Число	Строка	Дата	Число

	345	Безымянный друг	09.12.1974	1 365 000.00

Рис. 96. Структура табличного представления массива информации

У таблицы есть имя, колонки и строки. Имя нужно для того, чтобы можно было в базе данных отличить один массив информации от другого.

Головка таблицы («шапка» таблицы), или, другими словами, список колонок – это всего лишь описание порядка следования значений в строке. Поскольку для каждой строки эта последовательность одна и та же, принято описывать структуру строк через определение колонок.

Для каждой колонки задается имя и тип данных (заголовок колонки, боковик колонки и подзаголовок колонки). Основные типы: число, строка и дата.

Количество колонок в таблице фиксировано, а строк может быть сколько угодно или не быть вовсе.

Таблицы для учебного примера

Рассмотрим форму счета из учебного примера (таблица 1). Зададимся вопросом: «Сколько нужно таблиц для хранения информации из этого счета в базе данных?». Явно не одна.

Таблица 1

Счет:		№ 139 от 12.09.2003						
Продавец:		ООО "Балтиец"						
Покупатель:		Морейнис А. Н.						
Адрес доставки:		ул. Рабочая канавка, д.12, кв.37						
								Валюта: руб.
Номер п/п	Наименование товаров (услуг)	Единица изме- рения	Количе- ство	Цена за единицу измерения	Стоимость товаров (услуг) без налога	Нало- говая ставка	Сумма налога	Стоимость товаров (услуг), всего с учетом налога
1.	Банки стеклянные	шт.	7.00	3.25	22.75	18%	4.10	26.85
2.	Горшки цветочные	шт.	2.00	12.50	25.00	18%	4.50	29.50
3.	Грунт для цветов	кг.	160.00	32.40	5 184.00	18%	933.12	6 117.12
4.	Пробка медицинская	шт.	45.00	3.05	137.25	10%	13.73	150.98
Итого:					5 369.00		955.45	6 324.45
	Услуги доставки				4.70	18%	0.85	5.55
	Услуги упаковки				17.90	18%	3.22	21.12
Всего:					5 391.60		959.52	6 351.12

Во-первых, у документа есть «шапка» и табличная часть. Уже как минимум – две таблицы.

Во-вторых, в табличной части можно выделить строки с товарами и строки с услугами. Строки с товарами имеют номер по порядку, единицу измерения, цену, количество. Строки с услугами ничего этого не имеют. Колонки не заполнены.

Получается, что для хранения счета потребуется три таблицы: для «шапки» счета, для строк с товарами, для строк с услугами.

Таблица для «шапки» счета

Займемся таблицей для «шапки» счета. Любая таблица имеет имя. Для этой тоже надо придумать. Пусть будет EXDOC (таблица 2).

Таблица 2

Заголовки счета

Имя таблицы: <u>EXDOC</u>						
Номер счета	Дата счета	Покупатель	Адрес доставки	Стоимость товаров, руб.	Налог, руб.	Всего, руб.
DOC_ND	DOC_DD	CUSTOMER_N	ADDRESS	TOTAL_COST	TOTAL_TAX	TOTAL_DOC
number(6)	date	varchar2(60)	varchar2(120)	number(12, 2)	number(12, 2)	number(12, 2)
...
...
139	12.09.2003	Морейнис А. Н.	ул. Рабочая канавка, д. 12, кв. 37	5 391.60	959.52	6 351.12
...
...

Далее определяем колонки, которые будут в этой таблице. Берем «шапку» счета и выписываем из нее поля. Обращаю внимание, что имя продавца пропущено. Значение во всех строках будет одно и то же, а посему его можно не хранить. Зато включаем в таблицу колонки с общими итогами по документу (они пригодятся, когда будем изучать PL/SQL).

Кстати, это достаточно удобно. Вот почему. Для расчета общих итогов нужно будет обращаться и к таблице с товарами, и к таблице с услугами. Многократное обращение к таблицам можно избежать, если в таблицу с «шапкой» счета сейчас добавить колонки для хранения уже рассчитанных значений. Теперь каждой дадим имя, которое состоит из латинских букв. Укажем тип данных, которые будут в них храниться. В ORACLE есть следующие типы данных: для хранения чисел – NUMBER; для дат – DATE; для символьных строк - VARCHAR2.

Есть колонки для хранения целых чисел NUMBER (6) и для чисел с двумя знаками после запятой NUMBER(12, 2). У колонок, которые хранят символьные данные, разная размерность. Имя покупателя должно быть меньше 60 символов. Адрес доставки должен не превышать 120.

Таблицы для позиций с товарами и услугами

Займемся таблицей для позиций товаров.

Придумаем имя. Пусть будет EXPOS. Выпишем из формы счета поля – будущие колонки. Каждой дадим имя. Внимательно рассмотрим форму счета, для каждой колонки определяем тип данных.

Для проверки заполним таблицу данными из счета (таблица 3).

Таблица 3

Данные из счета

Имя таблицы: EXPOS								
Номер п/п	Наименование товара	Единица измерения	Количество	Цена за единицу, руб.	Стоимость товаров, руб.	Налоговая ставка, %	Налог, руб.	Стоимость товаров, руб.
POS_NO	GOOD_N	ED	QUANT	PRICE	COST	TAX	SUM_TAX	SUM_POS
number(5)	varchar2(60)	varchar2(8)	number(9, 2)	number(9, 2)	number(12, 2)	number(5, 2)	number(12, 2)	number(12, 2)
...
...
1	Банки стеклянные	шт.	7.00	3.25	22.75	18.00	4.10	26.85
2	Горшки цветочные	шт.	2.00	12.50	25.00	18.00	4.50	29.50
3	Грунт для цветов	кг.	160.00	32.40	5 184.00	18.00	933.12	6 117.12
4	Пробка медицинская	шт.	45.00	3.05	137.25	10.00	13.73	150.98
...
...

Ответим на вопрос: «Как в этой таблице записать строки другого счета?» То есть в форму будущей таблицы вписать данные другого счета. Строки первого счета выделим одним цветом, строки второго другим.

Когда строки раскрашены, можно отделить строки одного счета от другого (таблица 4). Но ORACLE не умеет раскрашивать записи в таблицах. Создав таблицу с таким набором колонок, не только ORACLE, но и мы сами, не сможем отделить строки одного счета от другого.

Таблица 4

Представление счетов

Имя таблицы: EXPOS								
Номер п/п	Наименование товара	Единица измерения	Количество	Цена за единицу, руб.	Стоимость товаров, руб.	Налоговая ставка, %	Налог, руб.	Стоимость товаров, руб.
POS_NO	GOOD_N	ED	QUANT	PRICE	COST	TAX	SUM_TAX	SUM_POS
number(5)	varchar2(60)	varchar2(8)	number(9, 2)	number(9, 2)	number(12, 2)	number(5, 2)	number(12, 2)	number(12, 2)
...
...
1	Банки стеклянные	шт.	7.00	3.25	22.75	18.00	4.10	26.85
2	Горшки цветочные	шт.	2.00	12.50	25.00	18.00	4.50	29.50
3	Грунт для цветов	кг.	160.00	32.40	5 184.00	18.00	933.12	6 117.12
4	Пробка медицинская	шт.	45.00	3.05	137.25	10.00	13.73	150.98
...
1	Грунт для цветов	кг.	1.00	32.40	32.40	18.00	5.83	38.23
2	Горшки цветочные	шт.	1.00	12.50	12.50	18.00	2.25	14.75
...
...

Для решения этой проблемы цвет нужно заменить колонками. Добавим колонки с реквизитами счета. Это «Номер счета» и «Дата счета». Имена колонок и тип данных берем из таблицы с шапками счетов EXDOC. Теперь можно сказать однозначно, какому счету принадлежит та или иная позиция (таблица 5).

Таблица 5

Модифицированное представление счетов

Имя таблицы: **EXPOS**

Номер счета	Дата счета	Номер п/п	Наименование товара	Единица измерения	Количество	Цена за единицу, руб.	Стоимость товаров, руб.	Налоговая ставка, %	Налог, руб.	Стоимость товаров, руб.
DOC_ND	DOC_DD	POS_NO	GOOD_N	ED	QUANT	PRICE	COST	TAX	SUM_TAX	SUM_POS
number(6)	date	number(5)	varchar2(60)	varchar2(8)	number(9, 2)	number(9, 2)	number(12, 2)	number(5, 2)	number(12, 2)	number(12, 2)
	
	
139	12.09.2003	1	Банки стеклянные	шт.	7.00	3.25	22.75	18.00	4.10	26.85
139	12.09.2003	2	Горшки цветочные	шт.	2.00	12.50	25.00	18.00	4.50	29.50
139	12.09.2003	3	Грунт для цветов	кг.	160.00	32.40	5 184.00	18.00	933.12	6 117.12
139	12.09.2003	4	Пробка медицинская	шт.	45.00	3.05	137.25	10.00	13.73	150.98
	
152	17.09.2003	1	Грунт для цветов	кг.	1.00	32.40	32.40	18.00	5.83	38.23
152	17.09.2003	2	Горшки цветочные	шт.	1.00	12.50	12.50	18.00	2.25	14.75
	
	

Осталась третья таблица: строки для услуг.

С ней все просто. Придумываем имя. Определяем колонки. При- своим латинские имена колонкам. Указываем тип, размерность. До- бавляем две колонки с реквизитами счета. Они устранят неоднознач- ности, обеспечат связь с таблицей счетов (таблица 6).

Таблица 6

Данные об услугах

Имя таблицы: **EXSVC**

Номер счета	Дата счета	Наименование услуги	Стоимость услуги, руб.	Налоговая ставка, %	Налог, руб.	Стоимость услуг, руб.
DOC_ND	DOC_DD	SERVICE_N	COST	TAX	SUM_TAX	SUM_SVC
number(6)	date	varchar2(60)	number(12, 2)	number(5, 2)	number(12, 2)	number(12, 2)
	
	
139	12.09.2003	Услуги доставки	4.70	18.00	0.85	5.55
139	12.09.2003	Услуги упаковки	17.90	18.00	3.22	21.12
	
152	17.09.2003	Услуги доставки	35.00	18.00	6.30	41.30
	
	

Таблицы для учебного примера готовы.

ЗАКЛЮЧЕНИЕ

В данном практикуме рассмотрен подход к работе со средствами реализации информационных технологий с тем, чтобы использовать технологии разработки объектов профессиональной деятельности в различных областях производственной деятельности, а также на предприятиях различного профиля и видов деятельности.

Авторы надеются, что степень подробности изложения материала является достаточной для самостоятельного решения студентами не только представленных в практикуме заданий, но и задач, поставленных перед ними в процессе их самостоятельной профессиональной деятельности.

КОНТРОЛЬНЫЕ ВОПРОСЫ

1. Дайте определение понятия «база данных».
2. Дайте определение понятия «банк данных».
3. Каковы предпосылки создания баз и банков данных?
4. Перечислите преимущества и недостатки использования банков данных.
5. Определите соотношение понятий «информация» и «данные».
6. Перечислите и определите назначение основных компонентов банков данных.
7. Определите основные функции и назначение СУБД.
8. Назовите отличительные особенности банка данных.
9. Перечислите основные требования, предъявляемые к БД.
10. Какие технические средства для создания БД?
11. Перечислите основные признаки классификации БД.
12. Определите основные категории пользователей БД.
13. Перечислите основные функции пользователей.
14. Приведите классификационную схему моделей БД.
15. Приведите сравнительный анализ процессов обработки данных средствами файловой системы и СУБД.
16. Приведите схему управления данными в СУБД.
17. Дайте определение понятия «предметная область».
18. Что является результатом абстрагированного описания предметной области?
19. Дайте сравнительную характеристику понятий «модель данных» и «модель базы данных».
20. Приведите типологию простых запросов.
21. Перечислите факторы, влияющие на выбор метода размещения данных (организацию файла).
22. Перечислите основные этапы создания БД.
23. В чем заключается даталогическое проектирование для конкретной СУБД?
24. Приведите примеры дополнительных функций, которые могут быть реализованы с помощью таблиц БД «Сессия».
25. Какие изменения должны быть внесены в структуру БД «Сессия» для реализации функции «стипендия»?
26. Перечислите основные серверные программные средства восстановления БД в составе СУБД.

ТЕСТОВЫЕ ВОПРОСЫ ДЛЯ КОНТРОЛЯ СФОРМИРОВАННОСТИ КОМПЕТЕНЦИЙ РАБОТЫ С БАЗАМИ ДАННЫХ

Ключ с ответами на тест

1	2	3	4	5	6	7	8	9	10	11	12
В	А	А	С	В	В	В	В	1-А, 2-С, 3-В	А	А	В
А	В	В	В	С	С	А	С	А-2, В-1, С-4	С	А	А
Д	С	С	А	А	В	А	С	1-Д, 2-С, 3-В	С	С	А
В	А	С	А	А	В	А	С	А-3, В-4, С-2	С	А	С

Тема 1.

1. Базы данных (БД) – это:

- совокупность электронных таблиц и всего комплекса аппаратно-программных средств для их хранения; изменения и поиска информации; для взаимодействия с пользователем;
- организованная совокупность данных, предназначенная для длительного хранения во внешней памяти компьютера и постоянного применения;
- программное обеспечение, управляющее хранением и обработкой данных;
- настраиваемые диалоговые окна, сохраняемые в компьютере в виде объектов специального типа.

2. Системы управления базами данных – это:

- инструмент для печати данных, содержащихся в таблицах и запросах, в красиво оформленном виде;
- настраиваемые диалоговые окна, сохраняемые в компьютере в виде объектов специального типа;

- совокупность баз данных и всего комплекса аппаратно-программных средств для их хранения; изменения и поиска информации; для взаимодействия с пользователем;
- программа, позволяющая создавать базы данных, а также обеспечивающая обработку (сортировку) и поиск данных

3. Информационные системы (ИС) – это:

- совокупность баз данных и всего комплекса аппаратно-программных средств для их хранения; изменения и поиска информации; для взаимодействия с пользователем;
- упорядоченные наборы данных;
- программное обеспечение, предназначенное для работы с базами данных;
- важнейший инструмент для отбора данных на основании заданных условий.

4. Системы управления базами данных – это:

- важнейший инструмент для отбора данных на основании заданных условий;
- программа, позволяющая создавать базы данных, а также обеспечивающая обработку (сортировку) и поиск данных;
- настраиваемые диалоговые окна, сохраняемые в компьютере в виде объектов специального типа;
- совокупность баз данных и всего комплекса аппаратно-программных средств для их хранения; изменения и поиска информации; для взаимодействия с пользователем.

Тема 2.

5. По характеру хранимой информации БД бывают:

- Фактографические
- Централизованные
- Иерархические

6. По способу хранения данных БД бывают:

- Фактографические
- Распределенные
- Табличные

7. По структуре организации данных БД бывают:

- Централизованные
- Документальные
- Сетевые

8. По характеру хранимой информации БД бывают:

- Документальные
- Распределенные
- Иерархические

Тема 3.

9. Укажите системы управления БД:

- Microsoft Access
- Open Office.org Calc
- Microsoft Power Point

10. Укажите системы управления БД:

- Microsoft Excel
- Open Office.org Base
- Open Office.org Writer

11. Укажите системы управления БД:

- Open Office.org Calc
- Microsoft Word
- Microsoft Access

12. Укажите системы управления БД:

- Microsoft Excel
- Open Office.org Impress
- Open Office.org Base

Тема 4.

13. Поле БД – это:

- Строка таблицы, содержащая набор значений свойств, в столбцах БД
- Заголовок таблицы БД
- Столбец таблицы, содержащий значения определенного свойства

14. Запись БД – это

- Столбец таблицы, содержащий значения определенного свойства
- Строка таблицы, содержащая набор значений свойств в полях БД
- Заголовок таблицы БД

15. В табличных БД полем называются

- Однородные данные обо всех объектах
- Наборы данных об одном объекте
- Заголовки таблицы БД

16. В табличных БД запись содержит

- Набор данных об одном объекте
- Название базы данных
- Однородные данные обо всех объектах

Тема 5.

17. Перечислите недостатки табличных БД:

- Возможность видеть одновременно несколько записей
- Содержит большое количество полей
- Легко просматривать и редактировать данные

18. Перечислите достоинства БД-форма:

- Возможность видеть одновременно несколько записей
- Содержит большое количество полей
- Легко просматривать и редактировать данные

19. Перечислите недостатки БД-форма:

- Возможность видеть только одну запись
- Содержит большое количество полей
- Легко просматривать и редактировать данные

20. Перечислите достоинства табличных БД:

- Возможность видеть одновременно несколько записей
- Содержит большое количество полей
- Сложно просматривать и редактировать данные

Тема 6.

21. Кто определяет количество полей в БД?

- Пользователь
- Разработчик
- И разработчик, и пользователь

22. Поля каких типов не может содержать БД?

- картинка
- счетчик
- ярлык

23. Какое свойство не является свойством поля БД?

- Размер поля
- Цвет поля
- Обязательное поле

24. Какое свойство не является свойством поля БД?

- Формат поля
- Цвет поля
- Обязательное поле

Тема 7.

25. Какие данные не могут быть ключом БД?

- Номер паспорта
- Дата рождения
- Логин эл. почты + пароль

26. Какие данные не могут быть ключом БД?

- Цвет глаз
- ИНН+СНИЛС
- Логин эл. почты + пароль

27. Какие данные могут быть ключом БД?

- Номер паспорта
- Номер дома
- Цвет волос

28. Какие данные могут быть ключом БД?
- ИНН+СНИЛС
 - Город проживания
 - Имя

Тема 8.

29. Чем запрос отличается от фильтра?
- Ничем
 - Запрос является самостоятельным объектом БД
 - Запрос может быть простым и сложным

30. Чем фильтр отличается от запроса?
- Ничем
 - Фильтр может быть простым и сложным
 - Фильтр привязан к конкретной таблице

31. Что называют сортировкой данных в БД?
- Отбор записей, удовлетворяющих условиям поиска
 - Вывод на печать упорядоченных записей
 - Упорядочение записей по значениям одного из полей

32. Для чего предназначены отчеты в БД?
- Для упорядочения записей в определенной последовательности
 - Для отбора записей, удовлетворяющим определенным условиям
 - Для печати данных, содержащихся в таблицах и запросах, в красиво оформленном виде

Тема 9.

33. Установите соответствие:

Отличительные особенности типов БД	Тип БД
А. Набор узлов, в котором каждый может быть связан с каждым	1. Табличные
В. Данные в виде одной таблицы	2. Сетевые
С. Набор взаимосвязанных таблиц	3. Иерархические
	4. Реляционные

34. Установите соответствие:

Тип ИС	Отличительные особенности типов ИС
1. Локальные	А. БД и СУБД находятся на одном компьютере
2. Файл-серверные	В. БД и основная СУБД находятся на сервере, СУБД на рабочей станции посылает запрос и выводит на экран результат
3. Клиент-серверные	С. БД находится на сервере сети, а СУБД – на компьютере пользователя
	Д. СУБД находится на сервере, а БД – на компьютере пользователя

35. Установите соответствие:

Тип ИС	Отличительные особенности типов ИС
4. Локальные	Е. СУБД находится на сервере, а БД – на компьютере пользователя
5. Файл-серверные	Ф. БД и основная СУБД находятся на сервере, СУБД на рабочей станции посылает запрос и выводит на экран результат
6. Клиент-серверные	Г. БД находится на сервере сети, а СУБД – на компьютере пользователя
	Н. БД и СУБД находятся на одном компьютере

36. Установите соответствие:

Отличительные особенности типов БД	Тип БД
Д. В виде многоуровневой структуры	5. Табличные
Е. Набор взаимосвязанных таблиц	6. Сетевые
Ф. Набор узлов, в котором каждый может быть связан с каждым	7. Иерархические
	8. Реляционные

Тема 10.

37. База данных – это:

- специальным образом организованная и хранящаяся на внешнем носителе совокупность взаимосвязанных данных о некотором объекте
- произвольный набор информации
- совокупность программ для хранения и обработки больших массивов информации

38. В записи файла реляционной БД может содержаться:

- исключительно однородная информация (данные только одного типа)
- только текстовая информация
- неоднородная информация (данные разных типов)

39. Какой из вариантов не является функцией СУБД?

- реализация языков определения и манипулирования данными
- обеспечение пользователя языковыми средствами манипулирования данными
- координация проектирования, реализации и ведения БД

40. Система управления базами данных представляет собой программный продукт, входящий в состав:

- прикладного программного обеспечения
- системного программного обеспечения
- систем программирования

Тема 11.

41. Какая наименьшая единица хранения данных в БД?

- хранимое поле
- хранимый файл
- ничего из вышеперечисленного

42. Что обязательно должно входить в СУБД?

- процессор языка запросов
- визуальная оболочка
- система помощи

43. Структура файла реляционной БД меняется:

- при изменении любой записи
- при уничтожении всех записей
- при удалении любого поля

44. Как называется набор хранимых записей одного типа?

- хранимый файл
- представление базы данных
- ничего из вышеперечисленного

Тема 12.

45. Язык запросов в СУБД называется:

- HTML
- SQL
- Jawa

46. Если данные некоторой БД находятся на нескольких компьютерах, объединенных в компьютерную сеть, то такая БД называется:

- Сетевая
- Централизованная
- Распределенная

47. Главный ключ БД – это:

- Поле или совокупность полей, однозначно определяющих запись
- Имя первого поля БД
- Сортировка данных по возрастанию

48. Наиболее распространенными в практике являются базы данных следующего типа:

- Распределенные
- Иерархические
- Реляционные

ГЛОССАРИЙ

Адресация – способ размещения и выборки данных в памяти

Атрибут – свойство, описывающее определенный аспект объекта, значение которого следует зафиксировать в описании *предметной области*

База данных – поименованная совокупность взаимосвязанных данных, отображающая состояние объектов и их отношений в некоторой предметной области, используемых несколькими пользователями и хранящимися с минимальной избыточностью, управляемых специальной системой, называемой *системой управления базами данных*

Банк данных – система, состоящая из данных программных, технических, языковых, организационно-методических средств, предназначенных для обеспечения централизованного накопления и коллективного многоцелевого использования

Блокировка – запрет выполнения некоторых операций над данными (например, обновления или добавления)

Браузер – специальная программа для просмотра гипертекстовых документов

Дескриптор – описатель документа или его фрагмента

Индекс – системная структура или специальный файл, создаваемые для повышения скорости выполнения запросов; индекс формируется из значений одного или нескольких столбцов таблицы и указателей на соответствующие строки таблицы и позволяет быстро находить нужную строку таблицы по заданному значению

Ключ внешний – атрибут (столбец), или набор атрибутов, подчиненной таблицы, который для главной таблицы является первичным ключом

Ключ возможный – атрибут (столбец), или набор атрибутов, который может быть использован для данного отношения (таблицы) в качестве первичного ключа

Ключ первичный – атрибут отношения (столбец таблицы), или набор из минимального числа атрибутов, который однозначно идентифицирует конкретную строку в таблице

Компиляция – преобразование программы из описания на входном языке (языке программирования) в ее представление на выходном языке (в машинных командах)

Метаданные – информация, описывающая данные, хранящиеся в базе данных

Метаинформация – информация, описывающая базу данных, а также другие части банка данных

Предметная область – часть реального мира, представляющая интерес для данного исследования или использования

Приложение – прикладная компьютерная программа, предназначенная для решения определенной практической задачи

Система базы данных – совокупность программного обеспечения, данных и аппаратного обеспечения компьютеров, которая реализует набор приложений и моделей данных и использует СУБД и прикладное программное обеспечение для создания конкретных информационных систем

Система управления базами данных – совокупность специальных языковых и программных средств, предназначенных для ведения и совместного использования базы данных многими пользователями

Структура данных – атрибутивная форма представления свойств и связей предметной области, ориентированная на выражение описания данных средствами формальных языков и, таким образом, учитывающая возможности и ограничения конкретных средств, с целью сведения описаний к стандартным типам и регулярным связям. Структура данных с точки зрения программирования – это способ отображения значений в памяти (размер области и порядок ее выделения, который и определит характер процедуры адресации – выборки).

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. Голицина, О. Л. Базы данных : учеб. пособие / О. Л. Голицина, Н. В. Максимов, И. И. Попов. – 2-е изд., испр. и доп. – М. : ФОРУМ: ИНФРА-М, 2009. – 400 с.
2. Голицина, О. Л. Основы алгоритмизации и программирования: учеб. пособие / О. Л. Голицина, И. И. Попов. – 3-е изд., испр. и доп. – М.: ФОРУМ, 2008. – 432 с.
3. Гофф, Дж. Р., SOL: полное руководство /Дж. Р. Гофф, П. Н. Вайнберг. – 2-е изд. – Киев. : ВНУ, 2001. – 816 с.
4. Илюшечкин, В. М. Основы использования и проектирования баз данных : учеб. пособие / В. М. Илюшечкин. – М. : Издательство Юрайт ; ИД Юрайт, 2011. – 213 с.
5. Кренке, Д. Теория и практика построения баз данных / Д. Кренке. – 8-е изд. – СПб. : Питер, 2003. – 800 с.
6. Петров В. Н. Информационные системы / В. Н. Петров. – СПб. : Питер, 2003. – 688 с.
7. Пирогов, В. Ю. Информационные системы и базы данных: организация и проектирование : учеб. пособие / В.Ю. Пирогов. – СПб. : БХВ-Петербург, 2009. – 528 с.
8. Роб, П. Системы баз данных: проектирование, реализация, управление / П. Роб, К. Коронел. – 5-е изд., перераб. и доп.: Пер. с англ. – СПб. : БХВ-Петербург, 2004. – 1040 с.
9. Советов Б. Я., Представление знаний в информационных системах: учебник для студ. высш. учеб. заведений / Б. Я. Советов, В. В. Цехановский, В. Д. Чертовской. – М. : Издательский центр «Академия», 2011. – 114 с.
10. ER/Studio User Guide/San Francico, CA : Embarcadero Technologies. Inc., 2006. – 660 pp.

Учебное издание

ЛЕБЕДЕВ Анатолий Валерьевич
ЧОРАКАЕВ Олег Эдуардович

УПРАВЛЕНИЕ ДАННЫМИ

Практикум по выполнению
практических и лабораторных работ

Редактор Н.А. Евдокимова

ЭИ № 1144. Объем данных 3,5 Мб

ЛР № 020640 от 22.10.97

Подписано в печать 17.10.2018.
Формат 60×84/16. Усл. печ. л. 5,81.
Тираж 75 экз. Заказ № 776.

Ульяновский государственный технический университет
432027, г. Ульяновск, ул. Сев. Венец, д. 32.
ИПК «Венец» УлГТУ, 432027, г. Ульяновск, ул. Сев. Венец, д. 32.

Тел.: (8422) 778-113
E-mail: venec@ulstu.ru
venec.ulstu.ru