

ФЕДЕРАЛЬНОЕ АГЕНТСТВО ОБРАЗОВАНИЯ
ДИМИТРОВГРАДСКИЙ ИНСТИТУТ ТЕХНОЛОГИЙ,
УПРАВЛЕНИЯ И ДИЗАЙНА

Лабораторная работа №1
По курсу: "Базы данных"
Вариант №5

Выполнил студент гр.ВТ-31:
Потеренко А.Г.
Проверил преподаватель:

Содержание

Общие сведения для программиста БД.....	3
Задание.....	4
Проектирование БД.....	4
Реализация алгоритма на PL/SQL.....	4
Реализация алгоритма на DELPHI.....	10
Тестирование.....	14

Общие сведения для программиста БД

В качестве СУБД будем использовать ORACLE 9.0.1.0.0 по Windows NT



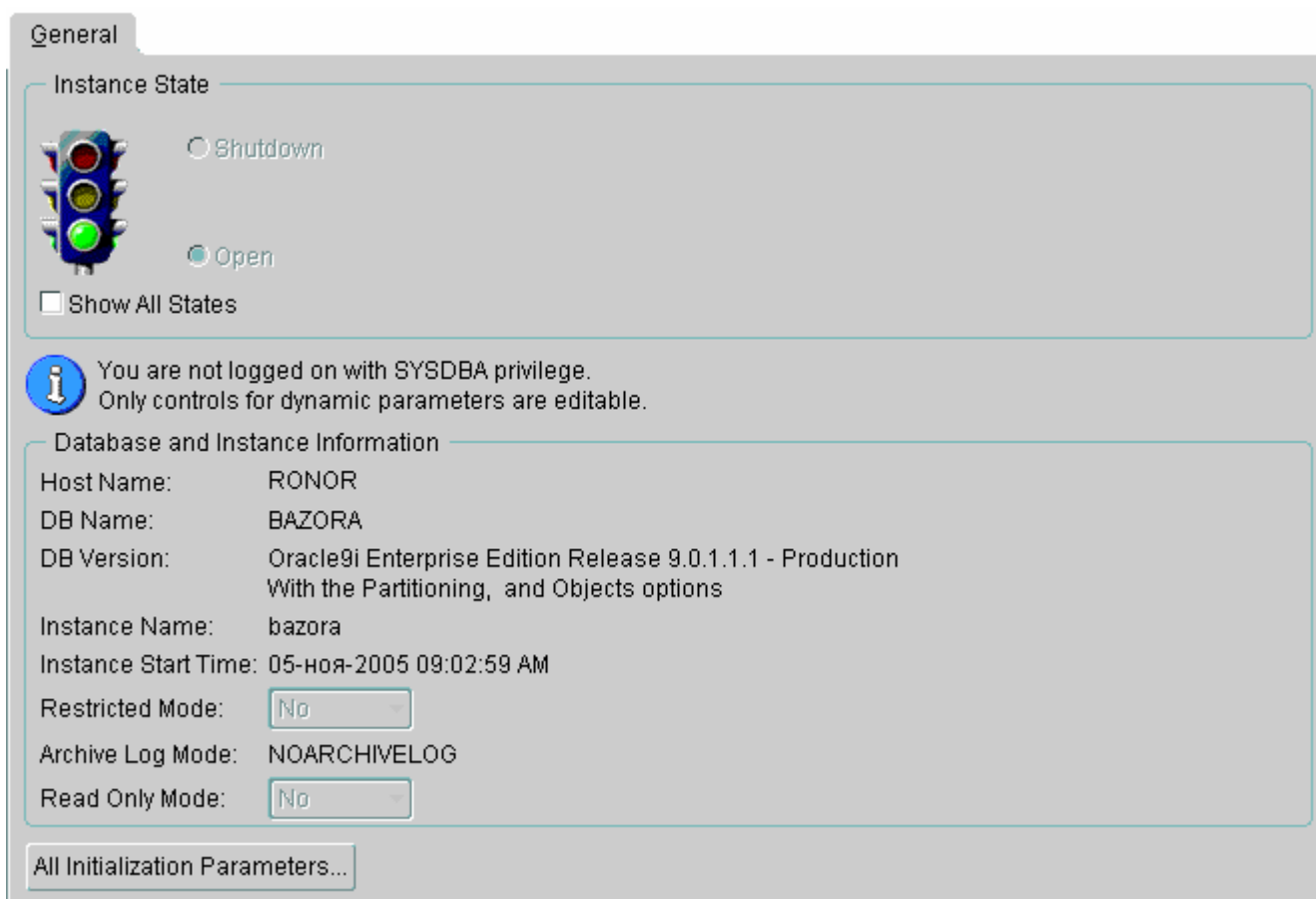
Администратор выдал нам сведения для входа в БД:

Login: system

Password: manager

Конечно можно было создать пользователя и дать ему права, но это тема выходит за рамки данной ЛР.

Продemonстрируем общую характеристику СУБД:



Здесь показано, на каком хосте находится экземпляр БД, его алиас и т.п.

Задание

Необходимо создать БД "Студенты гр. ТИЛ-211 НИТУД (Нижегородский институт)". Поля: ФИО, дата рождения, место рождения, оценка общих знаний, диплом. Также необходимо сортированные данные по убыванию по колонке "знания".

Необходимо предоставить пользователю отчет о данных студентах.

Писать программу будем на DELPHI7 через ADO.

Проектирование БД

Таблица №1. Уникальный идентификатор + ФИО+ДАТА+МЕСТО+ОЦЕНКА

TAB1_id	TAB1_FIO	TAB1_DATA	TAB1_MESTO	TAB1_OCENKA
integer NOT NULL (primary key)	char(30)	date	char(20)	integer

В данной таблице у одного человека все поля уникальные (Не может иметь две даты рождения), а дипломов может быть несколько (возможно мы проводим статистику по человеку в группе, а он уже закончил другой институт). Место рождения будем подразумевать город, в котором человек родился.

Таблица №2. Уникальный идентификатор + № Диплома.

TAB2_id	TAB2_keyTAB1	TAB2_DIPLOM
integer NOT NULL (primary key)	integer NOT NULL (foreign key)	integer (unique)

Здесь одно поле занимает № Диплома.

Реализация алгоритма на PL/SQL

Создание БД

```
CREATE TABLE TAB1
(
  TAB1_ID INTEGER NOT NULL PRIMARY KEY,
  TAB1_FIO CHAR(30),
  TAB1_DATA DATE,
  TAB1_MESTO CHAR(20),
  TAB1_OCENKA INTEGER CHECK ((TAB1_OCENKA>0) AND (TAB1_OCENKA<6))
);
CREATE TABLE TAB2
(
  TAB2_ID INTEGER NOT NULL PRIMARY KEY,
  TAB2_KEYTAB1 INTEGER NOT NULL,
  FOREIGN KEY (TAB2_KEYTAB1) REFERENCES TAB1 (TAB1_ID),
  TAB2_DIPLOM INTEGER UNIQUE
);
```

Удаление БД

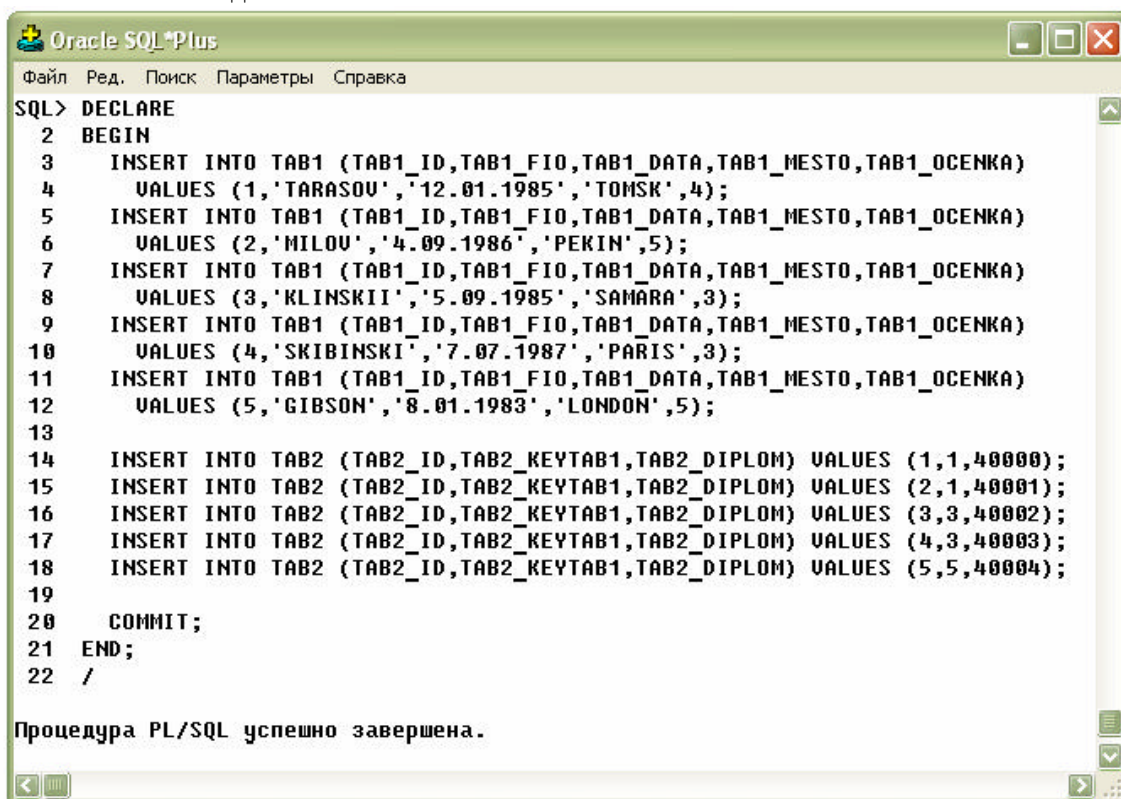
Сначала удаляем подчиненную таблицу, затем главную таблицу.

```
DROP TABLE TAB2;  
DROP TABLE TAB1;
```

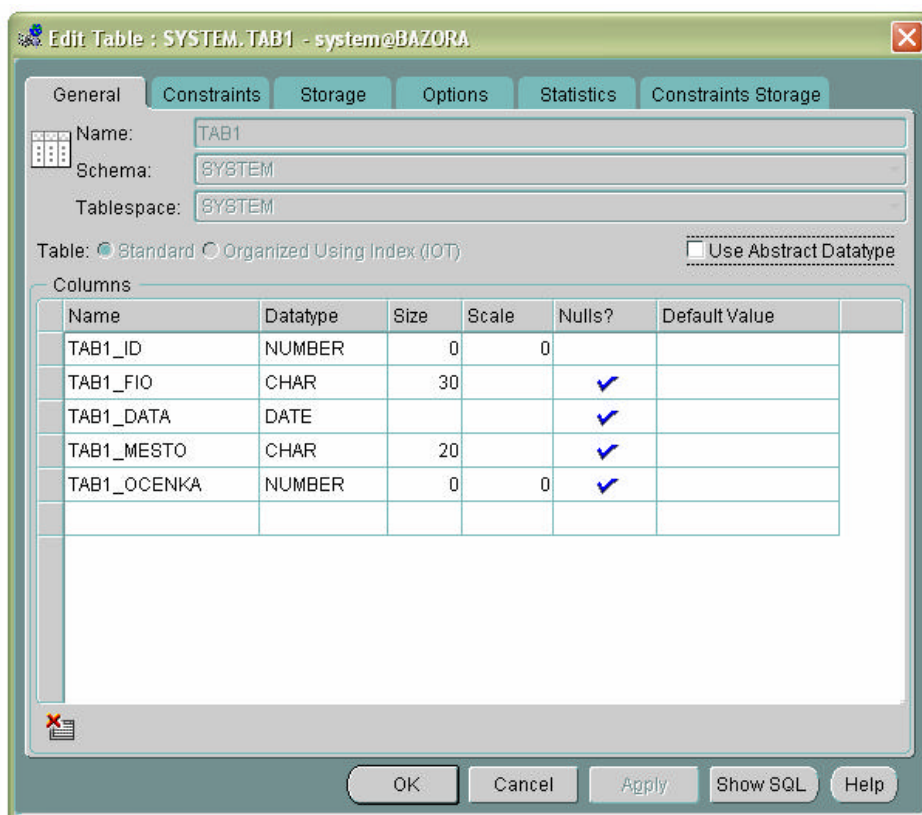
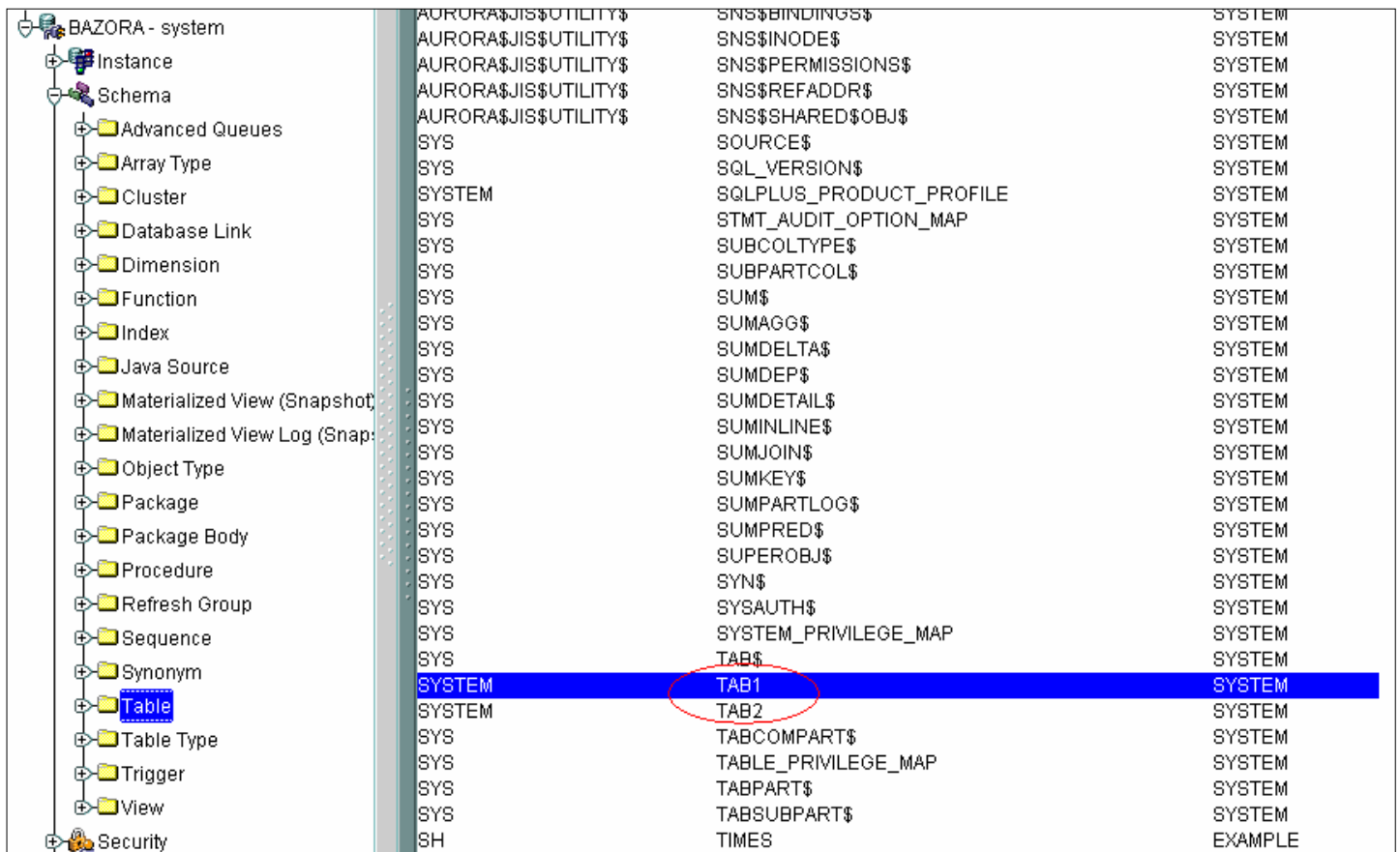
Вставка записей БД

```
DECLARE  
BEGIN  
    INSERT INTO TAB1 (TAB1_ID,TAB1_FIO,TAB1_DATA,TAB1_MESTO,TAB1_OCENKA)  
        VALUES (1,'TARASOV','12.01.1985','TOMSK',4);  
    INSERT INTO TAB1 (TAB1_ID,TAB1_FIO,TAB1_DATA,TAB1_MESTO,TAB1_OCENKA)  
        VALUES (2,'MILOV','4.09.1986','PEKIN',5);  
    INSERT INTO TAB1 (TAB1_ID,TAB1_FIO,TAB1_DATA,TAB1_MESTO,TAB1_OCENKA)  
        VALUES (3,'KLINSKII','5.09.1985','SAMARA',3);  
    INSERT INTO TAB1 (TAB1_ID,TAB1_FIO,TAB1_DATA,TAB1_MESTO,TAB1_OCENKA)  
        VALUES (4,'SKIBINSKI','7.07.1987','PARIS',3);  
    INSERT INTO TAB1 (TAB1_ID,TAB1_FIO,TAB1_DATA,TAB1_MESTO,TAB1_OCENKA)  
        VALUES (5,'GIBSON','8.01.1983','LONDON',5);  
  
    INSERT INTO TAB2 (TAB2_ID,TAB2_KEYTAB1,TAB2_DIPLOM) VALUES (1,1,40000);  
    INSERT INTO TAB2 (TAB2_ID,TAB2_KEYTAB1,TAB2_DIPLOM) VALUES (2,1,40001);  
    INSERT INTO TAB2 (TAB2_ID,TAB2_KEYTAB1,TAB2_DIPLOM) VALUES (3,3,40002);  
    INSERT INTO TAB2 (TAB2_ID,TAB2_KEYTAB1,TAB2_DIPLOM) VALUES (4,3,40003);  
    INSERT INTO TAB2 (TAB2_ID,TAB2_KEYTAB1,TAB2_DIPLOM) VALUES (5,5,40004);  
  
    COMMIT;  
END;
```

TARASOV ИМЕЕТ 2 ДИПЛОМА – 40000, 40001
KLINSKII ИМЕЕТ 2 ДИПЛОМА – 40002, 40003
GIBSON ИМЕЕТ 1 ДИПЛОМ – 40004



```
SQL> DECLARE  
2 BEGIN  
3     INSERT INTO TAB1 (TAB1_ID,TAB1_FIO,TAB1_DATA,TAB1_MESTO,TAB1_OCENKA)  
4         VALUES (1,'TARASOV','12.01.1985','TOMSK',4);  
5     INSERT INTO TAB1 (TAB1_ID,TAB1_FIO,TAB1_DATA,TAB1_MESTO,TAB1_OCENKA)  
6         VALUES (2,'MILOV','4.09.1986','PEKIN',5);  
7     INSERT INTO TAB1 (TAB1_ID,TAB1_FIO,TAB1_DATA,TAB1_MESTO,TAB1_OCENKA)  
8         VALUES (3,'KLINSKII','5.09.1985','SAMARA',3);  
9     INSERT INTO TAB1 (TAB1_ID,TAB1_FIO,TAB1_DATA,TAB1_MESTO,TAB1_OCENKA)  
10        VALUES (4,'SKIBINSKI','7.07.1987','PARIS',3);  
11    INSERT INTO TAB1 (TAB1_ID,TAB1_FIO,TAB1_DATA,TAB1_MESTO,TAB1_OCENKA)  
12        VALUES (5,'GIBSON','8.01.1983','LONDON',5);  
13  
14    INSERT INTO TAB2 (TAB2_ID,TAB2_KEYTAB1,TAB2_DIPLOM) VALUES (1,1,40000);  
15    INSERT INTO TAB2 (TAB2_ID,TAB2_KEYTAB1,TAB2_DIPLOM) VALUES (2,1,40001);  
16    INSERT INTO TAB2 (TAB2_ID,TAB2_KEYTAB1,TAB2_DIPLOM) VALUES (3,3,40002);  
17    INSERT INTO TAB2 (TAB2_ID,TAB2_KEYTAB1,TAB2_DIPLOM) VALUES (4,3,40003);  
18    INSERT INTO TAB2 (TAB2_ID,TAB2_KEYTAB1,TAB2_DIPLOM) VALUES (5,5,40004);  
19  
20    COMMIT;  
21 END;  
22 /  
  
Процедура PL/SQL успешно завершена.
```



Dialog box: Edit Table : SYSTEM.TAB2 - system@BAZORA

General Constraints Storage Options Statistics Constraints Storage

Name: TAB2
Schema: SYSTEM
Tablespace: SYSTEM

Table: ☒ Standard ☐ Organized Using Index (OTI) ☐ Use Abstract Datatype

Name	Datatype	Size	Scale	Nulls?	Default Value	
TAB2_ID	NUMBER	0	0			
TAB2_KEYTAB1	NUMBER	0	0			
TAB2_DIPLOM	NUMBER	0	0	✓		

OK Cancel Apply Show SQL Help

Выборка значений из БД

```
SELECT TAB1_FIO, TAB1_DATA, TAB1_MESTO, TAB1_OCENKA FROM TAB1
ORDER BY TAB1_OCENKA DESC
```

Oracle SQL*Plus

Файл Ред. Поиск Параметры Справка

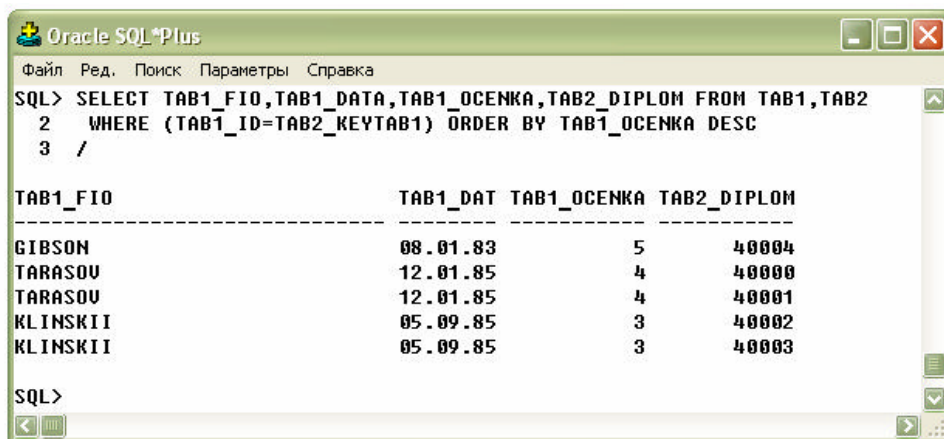
SQL> SELECT TAB1_FIO, TAB1_DATA, TAB1_MESTO, TAB1_OCENKA FROM TAB1
2 ORDER BY TAB1_OCENKA DESC
3 /

TAB1_FIO	TAB1_DATA	TAB1_MESTO	TAB1_OCENKA
MILOV	04.09.86	PEKIN	5
GIBSON	08.01.83	LONDON	5
TARASOV	12.01.85	TOMSK	4
KLINSKII	05.09.85	SAMARA	3
SKIBINSKI	07.07.87	PARIS	3

SQL>

Выводим сортированные по заданию данные о тех учащихся, которые имеют дипломы:

```
SELECT TAB1_FIO, TAB1_DATA, TAB1_OCENKA, TAB2_DIPLOM FROM TAB1, TAB2
WHERE (TAB1_ID=TAB2_KEYTAB1) ORDER BY TAB1_OCENKA DESC
```

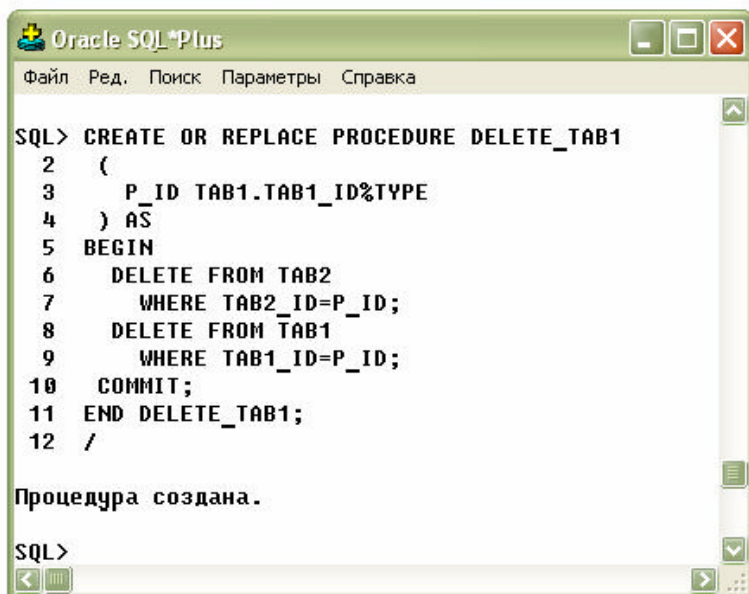


Oracle SQL*Plus window showing a query result. The query is: `SELECT TAB1_FIO, TAB1_DATA, TAB1_OCENKA, TAB2_DIPLOM FROM TAB1, TAB2 WHERE (TAB1_ID=TAB2_KEYTAB1) ORDER BY TAB1_OCENKA DESC`. The result is a table with 4 columns: TAB1_FIO, TAB1_DATA, TAB1_OCENKA, and TAB2_DIPLOM. The data is as follows:

TAB1_FIO	TAB1_DATA	TAB1_OCENKA	TAB2_DIPLOM
GIBSON	08.01.83	5	40004
TARASOV	12.01.85	4	40000
TARASOV	12.01.85	4	40001
KLINSKII	05.09.85	3	40002
KLINSKII	05.09.85	3	40003

Для удаления записи в таблице TAB1 можно воспользоваться хранимой процедурой:

```
CREATE OR REPLACE PROCEDURE DELETE_TAB1
(
  P_ID TAB1.TAB1_ID%TYPE
) AS
BEGIN
  DELETE FROM TAB2
  WHERE TAB2_ID=P_ID;
  DELETE FROM TAB1
  WHERE TAB1_ID=P_ID;
  COMMIT;
END DELETE_TAB1;
```

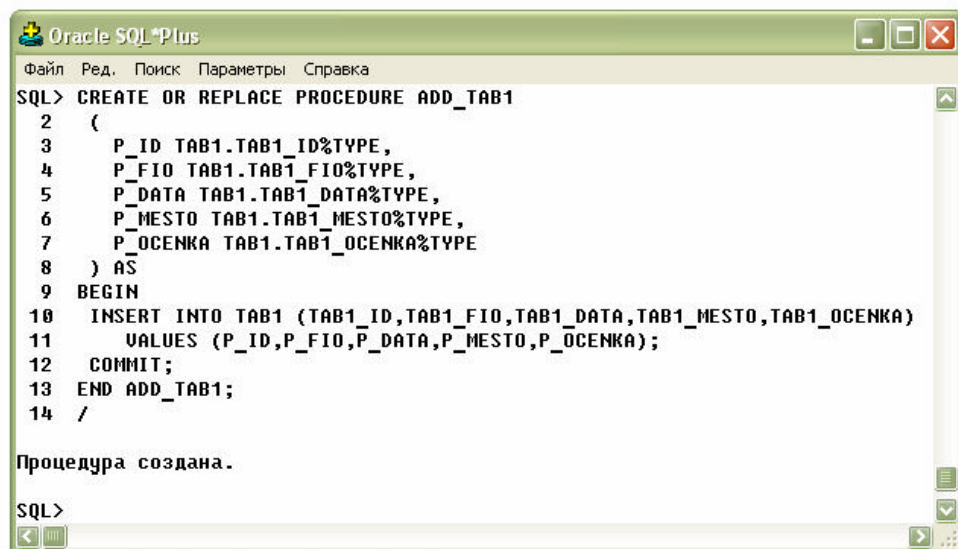


Oracle SQL*Plus window showing the creation of a stored procedure. The query is: `CREATE OR REPLACE PROCEDURE DELETE_TAB1 (P_ID TAB1.TAB1_ID%TYPE) AS BEGIN DELETE FROM TAB2 WHERE TAB2_ID=P_ID; DELETE FROM TAB1 WHERE TAB1_ID=P_ID; COMMIT; END DELETE_TAB1;`. The result is: `Процедура создана.`

Для выполнения данного запроса необходимо выполнить на сервере:
EXECUTE DELETE_TAB1 (порядковый номер человека);

Для добавления записи в таблице TAB1 можно воспользоваться хранимой процедурой:

```
CREATE OR REPLACE PROCEDURE ADD_TAB1
(
  P_ID TAB1.TAB1_ID%TYPE,
  P_FIO TAB1.TAB1_FIO%TYPE,
  P_DATA TAB1.TAB1_DATA%TYPE,
  P_MESTO TAB1.TAB1_MESTO%TYPE,
  P_OCENKA TAB1.TAB1_OCENKA%TYPE
) AS
BEGIN
  INSERT INTO TAB1 (TAB1_ID, TAB1_FIO, TAB1_DATA, TAB1_MESTO, TAB1_OCENKA)
  VALUES (P_ID, P_FIO, P_DATA, P_MESTO, P_OCENKA);
  COMMIT;
END ADD_TAB1;
```



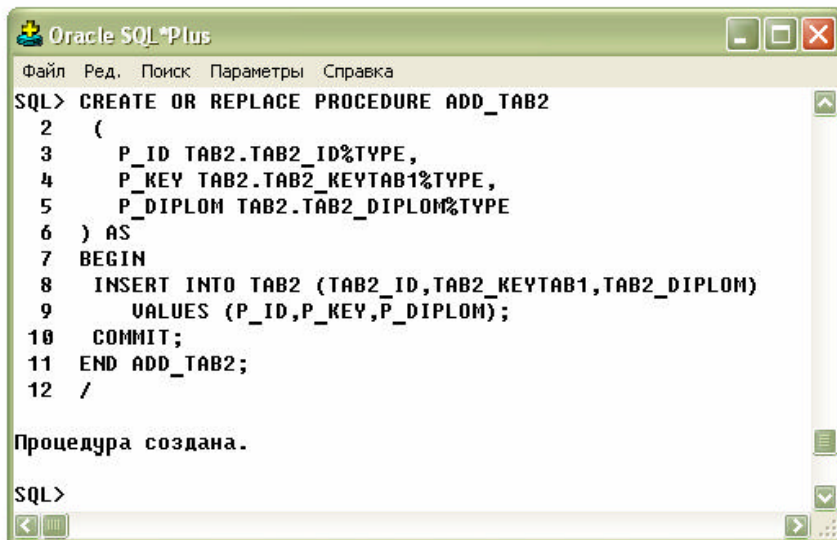
Для выполнения данного запроса необходимо выполнить на сервере:

EXECUTE ADD_TAB1 (необходимые параметры);

Так как человек, получивший диплом никуда его не денет, поэтому процедуру удаления из TAB2 делать не нужно.

Для добавления записи в таблице TAB2 можно воспользоваться хранимой процедурой:

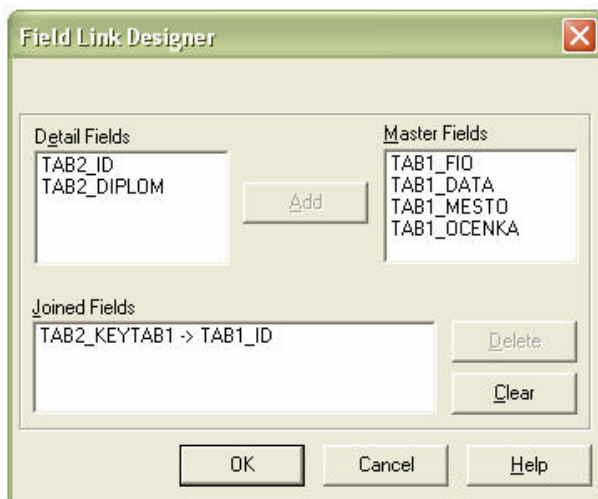
```
CREATE OR REPLACE PROCEDURE ADD_TAB2
(
  P_ID TAB2.TAB2_ID%TYPE,
  P_KEY TAB2.TAB2_KEYTAB1%TYPE,
  P_DIPLOM TAB2.TAB2_DIPLOM%TYPE
) AS
BEGIN
  INSERT INTO TAB2 (TAB2_ID, TAB2_KEYTAB1, TAB2_DIPLOM)
  VALUES (P_ID, P_KEY, P_DIPLOM);
  COMMIT;
END ADD_TAB2;
```



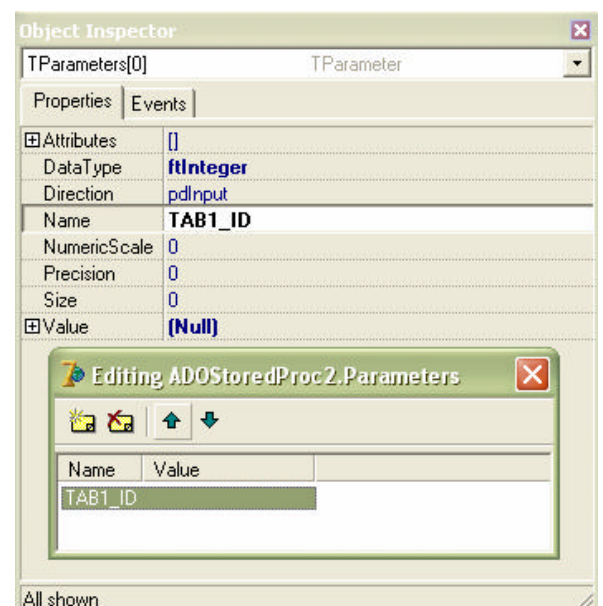
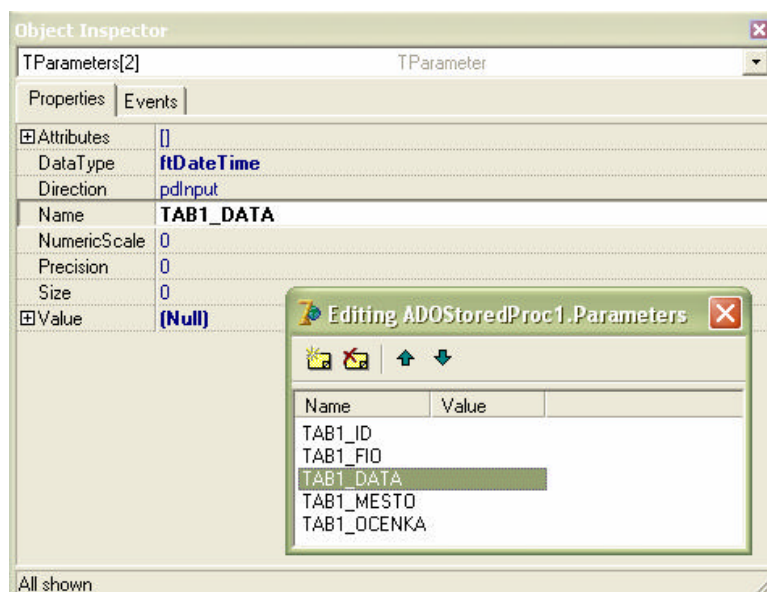
Реализация алгоритма на DELPHI

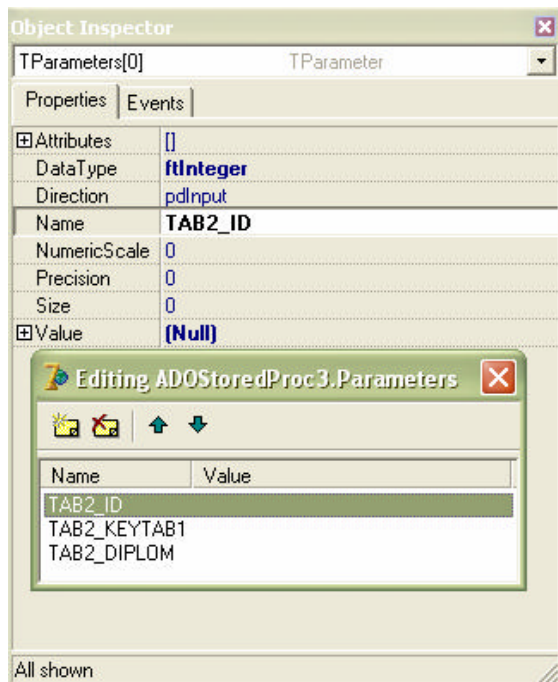
Структуру мы создали, теперь необходимо написать клиента для БД с помощью технологии ADO. Ее описание выходит за рамки данной ЛР.

Связываем таблицы:



Для ADO – компонентов, выполняющих хранимые процедуры:





Листинг программы

```

unit Unit1;

interface

uses
  Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
  Dialogs, DB, Grids, DBGrids, ADODB, StdCtrls, ExtCtrls, RpDefine, RpCon,
  RpConDS, RpRave;

type
  TForm1 = class(TForm)
    ADOConnection1: TADOConnection;
    ADOTable1: TADOTable;
    DataSource1: TDataSource;
    GroupBox1: TGroupBox;
    DBGrid1: TDBGrid;
    Button1: TButton;
    Label1: TLabel;
    Label2: TLabel;
    DBGrid2: TDBGrid;
    ADOTable2: TADOTable;
    DataSource2: TDataSource;
    ADOQuery1: TADOQuery;
    DataSource3: TDataSource;
    Button2: TButton;
    Button3: TButton;
    Button4: TButton;
    Button5: TButton;
    Label3: TLabel;
    Edit1: TEdit;
    Label4: TLabel;
    Label5: TLabel;
    Label6: TLabel;
    Edit2: TEdit;
    Edit3: TEdit;
    Edit4: TEdit;
    Edit5: TEdit;
    Label7: TLabel;
    Button6: TButton;
    Label8: TLabel;
  end;

```

```

Edit6: TEdit;
Button7: TButton;
Label9: TLabel;
Edit7: TEdit;
Label10: TLabel;
Edit8: TEdit;
ADOStoredProc1: TADOStoredProc;
ADOStoredProc2: TADOStoredProc;
ADOStoredProc3: TADOStoredProc;
Label11: TLabel;
Edit9: TEdit;
RvDataSetConnection1: TRvDataSetConnection;
RvDataSetConnection2: TRvDataSetConnection;
RvProject1: TRvProject;
Button8: TButton;
RvProject2: TRvProject;
Button9: TButton;
RvDataSetConnection3: TRvDataSetConnection;
RvProject3: TRvProject;
Button10: TButton;
procedure Button1Click(Sender: TObject);
procedure Button2Click(Sender: TObject);
procedure Button3Click(Sender: TObject);
procedure Button6Click(Sender: TObject);
procedure Button5Click(Sender: TObject);
procedure Button4Click(Sender: TObject);
procedure Button7Click(Sender: TObject);
procedure Button8Click(Sender: TObject);
procedure Button9Click(Sender: TObject);
procedure Button10Click(Sender: TObject);
private
{ Private declarations }
public
{ Public declarations }
end;

var
  Form1: TForm1;

implementation

{$R *.dfm}
//////////Соединение//////////
procedure TForm1.Button1Click(Sender: TObject);
begin
  try
    Form1.ADOConnection1.LoginPrompt:=false;
    Form1.ADOConnection1.Connected:=true;
    Showmessage('Соединение с БД выполнено!!!');
  except
    Showmessage('Ошибка соединения с БД!!!');
  end;
end;
//////////Запрос по заданию//////////
procedure TForm1.Button2Click(Sender: TObject);
begin
  Label1.Caption:='Запрос по заданию';
  Form1.DBGrid1.DataSource:=Form1.DataSource3;
  Form1.ADOQuery1.SQL.Text:=
    'SELECT TAB1_FIO,TAB1_DATA,TAB1_MESTO,TAB1_OCENKA '+
    'FROM TAB1 ORDER BY TAB1_OCENKA DESC';
  Form1.ADOQuery1.Active:=true;
end;
//////////Главный запрос//////////
procedure TForm1.Button3Click(Sender: TObject);
begin
  Form1.DBGrid1.DataSource:=Form1.DataSource1;
  Form1.ADOTable1.Active:=false;

```

```
Form1.ADOTable2.Active:=false;
Form1.ADOTable1.Active:=true;
Form1.ADOTable2.Active:=true;
end;
//////////Разрыв соединения//////////
procedure TForm1.Button6Click(Sender: TObject);
begin
Form1.ADOConnection1.Connected:=false;
Form1.ADOTable1.Active:=false;
Form1.ADOTable2.Active:=false;
end;
//////////Удаление записи//////////
procedure TForm1.Button5Click(Sender: TObject);
begin
try
Form1.ADOStoredProc2.Parameters.ParamByName('TAB1_ID').Value:=StrToInt(Edit7.Text);
Form1.ADOStoredProc2.ExecProc;
Showmessage('Удачное выполнение хранимой процедуры на сервере!!!');
Form1.ADOTable1.Active:=false;
Form1.ADOTable2.Active:=false;
Form1.ADOTable1.Active:=true;
Form1.ADOTable2.Active:=true;
except
Showmessage('Не удалось выполнить хранимую процедуру на сервере!!!');
end;
end;
//////////Добавить студента//////////
procedure TForm1.Button4Click(Sender: TObject);
begin
try
Form1.ADOStoredProc1.Parameters.ParamByName('TAB1_ID').Value:=StrToInt(Edit8.Text);
Form1.ADOStoredProc1.Parameters.ParamByName('TAB1_FIO').Value:=Edit1.Text;
Form1.ADOStoredProc1.Parameters.ParamByName('TAB1_DATA').Value:=StrToDate(Edit2.Text);
Form1.ADOStoredProc1.Parameters.ParamByName('TAB1_MESTO').Value:=Edit3.Text;
Form1.ADOStoredProc1.Parameters.ParamByName('TAB1_OCENKA').Value:=StrToInt(Edit4.Text);
Form1.ADOStoredProc1.ExecProc;
Showmessage('Удачное выполнение хранимой процедуры на сервере!!!');
Form1.ADOTable1.Active:=false;
Form1.ADOTable2.Active:=false;
Form1.ADOTable1.Active:=true;
Form1.ADOTable2.Active:=true;
except
Showmessage('Не удалось выполнить хранимую процедуру на сервере!!!');
end;
end;
//////////Добавить диплом для студента//////////
procedure TForm1.Button7Click(Sender: TObject);
begin
try
Form1.ADOStoredProc3.Parameters.ParamByName('TAB2_ID').Value:=
STRTOINT(Edit9.Text);
Form1.ADOStoredProc3.Parameters.ParamByName('TAB2_KEYTAB1').Value:=
STRTOINT(Edit5.Text);
Form1.ADOStoredProc3.Parameters.ParamByName('TAB2_DIPLOM').Value:=
STRTOINT(Edit6.Text);
Form1.ADOStoredProc3.ExecProc;
Showmessage('Удачное выполнение хранимой процедуры на сервере!!!');
Form1.ADOTable1.Active:=false;
Form1.ADOTable2.Active:=false;
Form1.ADOTable1.Active:=true;
Form1.ADOTable2.Active:=true;
except
Showmessage('Не удалось выполнить хранимую процедуру на сервере!!!');
end;
end;
//////////Отчет о работе//////////
procedure TForm1.Button8Click(Sender: TObject);
begin
```

```

try
  Form1.RvProject2.Execute; //Главная таблица
except
  Showmessage('Не удалось напечатать отчет!!!');
end;
end;
procedure TForm1.Button9Click(Sender: TObject);
begin
try
  Form1.RvProject1.Execute; //Подчиненная таблица
except
  Showmessage('Не удалось напечатать отчет!!!');
end;
end;
procedure TForm1.Button10Click(Sender: TObject);
begin
try
  Form1.RvProject3.Execute; //Запрос-отчет
except
  Showmessage('Не удалось напечатать отчет!!!');
end;
end;
end.

```

Тестирование

Лабораторная работа №1 по БД, ВТ-31. Потеренко А.Г. 2005

Соединение с БД Разорвать соединение Выполнить запрос по заданию Главный запрос

Главная таблица

TAB1_ID	TAB1_FIO	TAB1_DATA	TAB1_MESTO	TAB1_OCENKA
1	TARASOV	12.01.1985	TOMSK	4
2	MILOV	04.09.1986	PEKIN	5
3	KLINSKII	05.09.1985	SAMARA	3
4	SKIBINSKI	07.07.1987	PARIS	3
5	GIBSON	08.01.1983	LONDON	5

Подчиненная таблица

TAB2_ID	TAB2_KEYTAB1	TAB2_DIPLOM
3	3	40002
4	3	40003

ФИО КОД ДИПЛОМА Подготовить отчет №1

ДАТА РОЖДЕНИЯ КОД ЧЕЛОВЕКА Подготовить отчет №2

МЕСТО РОЖДЕНИЯ № ДИПЛОМА Подготовить отчет №3

ОЦЕНКА ЗНАНИЙ Добавить диплом для студента

КОД ЧЕЛОВЕКА КОД ЧЕЛОВЕКА

Добавить студента Удалить студента

Лабораторная работа №1 по БД, ВТ-31. Потеренко А.Г. 2005

Запрос по заданию

TAB1_FIO	TAB1_DATA	TAB1_MESTO	TAB1_OCENKA
MILOV	04.09.1986	PEKIN	5
GIBSON	08.01.1983	LONDON	5
TARASOV	12.01.1985	TOMSK	4
KLINSKII	05.09.1985	SAMARA	3
SKIBINSKI	07.07.1987	PARIS	3

Подчиненная таблица

TAB2_ID	TAB2_KEYTAB1	TAB2_DIPLOM
3	3	40002
4	3	40003

ФИО КОД ДИПЛОМА
 ДАТА РОЖДЕНИЯ КОД ЧЕЛОВЕКА
 МЕСТО РОЖДЕНИЯ № ДИПЛОМА
 ОЦЕНКА ЗНАНИЙ
 КОД ЧЕЛОВЕКА КОД ЧЕЛОВЕКА

Report Preview

File Page Zoom

Page 1 of 1 Zoom 90.0 %

Главная таблица

Код студента	ФИО	Дата рождения	Место рождения	Оценка
1	TARASOV	12.01.19	TOMSK	4
2	MILOV	04.09.1	PEKIN	5
3	KLINSKII	05.09.19	SAMARA	3
4	SKIBINSKI	07.07.19	PARIS	3
5	GIBSON	08.01.19	LONDON	5

Report Preview

File Page Zoom

Page 1 of 1 Zoom 90.0 %

Подчиненная таблица

Код студента	Номер диплома
1	40000
1	40001
3	40002
3	40003
5	40004

Report Preview

File Page Zoom

Page 1 of 1 Zoom 100.0 %

Запрос (сортировка по полю "Оценка")

ФИО	Дата рождения	Место рождения	Оценка
MILOV	04.09.1986	PEKIN	5
GIBSON	08.01.1983	LONDON	5
TARASOV	12.01.1985	TOMSK	4
KLINSKII	05.09.1985	SAMARA	3
SKIBINSKI	07.07.1987	PARIS	3