МИНИСТЕРСТВО ПО ОБРАЗОВАНИЮ И НАУКЕ РФ

ВОЛОГОДСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ

Кафедра Автоматики и Вычислительной Техники

Базы данных

Пояснительная записка к курсовой работе

«Электронный деканат»

Выполнил: Веселов Д.А.

Группа: ЭПО-21

Проверила: Ржеуцкая С.Ю.

Вологда

2011

Содержание:

[1. Постановка задачи 3](#_Toc312756456)

[2. Структура базы данных 3](#_Toc312756457)

[3. Приложение работника деканата 5](#_Toc312756459)

[3.1. Хранимый код для приложения 5](#_Toc312756460)

[3.2. Скриншоты приложения 7](#_Toc312756461)

[3.3. Часть наиболее интересных запросов из приложения 13](#_Toc312756462)

[4. Приложение для преподавателя 13](#_Toc312756463)

[4.1. Хранимый код для приложении 13](#_Toc312756464)

[4.2. Скриншоты приложения 14](#_Toc312756465)

[4.3. Часть наиболее интересных запросов из приложения 15](#_Toc312756466)

[5. Приложение декана 16](#_Toc312756467)

[5.1. Хранимый код для приложения 16](#_Toc312756468)

[5.2. Скриншоты приложения 18](#_Toc312756469)

[5.3. Часть наиболее интересных запросов из приложения 22](#_Toc312756470)

# Постановка задачи

Система учета и анализа успеваемости студентов (электронный деканат). Базовая функциональность:

* Ввод и корректировка всех данных;
* Печать экзаменационных ведомостей и ввод оценок в базу данных;
* Автоматический перевод студентов на следующий курс, отчисление неуспевающих, оформление академического отпуска и восстановление из академического отпуска;
* Печать вкладышей к диплому для окончивших обучение и перемещение данных о них в архив выпускников;
* Анализ успеваемости студента, группы, курса, специальности или факультета в целом

# Структура базы данных

# 

Рис. 1. Структура таблицы «Предметы»



Рис. 2. Структура таблицы «Названия предметов»

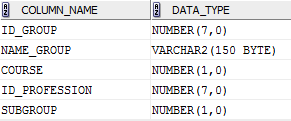


Рис. 3. Структура таблицы «Группы»

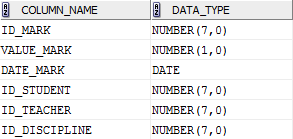


Рис. 4. Структура таблицы «Оценки»

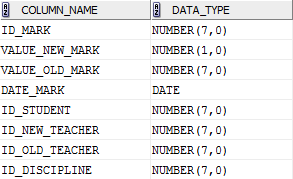


Рис. 5. Структура таблицы «История оценок»



Рис. 6. Структура таблицы «Специальности»



Рис. 7. Структура таблицы «Статусы обучения»

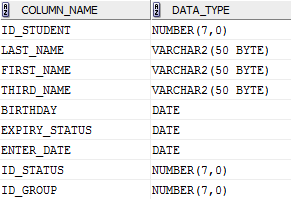


Рис. 8. Структура таблицы «Студенты»

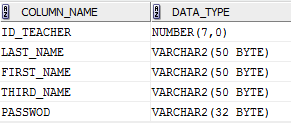


Рис. 9. Структура таблицы «Преподаватели»



Рис. 9. Структура таблицы «Предметы преподавателей»



Рис. 10. Структура таблицы «Типы предметов»

# Приложение работника деканата

Работник деканата может добавлять, изменять и удалять:

* Специальности;
* Группы;
* Студентов;
* Предметы;
* Предметы групп;
* Предметы преподавателей.

# Хранимый код для приложения

В этом (и в двух следующих) приложении используются хранимые функции для:

* Получения названия предмета по его идентификатору

|  |
| --- |
| create or replace  FUNCTION GET\_DISCNAME  (  ids IN NUMBER  ) RETURN VARCHAR2 AS  dname varchar(100);  BEGIN  SELECT discnames.DISC\_NAME into dname FROM discnames WHERE discnames.id\_discname = ids;  RETURN dname;  END GET\_DISCNAME; |

* Получения названия предмета вместе с названием группы по идентификатору предмета и идентификатору группы

|  |
| --- |
| create or replace  FUNCTION GET\_DISCNAME\_WITH\_GROUPNAME  (  ids IN NUMBER,  idg IN varchar2  ) RETURN VARCHAR2 AS  dname varchar(100);  gname varchar(150);  BEGIN  SELECT discnames.DISC\_NAME into dname FROM discnames WHERE discnames.id\_discname = ids;  SELECT get\_groupname(name\_group, course, subgroup) into gname FROM groups WHERE groups.id\_group = idg;  RETURN dname || ' (' || gname || ')';  END GET\_DISCNAME\_WITH\_GROUPNAME; |

* Получения полного ФИО студентов и преподавателей

|  |
| --- |
| create or replace  FUNCTION GET\_FULLNAME  (  LAST\_NAME IN VARCHAR2  , FIRST\_NAME IN VARCHAR2  , THIRD\_NAME IN VARCHAR2  ) RETURN VARCHAR2 AS  BEGIN  RETURN LAST\_NAME || ' ' || FIRST\_NAME || ' ' || THIRD\_NAME;  END GET\_FULLNAME; |

* Получения полного названия группы

|  |
| --- |
| create or replace  FUNCTION GET\_GROUPNAME  (  NAME\_GROUP IN VARCHAR2  , COURSE IN NUMBER  , SUBGROUP IN NUMBER  ) RETURN VARCHAR2 AS  BEGIN  RETURN name\_group || '-' || to\_char(course) || to\_char(subgroup);  END GET\_GROUPNAME; |

# Скриншоты приложения

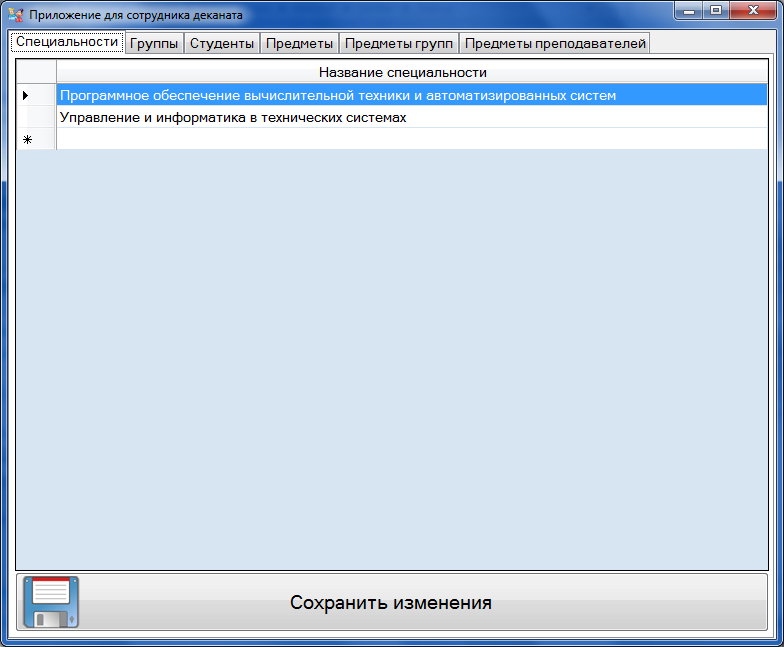


Рис. 11. Вкладка «Специальности на факультете»

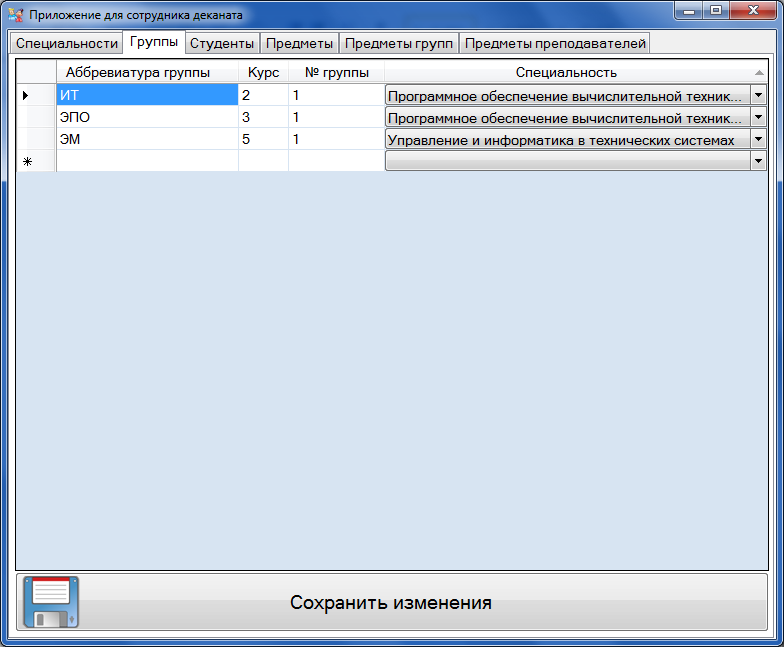


Рис. 12. Вкладка «Группы»

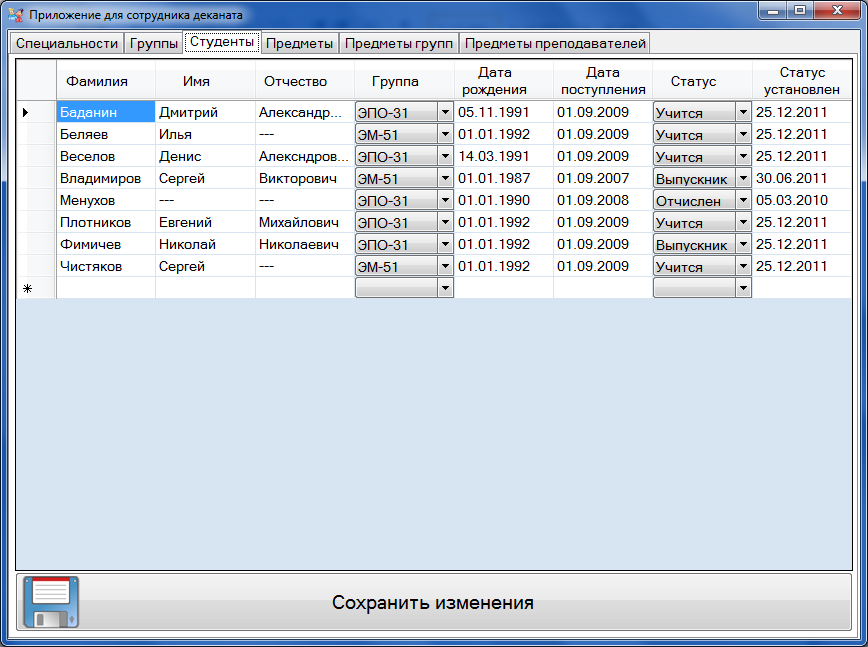


Рис. 13. Вкладка «Студенты»

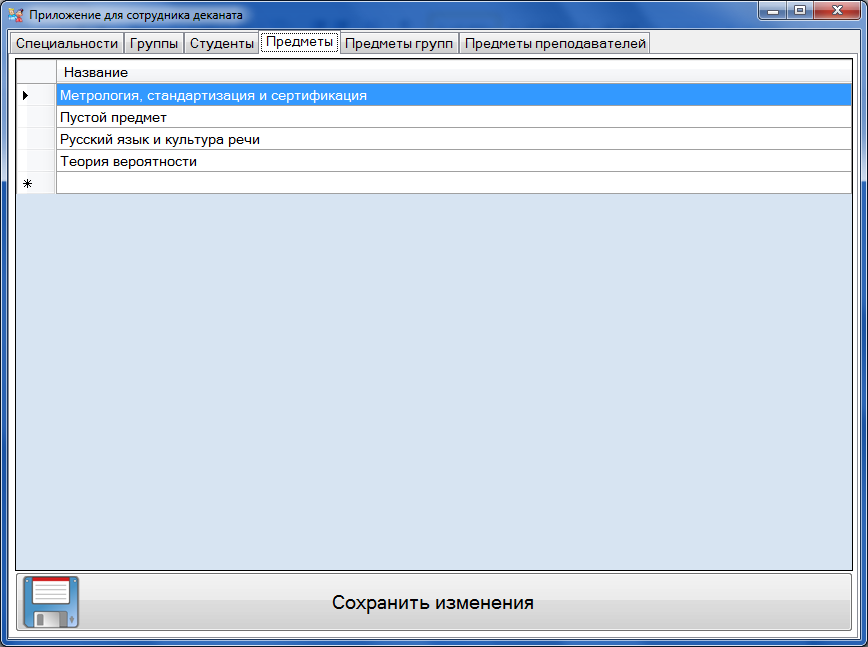


Рис. 14. Вкладка «Названия предметов»

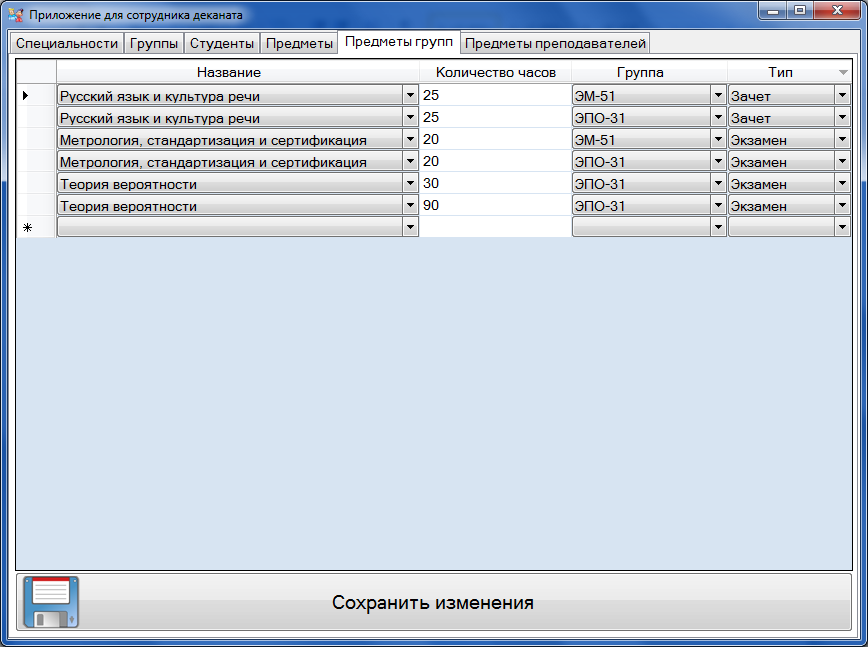


Рис. 15. Вкладка «Предметы групп»

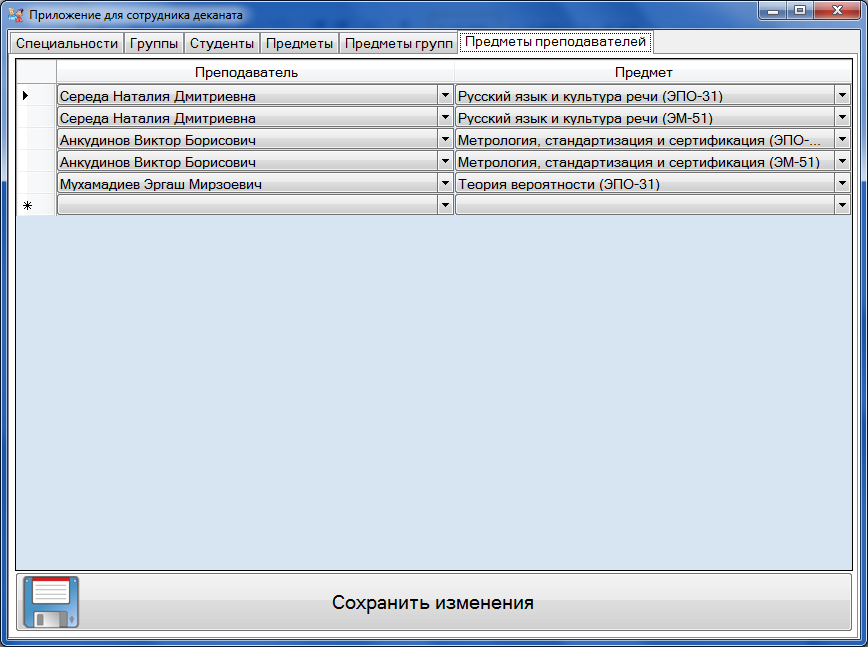


Рис. 16. Вкладка «Предметы преподавателей»

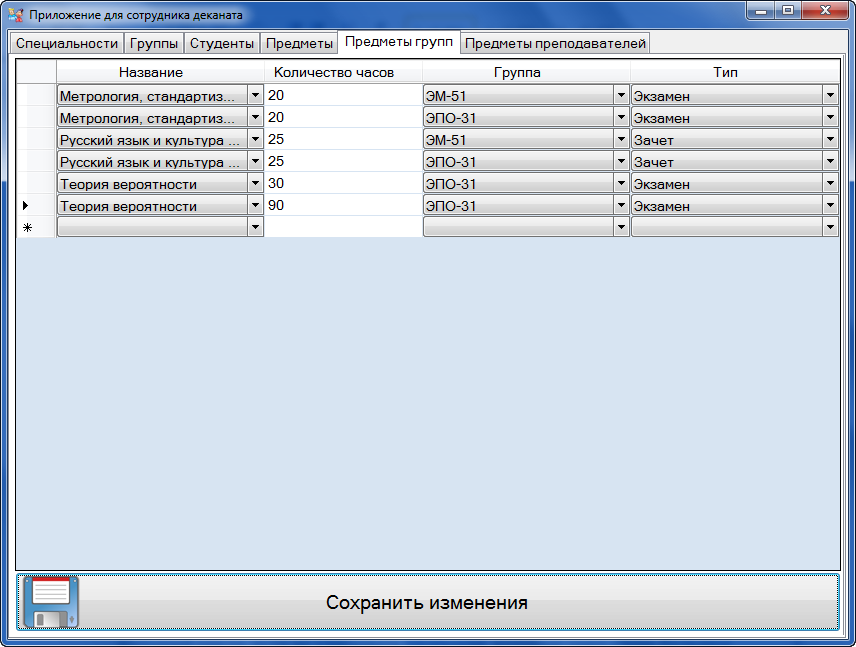


Рис. 17. Таблица с предметами групп

# Часть наиболее интересных запросов из приложения

Все запросы – банальные SELECT без WHERE из таблиц. Всё интересное – в следующих приложениях.

# Приложение для преподавателя

Преподаватель может изменять оценки у своих предметов, если ему выдана ведомость. Для каждого преподавателя есть личный пароль, чтобы он не смог изменять оценки у чужих предметов.

# Хранимый код для приложении

В данном приложении используется хранимая процедура для авторизации преподавателя

|  |
| --- |
| create or replace  PROCEDURE AUTH\_TEACHER  (  IDX IN teachers.id\_teacher%type  , PWD IN teachers.passwod%type  , ERR OUT NUMBER  ) AS  authed NUMBER;  BEGIN  SELECT COUNT(id\_teacher) into authed from teachers  WHERE (id\_teacher = IDX) AND (passwod = PWD);  if authed = 1 then  err := 0;  else  err := 2;  end if;    EXCEPTION  when TOO\_MANY\_ROWS then  err := 3;  when OTHERS then  err := 4;  END AUTH\_TEACHER; |

# Скриншоты приложения

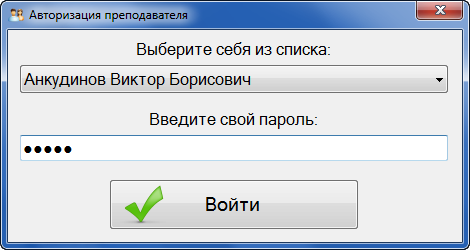


Рис. 18. Авторизация преподавателя

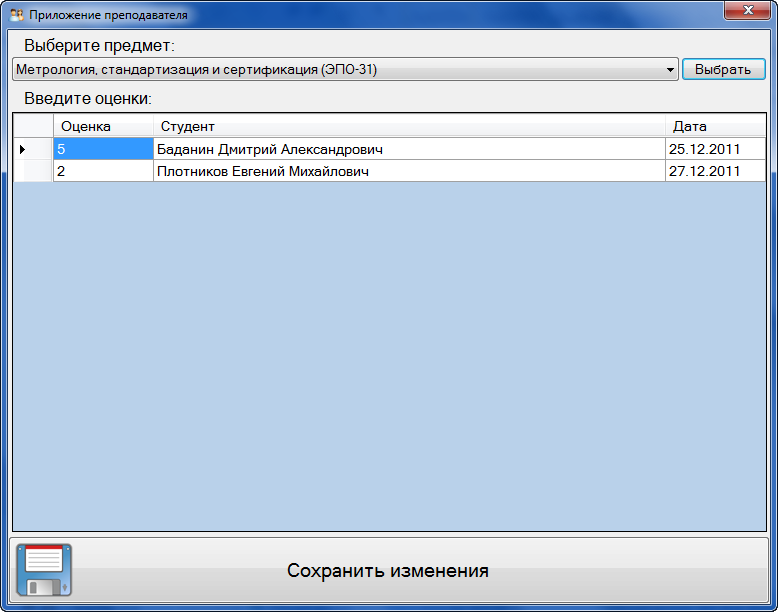


Рис. 19. Выставление оценок в приложении

# Часть наиболее интересных запросов из приложения

Выбор всех предметов преподавателя:

|  |
| --- |
| SELECT d.ID\_DISCIPLINE, d.ID\_GROUP, get\_discname\_with\_groupname(d.ID\_DISCNAME, d.ID\_GROUP) AS DISCNAME  FROM DISCIPLINES d, TEACHERS2DISCIPLINES td, TEACHERS t  WHERE d.ID\_DISCIPLINE = td.DISCIPLINES\_ID\_DISCIPLINE AND td.TEACHERS\_ID\_TEACHER = t.ID\_TEACHER AND (t.ID\_TEACHER = :tid) |

Выбор всех оценок из экзаменационной ведомости для определённого преподавателя:

|  |
| --- |
| SELECT  m.id\_mark, m.value\_mark, m.date\_mark, get\_fullname(s.last\_name, s.first\_name, s.third\_name) AS FULLNAME  FROM  marks m  JOIN  students s  ON  s.id\_student = m.id\_student  WHERE  (m.id\_teacher = :tid) AND (m.id\_discipline = :did) |

# Приложение декана

Декан может просматривать успеваемость факультета, курса и группы, а так же должников; может выдавать экзаменационные ведомости и печатать вкладыши к диплому; так же переводить студентов на другой курс; добавлять/изменять и удалять преподавателей.

# Хранимый код для приложения

В приложении используется хранимая процедура для выдачи экзаменационной ведомости преподавателю:

|  |
| --- |
| create or replace  PROCEDURE EXAM\_SHEET  (  IDGR IN NUMBER  , IDTE IN NUMBER  , IDDI IN NUMBER  , ERR OUT NUMBER  ) AS  marked NUMBER;  cursor studs is  SELECT id\_student FROM students WHERE (id\_status = 1) AND (id\_group = idgr);  BEGIN  err := 0;    FOR stud\_id IN studs LOOP    SELECT count(value\_mark) INTO marked FROM marks WHERE (id\_student = stud\_id.id\_student) AND (id\_teacher = idte) AND (id\_discipline = iddi);    IF marked = 0 THEN  INSERT INTO marks(id\_mark, value\_mark, id\_student, id\_teacher, id\_discipline) VALUES(null, 0, stud\_id.id\_student, idte, iddi);  END IF;    END LOOP;  EXCEPTION  WHEN CURSOR\_ALREADY\_OPEN THEN  err := 1;  WHEN DUP\_VAL\_ON\_INDEX THEN  err := 2;  WHEN INVALID\_CURSOR THEN  err := 3;  WHEN INVALID\_NUMBER THEN  err := 4;  WHEN LOGIN\_DENIED THEN  err := 5;  WHEN NO\_DATA\_FOUND THEN  err := 6;  WHEN NOT\_LOGGED\_ON THEN  err := 7;  WHEN PROGRAM\_ERROR THEN  err := 8;  WHEN STORAGE\_ERROR THEN  err := 9;  WHEN TIMEOUT\_ON\_RESOURCE THEN  err := 10;  WHEN TOO\_MANY\_ROWS THEN  err := 11;  WHEN VALUE\_ERROR THEN  err := 13;  WHEN ZERO\_DIVIDE THEN  err := 14;  WHEN OTHERS THEN  err := 15;  END EXAM\_SHEET; |

Представление для получения оценок факультета, курса (у представления в WHERE дописывается дополнительное условие), группы (так же во WHERE дописывается дополнительное условие)

|  |
| --- |
| CREATE OR REPLACE FORCE VIEW "DEKANAT"."VIEW\_MARKS" ("COURSE", "ID\_GROUP", "GROUPNAME", "FULLNAME", "VALUE\_MARK", "DATE\_MARK", "DISCNAME")  AS  SELECT G.COURSE,  G.ID\_GROUP,  GET\_GROUPNAME(G.NAME\_GROUP, G.COURSE, G.SUBGROUP) AS GROUPNAME,  GET\_FULLNAME(S.LAST\_NAME, S.FIRST\_NAME, S.THIRD\_NAME) AS FULLNAME,  M.VALUE\_MARK,  M.DATE\_MARK,  GET\_DISCNAME(D.ID\_DISCNAME) AS DISCNAME  FROM MARKS M,  STUDENTS S,  "GROUPS" G,  DISCIPLINES D  WHERE M.ID\_STUDENT = S.ID\_STUDENT  AND S.ID\_GROUP = G.ID\_GROUP  AND M.ID\_DISCIPLINE = D.ID\_DISCIPLINE  AND (S.ID\_STATUS = 1)  ORDER BY GROUPNAME,  FULLNAME; |

Представление для получения всех курсов на факультете:

|  |
| --- |
| CREATE OR REPLACE FORCE VIEW "DEKANAT"."VIEW\_COURSES" ("COURSE")  AS  SELECT DISTINCT course FROM groups ORDER BY course; |

# Скриншоты приложения

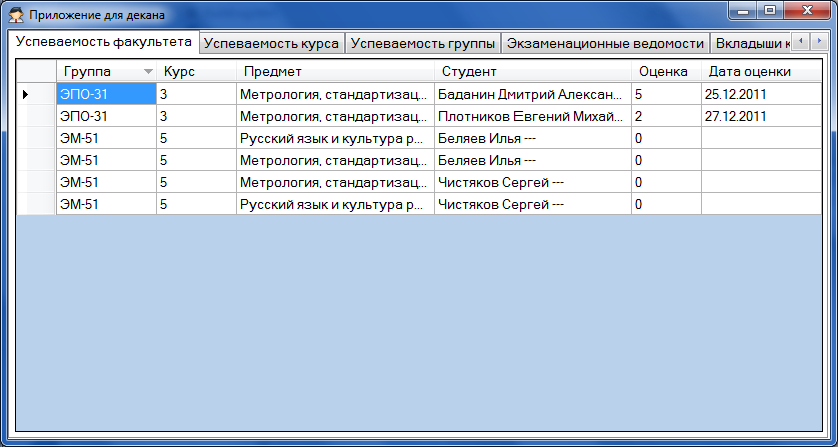


Рис. 20. Просмотр успеваемости факультета

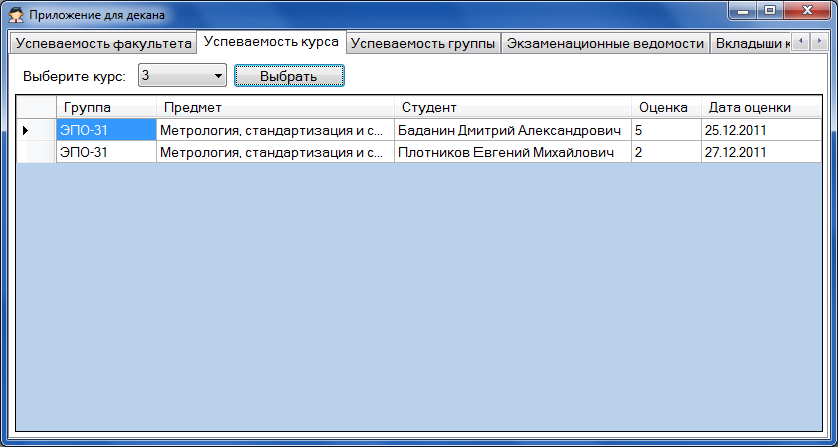


Рис. 21. Просмотр успеваемости курса

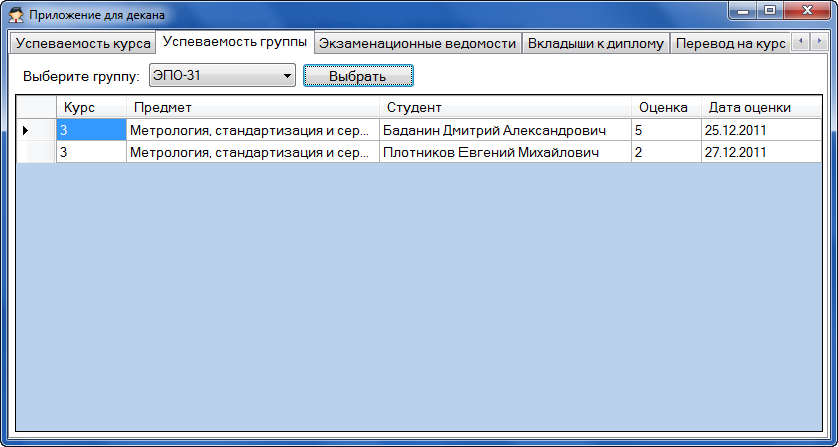


Рис. 22. Просмотр успеваемости группы

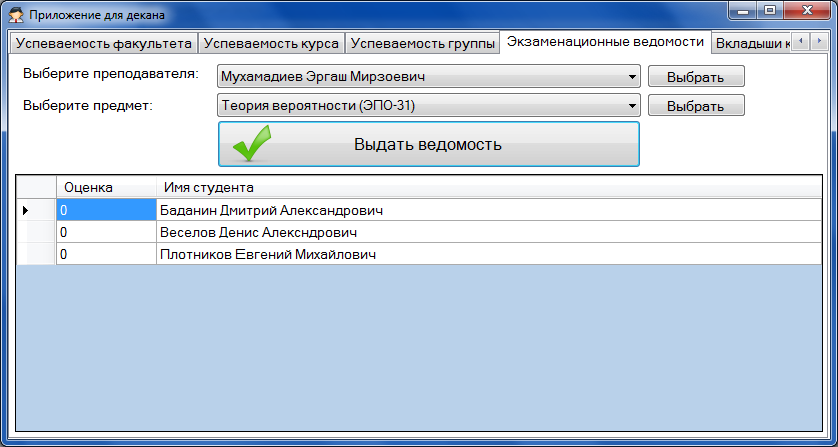


Рис. 23. Выдача экзаменационной ведомости для определённого преподавателя и группы

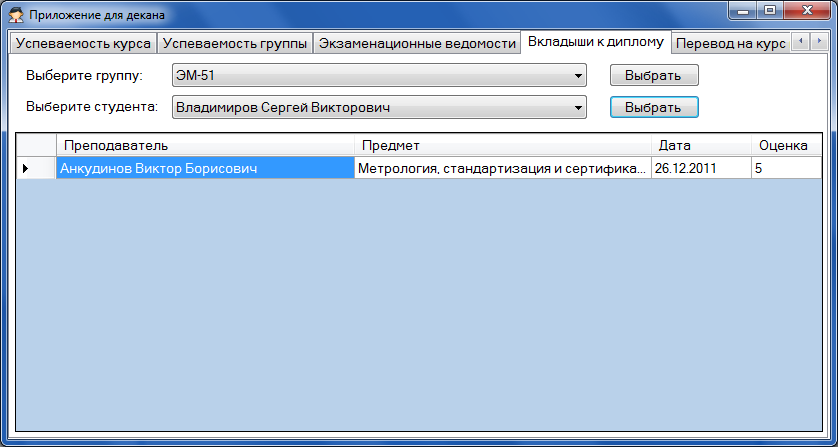


Рис. 24. Просмотр вкладыша к диплому для определённого студента из определённой группы

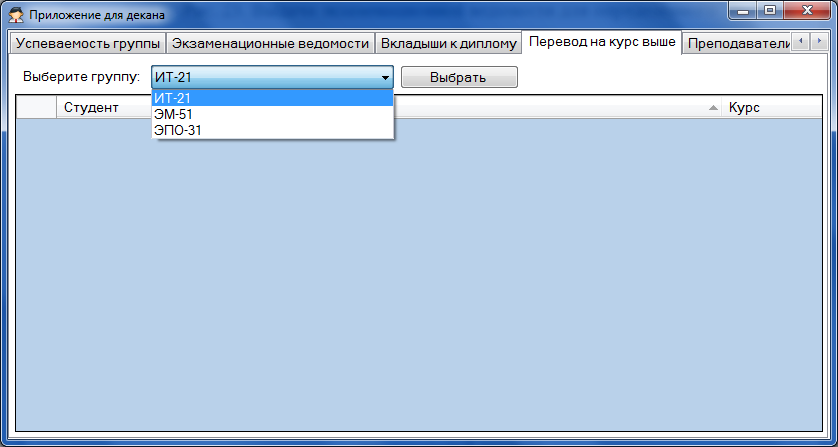


Рис. 25. Перевод на курс выше всех студентов из определённой группы

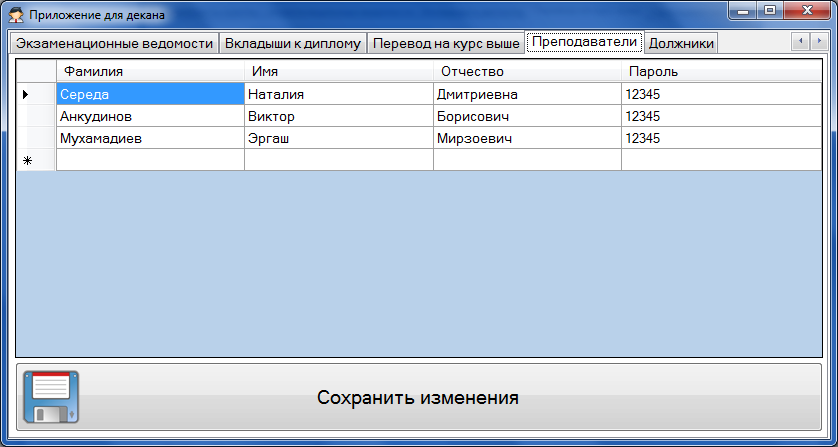


Рис. 26. Преподаватели на факультете

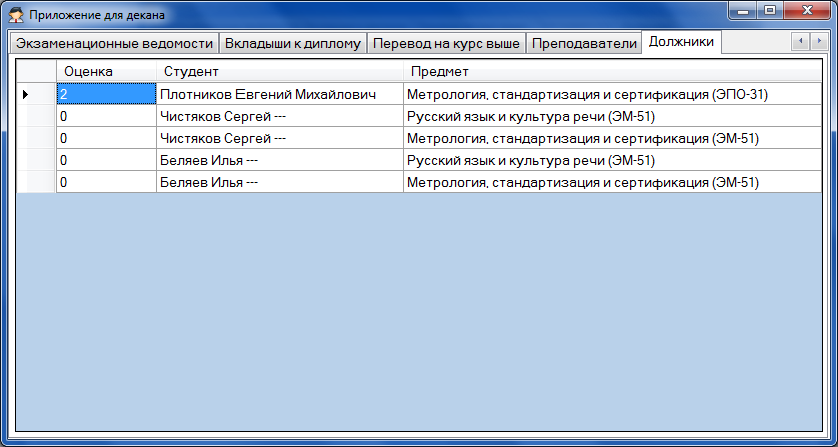


Рис. 27. Должники на факультете

# Часть наиболее интересных запросов из приложения

Выбор всех оценок для вкладыша к диплому для определённого студента из выбранной группы:

|  |
| --- |
| SELECT m.VALUE\_MARK, m.DATE\_MARK, get\_fullname(t.LAST\_NAME, t.FIRST\_NAME, t.THIRD\_NAME) AS FULLNAME, get\_discname(d.ID\_DISCNAME) AS DISCNAME  FROM MARKS m, TEACHERS t, DISCIPLINES d  WHERE t.ID\_TEACHER = m.ID\_TEACHER AND d.ID\_DISCIPLINE = m.ID\_DISCIPLINE AND (m.ID\_STUDENT = :sid) AND (m.ID\_DISCIPLINE IN  (SELECT ID\_DISCIPLINE  FROM DISCIPLINES d2  WHERE (ID\_GROUP = :gid)))  ORDER BY m.DATE\_MARK |

Выбор всех предметов для определённого преподавателя:

|  |
| --- |
| SELECT ID\_DISCIPLINE, get\_discname\_with\_groupname(ID\_DISCNAME, ID\_GROUP) AS DISCNAME, ID\_GROUP  FROM DISCIPLINES  WHERE (ID\_DISCIPLINE IN  (SELECT DISCIPLINES\_ID\_DISCIPLINE  FROM TEACHERS2DISCIPLINES  WHERE (TEACHERS\_ID\_TEACHER = :tid)))  ORDER BY DISCNAME |

Выбор всех групп, которые на 5-м курсе:

|  |
| --- |
| SELECT ID\_GROUP, get\_groupname(NAME\_GROUP, COURSE, SUBGROUP) AS GROUPNAME  FROM "GROUPS"  WHERE COURSE = 5  ORDER BY GROUPNAME |

Вывод всех студентов, которые только что были переведены на курс выше:

|  |
| --- |
| SELECT s.ID\_STUDENT, get\_fullname(s.LAST\_NAME, s.FIRST\_NAME, s.THIRD\_NAME) AS FULLNAME, g.COURSE  FROM STUDENTS s, "GROUPS" g  WHERE g.ID\_GROUP = s.ID\_GROUP AND (s.ID\_GROUP = :idg) AND (to\_date(s.EXPIRY\_STATUS, 'DD-MM-RR') = to\_date(sysdate, 'DD-MM-RR')) |

Выбор всех студентов должников:

|  |
| --- |
| SELECT  m.value\_mark, get\_fullname(s.last\_name, s.first\_name, s.third\_name) AS FULLNAME, get\_discname\_with\_groupname(d2.id\_discname, d2.id\_group) AS DISCNAME  FROM  marks m  JOIN  disciplines d2  ON  d2.id\_discipline = m.id\_discipline  JOIN  students s  ON  s.id\_student = m.id\_student  WHERE (m.id\_discipline in  (  SELECT  d.id\_discipline  FROM  disciplines d  WHERE  (d.id\_group IN  (  SELECT id\_group FROM groups  ))  )) AND (m.value\_mark < 3) AND (s.id\_status = 1) |

Выбор всех студентов 5-курсников из определённой группы для печати вкладыша к диплому:

|  |
| --- |
| SELECT ID\_STUDENT, get\_fullname(LAST\_NAME, FIRST\_NAME, THIRD\_NAME) AS FULLNAME  FROM STUDENTS  WHERE (ID\_STATUS = 4) AND (ID\_GROUP = :gid) |

Выбор оценок для всего факультета через преставление VIEW\_MARKS:

|  |
| --- |
| SELECT COURSE, GROUPNAME, FULLNAME, VALUE\_MARK, DATE\_MARK, DISCNAME  FROM VIEW\_MARKS |

Выбор оценок для определённого курса из представления VIEW\_MARKS:

|  |
| --- |
| SELECT COURSE, ID\_GROUP, GROUPNAME, FULLNAME, VALUE\_MARK, DATE\_MARK, DISCNAME  FROM VIEW\_MARKS  WHERE (COURSE = :cid) |

Выбор оценок для определённой группы через представление VIEW\_MARKS:

|  |
| --- |
| SELECT COURSE, FULLNAME, VALUE\_MARK, DATE\_MARK, DISCNAME, GROUPNAME  FROM VIEW\_MARKS  WHERE (ID\_GROUP = :gid) |