МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное бюджетное образовательное учреждение высшего образования

«Тихоокеанский государственный университет»

Кафедра «Программное обеспечение вычислительной техники и автоматизированных систем»

Реализация приложения по управлению распределённой БД

Лабораторная работа №6

по дисциплине «Проектирование приложений баз данных»

Выполнил студент Пшеничный Д. О.

Факультет, группа ФКФН, ПО(аб)-81

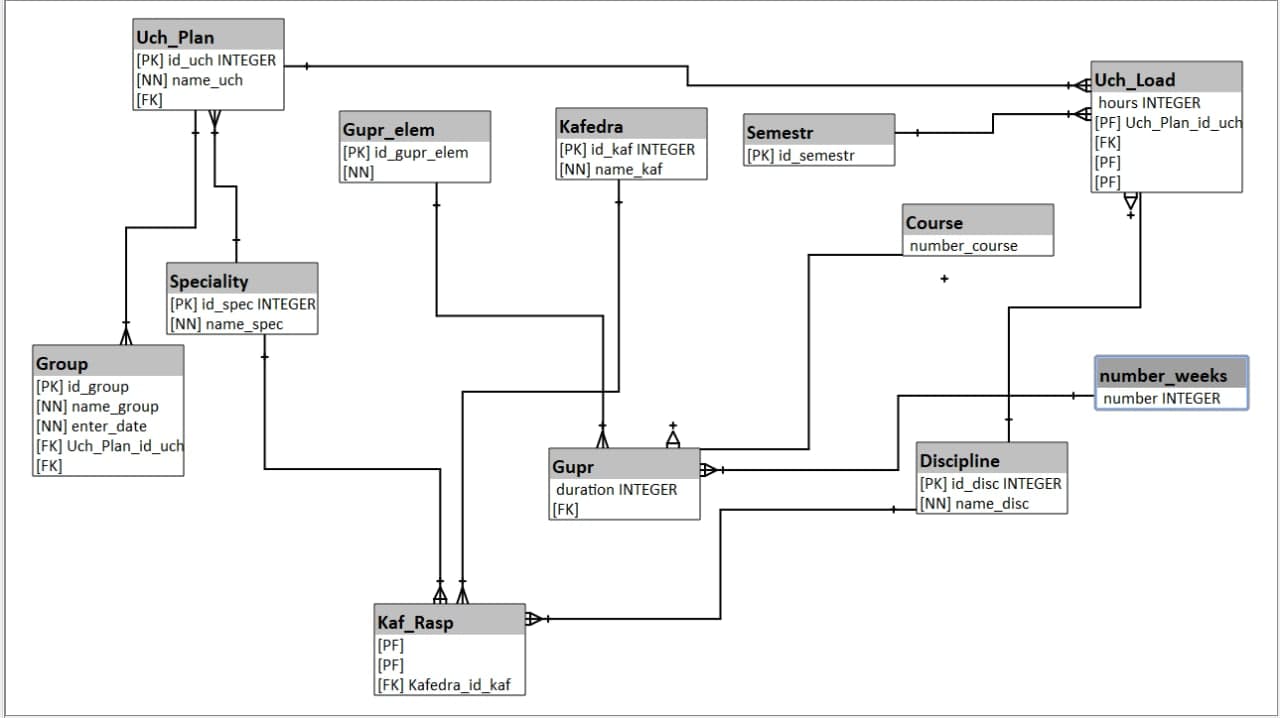
Руководитель работы  **Федосеев А.А.**

Хабаровск – 2021г.

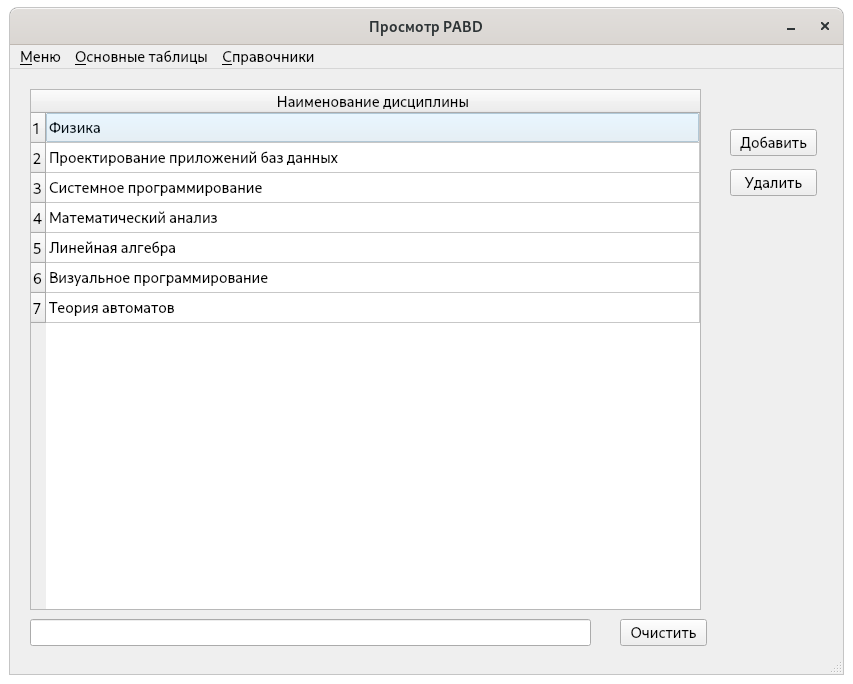
**Постановка задачи**

Реализовать функционирование ВСЕХ экранных форм приложения, созданного в лабораторной работе № 5. Требования к интерфейсу приложения, средствам разработки и процедуре тестирования остаются прежними. При защите работы в ОБЯЗАТЕЛЬНОМ порядке иметь сданную практическую работу № 2.

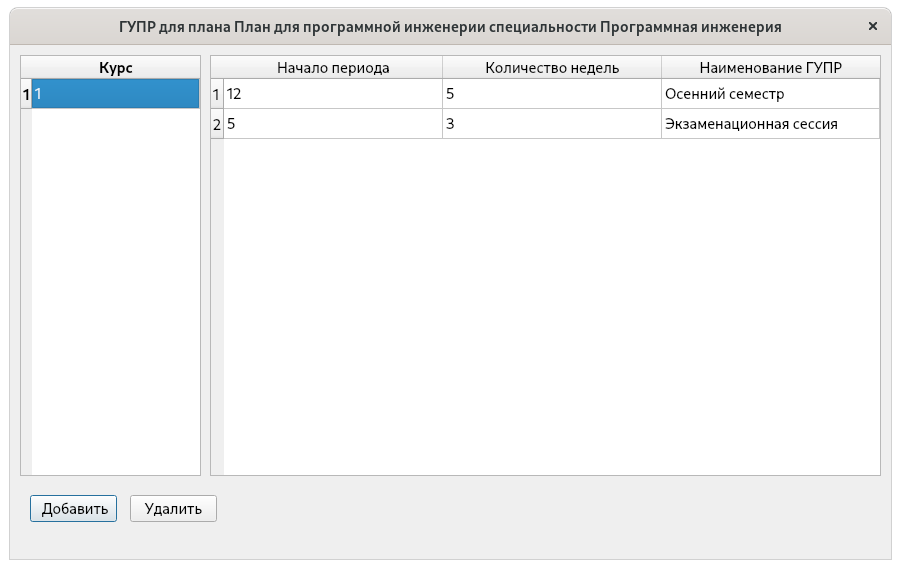
**Концептуальная модель**



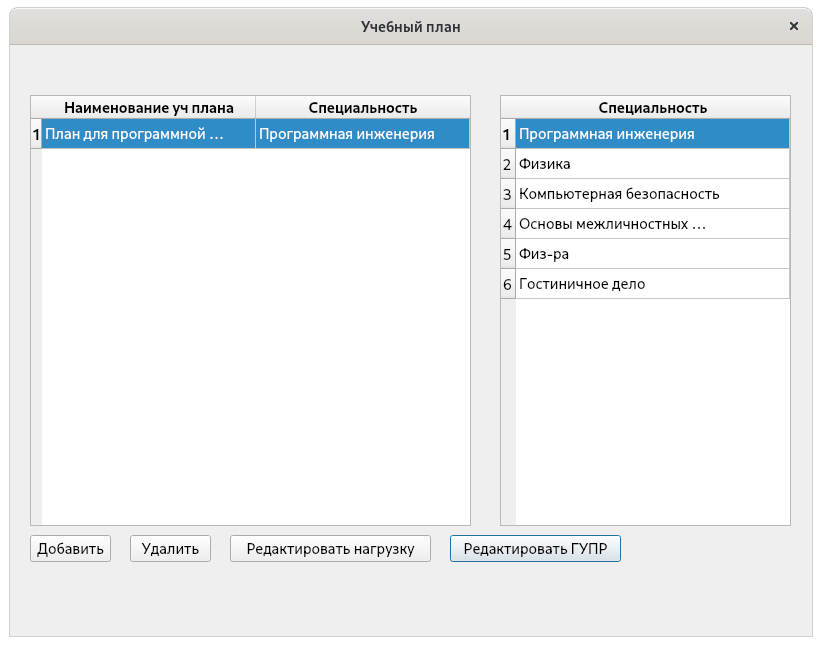
**Макет экранной формы справочника (A)**



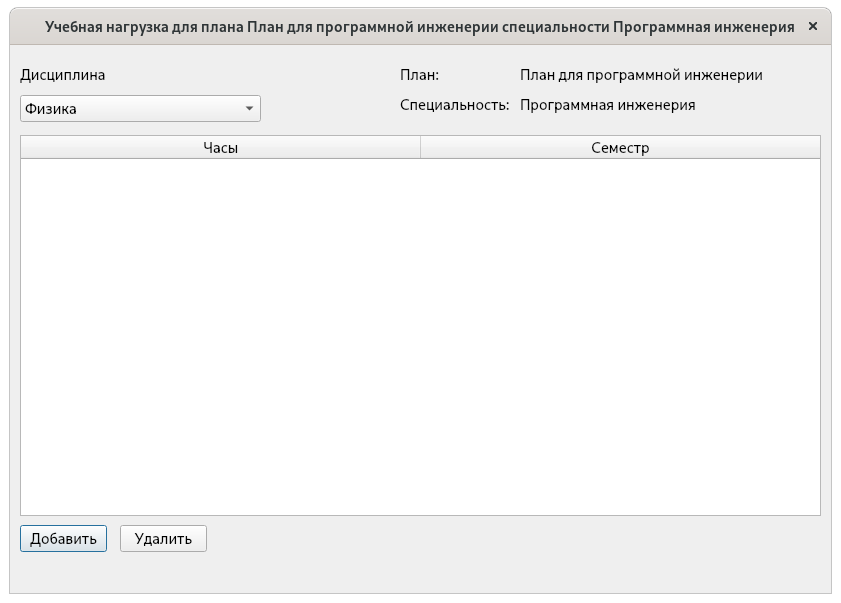
**Макет формы ГУПР (B)**



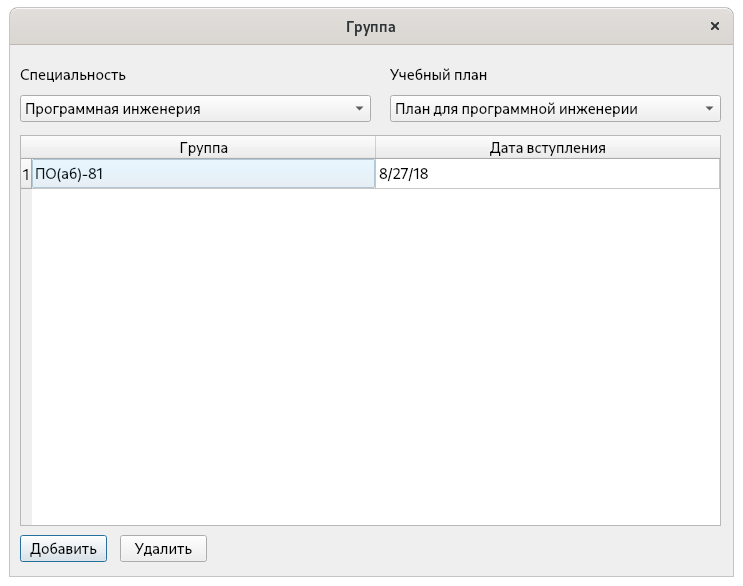
**Макет формы Учебный план (C)**



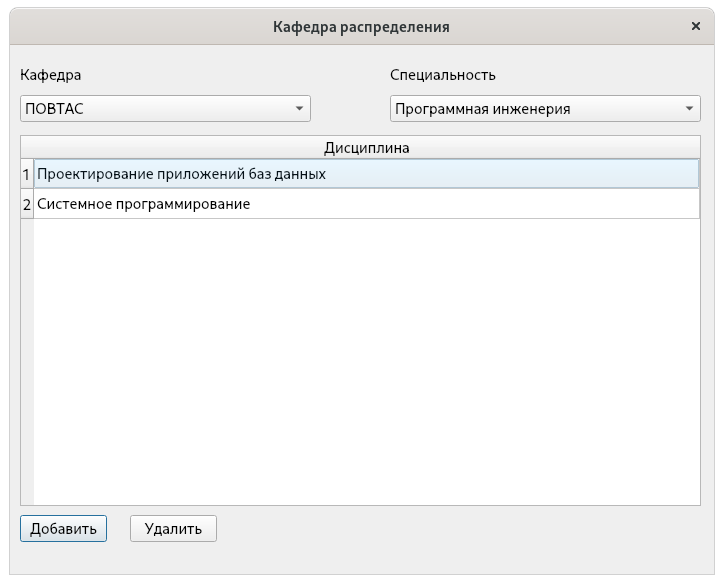
**Макет формы Учебная нагрузка (D)**



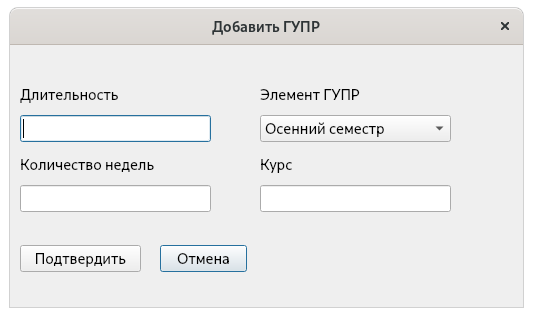
**Макет формы Группа (E)**



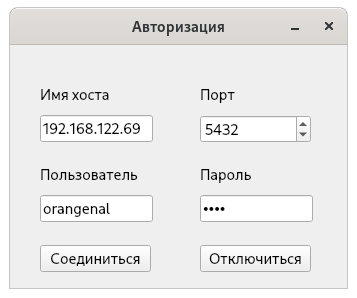
**Макет формы Кафедра распределения (F)**



**Универсальная форма добавления записи (G)**



**Форма авторизации (H)**



**Скрипт по созданию БД**

CREATE TABLE IF NOT EXISTS Discipline (id\_disc SERIAL, name\_disc VARCHAR(50) UNIQUE NOT NULL, PRIMARY KEY (id\_disc));

CREATE TABLE IF NOT EXISTS Speciality (id\_spec SERIAL, name\_spec VARCHAR(50) UNIQUE NOT NULL, PRIMARY KEY (id\_spec));

CREATE TABLE IF NOT EXISTS Uch\_Plan (id\_uch SERIAL, name\_uch VARCHAR(50) UNIQUE NOT NULL, Speciality\_id\_spec INTEGER, PRIMARY KEY (id\_uch), FOREIGN KEY (Speciality\_id\_spec) REFERENCES Speciality (id\_spec) ON DELETE CASCADE ON UPDATE RESTRICT);

CREATE TABLE IF NOT EXISTS Gruppa (id\_group SERIAL, name\_group VARCHAR(50) UNIQUE NOT NULL, enter\_date DATE NOT NULL, Uch\_Plan\_id\_uch INTEGER, Speciality\_id\_spec INTEGER, PRIMARY KEY (id\_group), FOREIGN KEY (Speciality\_id\_spec) REFERENCES Speciality (id\_spec) ON DELETE CASCADE ON UPDATE RESTRICT, FOREIGN KEY (Uch\_Plan\_id\_uch) REFERENCES Uch\_Plan (id\_uch) ON DELETE CASCADE ON UPDATE RESTRICT);

CREATE TABLE IF NOT EXISTS Gupr\_elem (id\_gupr\_elem SERIAL, name\_gupr\_elem VARCHAR(50) UNIQUE NOT NULL, PRIMARY KEY (id\_gupr\_elem));

CREATE TABLE IF NOT EXISTS Gupr (duration INTEGER, number\_course INTEGER, number\_weeks INTEGER, Gupr\_elem\_id\_gupr\_elem INTEGER, uch\_plan\_id\_gupr INTEGER, FOREIGN KEY (Gupr\_elem\_id\_gupr\_elem) REFERENCES Gupr\_elem (id\_gupr\_elem) ON DELETE CASCADE ON UPDATE RESTRICT, FOREIGN KEY (uch\_plan\_id\_gupr) REFERENCES Uch\_Plan (id\_uch) ON DELETE CASCADE ON UPDATE RESTRICT);

CREATE TABLE IF NOT EXISTS Kafedra (id\_kaf SERIAL, name\_kaf VARCHAR(50) UNIQUE NOT NULL, PRIMARY KEY (id\_kaf));

CREATE TABLE IF NOT EXISTS Kaf\_Rasp (Speciality\_id\_spec INT, Discipline\_id\_disc INTEGER, Kafedra\_id\_kaf INTEGER, PRIMARY KEY (Speciality\_id\_spec, Discipline\_id\_disc), FOREIGN KEY (Discipline\_id\_disc) REFERENCES Discipline (id\_disc) ON DELETE CASCADE ON UPDATE RESTRICT, FOREIGN KEY (Kafedra\_id\_kaf) REFERENCES Kafedra (id\_kaf) ON DELETE CASCADE ON UPDATE RESTRICT, FOREIGN KEY (Speciality\_id\_spec) REFERENCES Speciality (id\_spec) ON DELETE CASCADE ON UPDATE RESTRICT);

CREATE TABLE IF NOT EXISTS Uch\_Load (hours VARCHAR(50), Uch\_Plan\_id\_uch INTEGER, Speciality\_id\_spec INTEGER, Semestr\_id\_semestr INTEGER, Discipline\_id\_disc INTEGER, PRIMARY KEY (Uch\_Plan\_id\_uch, Semestr\_id\_semestr, Discipline\_id\_disc), FOREIGN KEY (Discipline\_id\_disc) REFERENCES Discipline (id\_disc) ON DELETE CASCADE ON UPDATE RESTRICT, FOREIGN KEY (Speciality\_id\_spec) REFERENCES Speciality (id\_spec) ON DELETE CASCADE ON UPDATE RESTRICT, FOREIGN KEY (Uch\_Plan\_id\_uch) REFERENCES Uch\_Plan (id\_uch) ON DELETE CASCADE ON UPDATE RESTRICT);

CREATE TABLE IF NOT EXISTS Log\_Users (id\_action SERIAL UNIQUE NOT NULL, User\_Name VARCHAR(50), TimeAction TIMESTAMP, TableAction VARCHAR(30), Action VARCHAR(50), PRIMARY KEY(id\_action));

**Фрагмент программы, раскрывающий работу справочной формы**

void MainWindow::OpenF(QAction\* a){ selectedTable = a->text(); switch (const\_hash(selectedTable.toStdString().c\_str())) { case 1105689965: formAutorization->show(); break; case 3566738188: selectTable("course"); break; case 34458455: selectTable("discipline"); break; case 2821064468: selectTable("speciality"); break; case 2581877950: selectTable("gupr\_elem"); break; case 1721633621: formUch\_Plan->SetDB(db); formUch\_Plan->Update(); formUch\_Plan->show(); break; case 1202426404: selectTable("kafedra"); break; case 4111733963: selectTable("number\_weeks"); break; case 3415898536: selectTable("semestr"); break; case 1436716544: formGupr->SetDB(db); formGupr->show(); break; case 3862366129: formGruppa->SetDB(db); formGruppa->show(); break; case 2958382714: formKaf->SetDB(db); formKaf->show(); break; case 625336143: formUch\_Load->SetDB(db); formUch\_Load->show(); break; *//Учебная* *нагрузка* }

}

void MainWindow::selectTable(QString nameTable){ ui->tableView->show(); table = new QSqlRelationalTableModel(0, db); table->*setTable*(nameTable); table->*setEditStrategy*(QSqlTableModel::*OnManualSubmit*); ui->tableView->*setModel*(*table*); ui->tableView->showColumn(0); ui->tableView->horizontalHeader()->setSectionResizeMode(QHeaderView::*Stretch*);

if(nameTable == "discipline") { table->*setHeaderData*(1, Qt::*Horizontal*, QObject::tr("Наименование дисциплины"), Qt::*DisplayRole*); ui->tableView->hideColumn(0); }

else if(nameTable == "course") { table->*setHeaderData*(0, Qt::*Horizontal*, QObject::tr("Номер курса"), Qt::*DisplayRole*); }

else if(nameTable == "number\_weeks") { table->*setHeaderData*(0, Qt::*Horizontal*, QObject::tr("Количество недель"), Qt::*DisplayRole*); }

else if(nameTable == "speciality") { table->*setHeaderData*(1, Qt::*Horizontal*, QObject::tr("Специальность"), Qt::*DisplayRole*); ui->tableView->hideColumn(0); }

else if(nameTable == "gupr\_elem") { table->*setHeaderData*(1, Qt::*Horizontal*, QObject::tr("Элемент ГУПР"), Qt::*DisplayRole*); ui->tableView->hideColumn(0); }

else if(nameTable == "kafedra") { table->*setHeaderData*(1, Qt::*Horizontal*, QObject::tr("Кафедра"), Qt::*DisplayRole*); ui->tableView->hideColumn(0); }

else if(nameTable == "semestr") { table->*setHeaderData*(0, Qt::*Horizontal*, QObject::tr("Семестр"), Qt::*DisplayRole*); }

table->*select*(); *//ui->tableView->*}

**Фрагмент программы, раскрывающий работу формы главный-подчинённый**

void Gupr::currentGuprChanged(QModelIndex cur\_ind, QModelIndex){ //qDebug() << "CHANGED"; if(cur\_ind.isValid()) { QSqlRecord record = gupr\_elTable->record(cur\_ind.row()); currentCourse = record.value("number\_course").toInt(); //QString id = record.value(""); //qDebug() << QString("gupr.number\_course=%1").arg(record.value("number\_course").toInt()); //guprTable->setFilter(QString("gupr.number\_course=%1").arg(record.value("number\_course").toInt())); } else { //guprTable->setFilter("gupr.number\_course=-1"); currentCourse = -1; }

UpdateFilter();}