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»

Проектирование системы экранных форм

Лабораторная работа №5

по дисциплине «Проектирование приложений баз данных»

Выполнил студент Пшеничный Д. О.

Факультет, группа ФКФН, ПО(аб)-81

Руководитель работы  **Федосеев А.А.**

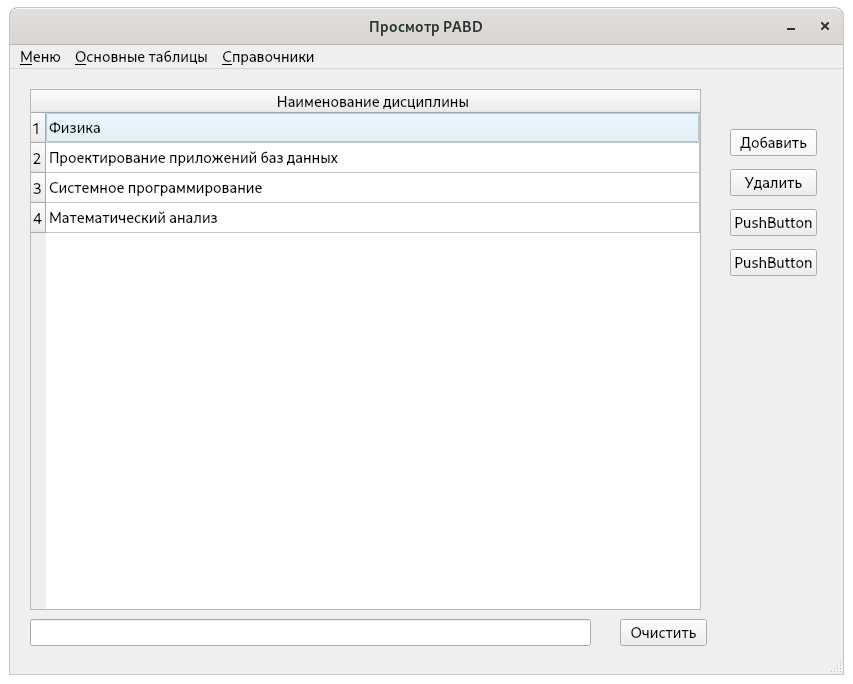
Хабаровск – 2021г.

**Постановка задачи**

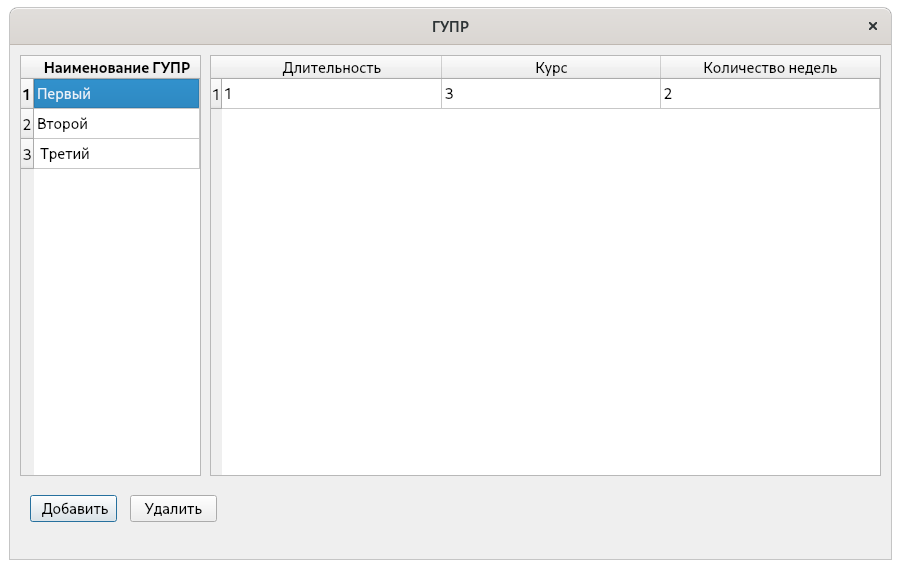
Используя библиотеку Qt построить макеты экранных форм по ведению распределённой базы данных. Структура базы данных определена в практической работе № 1, интерфейс программы — в практической работе № 2. Как минимум одна экранная форма должна быть построена по принципу “главный-подчинённый”, во всех формах по ведению справочных таблиц должны присутствовать строки-фильтры. Используемая СУБД — PostgreSQL, используемая библиотека для создания GUI — библиотека Qt.

Выполнить реализацию функционирования одной из экранных форм для редактирования справочных таблиц. Остальные экранные формы заполнить актуальными данными с использованием SQL‑запросов без возможности редактирования. Приложение тестируется на двух узлах ЛВС. На первом установлена СУБД и база данных. На втором работает клиентское приложение. При защите работы в ОБЯЗАТЕЛЬНОМ порядке иметь сданную практическую работу № 2.

**Скриншот активной экранной формы**



**Скриншот формы главный-подчинённый**



**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 активной формы**

void Gupr::Update(){ guprTable = new QSqlRelationalTableModel(0, db); gupr\_elTable = new QSqlRelationalTableModel(0, db);

guprTable->*setTable*("gupr"); gupr\_elTable->*setTable*("gupr\_elem");

guprTable->*setEditStrategy*(QSqlTableModel::*OnManualSubmit*);

guprTable->*setHeaderData*(0, Qt::*Horizontal*, QObject::tr("Длительность"), Qt::*DisplayRole*); guprTable->*setHeaderData*(1, Qt::*Horizontal*, QObject::tr("Курс"), Qt::*DisplayRole*); guprTable->*setHeaderData*(2, Qt::*Horizontal*, QObject::tr("Количество недель"), Qt::*DisplayRole*); guprTable->*setHeaderData*(3, Qt::*Horizontal*, QObject::tr("Наименование ГУПР"), Qt::*DisplayRole*); *//gupr\_elTable->setHeaderData(0,* *Qt::Horizontal,* *QObject::tr("ID* *ГУПР"),* *Qt::DisplayRole);* gupr\_elTable->*setHeaderData*(1, Qt::*Horizontal*, QObject::tr("Наименование ГУПР"), Qt::*DisplayRole*);

*//guprTable->setRelation(3,* *QSqlRelation("gupr\_elem",* *"id\_gupr\_elem",* *"name\_gupr\_elem"));*

guprTable->*select*(); gupr\_elTable->*select*();

ui->guprTableView->*setModel*(*guprTable*); ui->gupr\_elTableView->*setModel*(*gupr\_elTable*); ui->guprTableView->horizontalHeader()->setSectionResizeMode(QHeaderView::*Stretch*); ui->gupr\_elTableView->horizontalHeader()->setSectionResizeMode(QHeaderView::*Stretch*); ui->gupr\_elTableView->hideColumn(0); ui->guprTableView->hideColumn(3);

connect(ui->gupr\_elTableView->selectionModel(), SIGNAL(currentRowChanged(QModelIndex, QModelIndex)), this, SLOT(currentGuprChanged(QModelIndex, QModelIndex)));}

**Фрагмент программы, реализующий разрыв соединения с базой**

void MainWindow::Disconnect(){ model = new QSqlRelationalTableModel(0, db); ui->tableView->*setModel*(*model*); db.close(); formAutorization->hide();}