# Учет очереди на получение квартир по организациям (база данных)

Министерство общего и профессионального образования РФ

## Уральский государственный экономический университет

# Кафедра информационных систем в экономике

## Курсовая работа

**по основам алгоритмизации и языкам программирования**

Тема: ***Базы данных,***

***Учёт очереди на получение квартир***

***по организациям***

### Исполнитель: студент гр. ИСЭ-99-1

**Костюченко А.В.**

#### Руководитель: доцент Плещёв В. В.

## Екатеринбург

## 2001

**Организационно-экономическая сущность:** Необходимо разработать **комплекс по учёту очереди на получение квартир по организациям.** А именно создать **файл** кудавойдут:

Фамилия, номер очереди, номер приказа, код организации, код основания, дата , номер паспорта, численность семьи, адрес.

 Также необходимо создать два справочника:

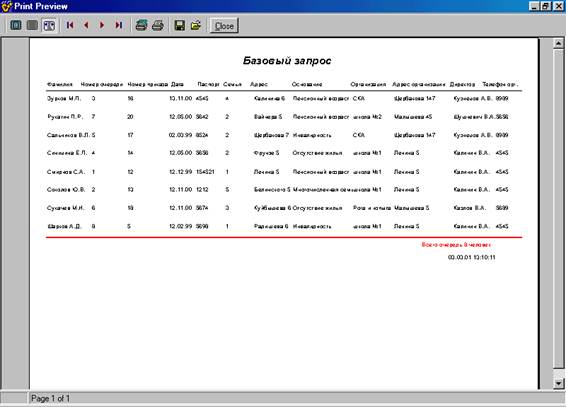
**Справочник по организациям** (код, наименование организации, адрес, директор, телефон);

**справочник по основаниям** (код, основание).

(вышеперечисленные таблицы создаются в Database Desktop)

Также необходимо создать **отчёт с итогами по каждой организации и базовый отчет.**

**Базовый отчет:**

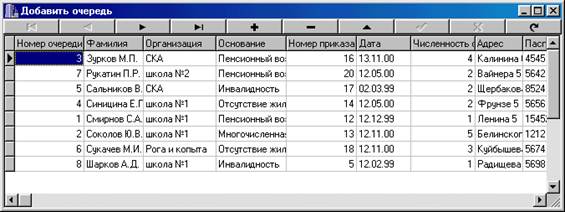


**Отчет по организациям:**

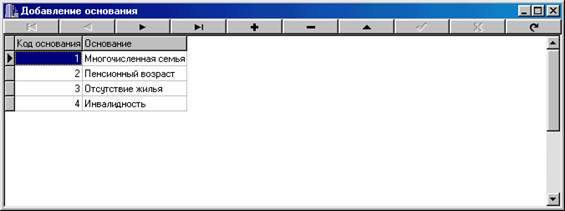
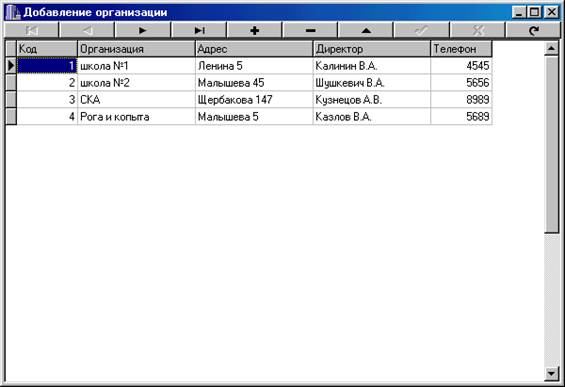


Программа предназначена для учёта очереди по организациям, а также для предоставления оперативной информации о очереди.

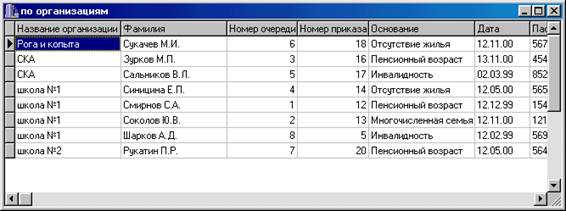
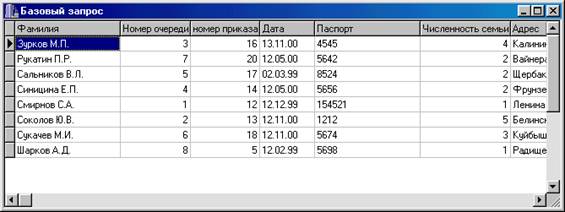
К **входящей** информации относятся: номер очереди, фамилия, организация, основание, номер приказа, дата, численность семьи, адрес , паспорт. Т. е файл:



Также Справочник 1 и 2:



К **выходящей** информации в отчёте относится запросы, включая базовый:



А так же к выходящей информации относятся отчёты (примеры отчётов приведены выше).

**Создание программы.**

Перед тем как непосредственно начать работу в C++Builder необходимо создать три таблицы. Зайдём в Database Desktope, нужно создать свой Alias .Для этого необходимо выбрать Tools -> Alias Manager,  нажимаем New, указываем имя алиаса в строке Database alias, потом выбирается путь к таблицам,нажимаем Browse (это есть тот  путь, где мы сохраняли таблицы). Потом нажимается OK, и подтверждается создание нового аляса. Alias .После того как Alias установлен нужно зайти в Working Direсtory  и выбрать свой Alias.Д алее выберем File ->New -> Table; После этого появится окно Table Type, здесь необходимо выбрать Paradox7. После выбора появляется следующее окно Create Paradox7 Table. В этом окне  необходимо указать название переменных (Field Name) , тип переменной (Type), размер строки, если это переменная строковая (Size),  и далее, если это поле является ключевым, указывается \* (поле Key). Примечание: Ключевые поля должны следовать друг за другом. Ключевые переменные являются общими для таблиц.  В квадратик (Required Field) помещается галка, если это поле обязательное. Потом это окно сохраняется, при этом указывается путь, этот же путь в последующем прописывается в Alais Manager. После создания трёх таблиц, заходим в главную таблицу (ту в которой содержаться общие для других таблиц переменные), и связываем таблицы. Делается это так: В Table Checks выбирается Referential. Далее нажимается Define. Появляется окно, в котором связываются переменные этой главной таблицы с другими переменными, содержащимися в файлах.

Теперь начинаем непосредственно работать с C++Builder.  Для начала создадим модуль: Для этого откроем New, и выберем  DataModule.  Начнём работу с создания базовых запросов, соответственно приведу пример, как создавать только один базовый запрос, все остальные делаются по аналогии.

В созданый модуль помещаем с панели инструментов DataAccess инструмент Query. Рядом на этом же модуле помещаем DataSource. Выделяем Query, и в Properties (свойствах) в DatabaseName выбирается аляс (именно тот который мы создали). Далее выделяем DataSource, и в свойствах в DataSet указывается название иконки базового запроса. Потом опять выделяем Query  и нажимаем правую кнопку мыши и выбираем SQLBuilder. Итак мы находимся в SQLBuilder. Сверху в панели инструментов в окне Database указывается созданный нами алиас. Чуть левее находится окошечко Table. Постепенно выводим на экран в той последовательности, в которой даны таблицы. Появляются окошечки. В левом верхнем углу каждого окошечка ставим галку. Потом выбираем из таблиц общие переменные. Нажимая на общую переменную, не отпуская мыши, перетаскиваем её на таблицу, где есть такая же переменная.  Если процедура прошла удачно, то появится(ятся) связь(и), и общие переменные выделятся. Потом начинаем группировать запрос. Выделяем только те переменные, которые требуются в запросе. Все остальные галки убираются. Далее сортируем, это делается в вкладке Sorting. Для того, чтобы убедится, что базовый запрос работает, запустим его, это делается при помощи кнопки Execute Query. Если всё работает, то просто закрываем SQLBuilder, при этом выдаётся сообщение о том, необходимо ли сохранить, сохраняем. В свойствах Query  Active делаем True. Далее создаём новую форму (форма первая необходима для размещения на ней меню,  т. е. Она является связующей). На форме размещается элемент DBGrid, который берется с панели Data Controls.  Под окном DBGrid размещается панель клавиш DBNavigator.  Сделаем активным окно DBGrid. В меню выбираем File -> Include Unit Hdr. Выбираются все Unit, и нажимается OK. В свойстве DataSource выбирается тот Souse, который подсоединён к данному Query. У DBNavigator  в свойстве DataSource так же выбирается нужный Source.  После этого в окне DBGrid должна появится таблица.  Теперь необходимо расшифровать переменные. Для этого сделаем активной иконку Query, далее нажмём правую кнопку мыши и выберем Fields Editor.  Далее появится окно, необходимо указатель мыши поставить в это пустое окно, и нажмём правую кнопку мыши. Появятся переменные, которые содержит запрос, выберем (выделим) все переменные и нажмём OK. Сделаем текущей одну из появившихся переменных, и в поле DisplayWidth поменяем название на то, которое соответствует этой переменной. Проделаем эту процедуру со всеми остальными переменными, после этого просто закроем окно с переменными, в самой таблице все переменные поменялись на соответствующие им набранные тексты. На этом  создание запроса заканчивается.

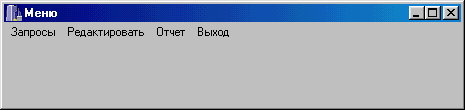
Создание форм происходит практически аналогично. В модуль помещается пара Table – DataSource. Сделаем активным Table. В свойстве DatabaseName укажем алиас, а также в TableName укажем имя таблицы, которую мы хотим вывести на форму. В DataSource в свойстве DataSet укажем соответствующий Table. Далее сделаем активным опять Table и укажем в свойстве Active     True. Потом создадим форму, на неё поместим DBGrid а также DBNavigator. Зайдём в File -> IncludeUnit и выберем все Unit которые там есть (при этом активным должно быть окно DBGrid). Далее в свойстве DataSource укажем нужную связь модуля и соответствующего DataSource. По идеи на форме должна появится таблица.

Теперь создадим отчёт. Для этого создадим новую форму. На ней разместим окно QuickRep. Найти этот инструмент можно в панели QReport. Два раза щёлкнем по свойству Bands, и поставим false в HasPageFoote и HasTitle. Далее в меню File выберем IncludeUnite, и выберем все Unit. Далее в свойстве DataSet укажем связь с соответствующим DataSouse. В поле Page Heater указывается то, что находится в шапке таблицы. Надпись можно создать при помощи инструмента QRLabel. В свойстве Caption этого инструмента вводим текст. Значения из таблицы могут быть отображены в отчёте следующим образом: Выбирается инструмент QRDBText. При помощи  свойства DataSet связываем этот инструмент с таблицей. Далее указываем DataField, здесь выбираем нежную нам переменную, которая будет меняться. Свойство ParentFont делаем true.   В шапке так же можно разместить номер листа или дату и время. Это делается при помощи инструмента QRSysData. В свойстве Data этого инструмента выбирается нужный формат. Далее под шапкой следует Column Header. Это тоже составная шапки. Далее идёт Group Header. В свойстве Expression указывается фильтрация таблицы, Свойство ForceNewPage указывается true что означает что фильтрация будет связана с появлением данных на новом листе. В поле Detail распологаются QRDBText, которые берутся из привязанной к отчёту таблицы или запроса.  В поле Group Footer распологаются промежуточные результаты. Промежуточным результатом может быть сумма и т. д. Нахождение её возможно благодаря инструменту QRExpr. В свойстве Expression задаётся формула. В поле Summary располагается конечный результат, который высчитывается по формуле. И эту формулу можно получить при помощи свойства Expression (сначала необходимо поместить в это поле инструмент QRExpr). Предварительный запуск этого отчёта осуществляется следующим образом: Необходимо правой кнопкой мыши нажать в область QuickRep, затем выбрать Preview.

Теперь необходимо все эти формы связать. Для это мы оставили самую первую форму. На ней размещаем инструмент MainMenu. Двойное нажатие на иконку позволяет открыть окно. В нём начинаем создавать меню. В Caption указываем названия пунктов меню. Связать названия можно при помощи двойного нажатия на заполненный текстом пункт меню, здесь пишется (если это просто таблица или запрос) Form1->Show(); причём номера форм разные. Если это базовый запрос, то необходимо при двойном нажатии на пункт меню написать Form1->QuickRep1->Preview(); Формы соответственно тоже разные. После всего этого необходимо сделать активной иконку MainMenu и опять подсоединить Unitы, File->Unit Includ. После этого форма 1 должна заработать.

**Обращение к программе** происходит посредством запуска программы. Запускается программа при помощи иконки в форме треугольника.

Появляется окно:



После этого можно работать с этим окном и вызывать необходимые формы или запросы.