

Далее нажимаем кнопку «зашифровать». Тут же появится наш зашифрованный текст. Чтобы удостовериться в правильности работы программы мы перенесём этот зашифрованный текст в верхнее пустое поле и нажмём кнопку «расшифровать». После чего у нас появляется расшифрованный текст, который совпадает с изначальным текстом, что доказывает, что наша программа работает правильно. Результат работы программы представлен на рисунке 3.

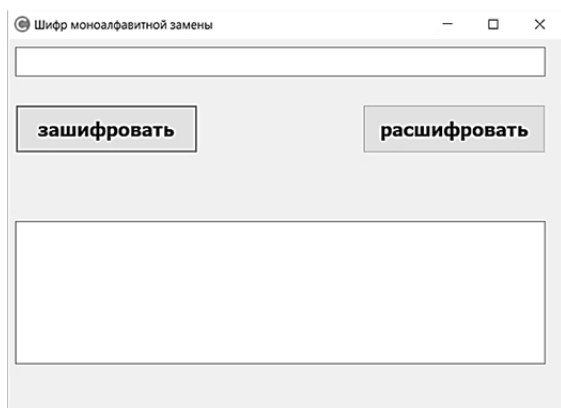


Рисунок 2 — Вид главной формы приложения

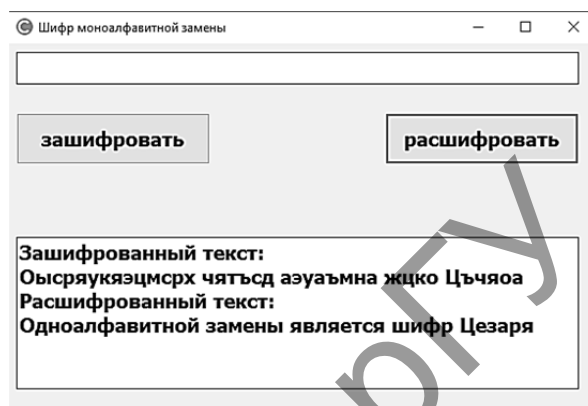


Рисунок 3 — Результат работы программы

Заключение. В ходе выполнения исследовательской работы было разработано приложение, предназначенное для реализации шифрования и дешифрования с помощью алгоритма моноалфавитной замены.

В ходе выполнения данного проекта были закреплены и получены новые знания по языку программирования C++ и среде C++ Builder, усовершенствованы навыки объектно-ориентированного программирования, изучены классы, шаблоны классов и особенности работы с ними.

На данном этапе функционал программы логически завершен. Данная программа проста в использовании, что позволяет любому пользователю быстро овладеть функционалом программы.

Список цитируемых источников

1. Шифр многоалфавитной замены [Электронный ресурс]. — Режим доступа : <https://thelib.info/informatika/1477165-shift-mnogoalfavitnoj-zameny/>. — Дата доступа : 12.02.2021.

УДК 004.942

А. Г. Кедышко, О. Д. Кравчук

Учреждение образования «Барановичский государственный университет», Барановичи, Республика Беларусь

СРАВНЕНИЕ МЕХАНИЗМА И ПРОИЗВОДИТЕЛЬНОСТИ ВЛОЖЕННЫХ ЗАПРОСОВ В ЯЗЫКАХ ЗАПРОСОВ: SQL И 1C

Введение. Окружающий мир представляет собой совокупность фактов, сведений и знаний, но пока они никак не собраны и не структурированы, они не представляют полезность для человечества. И именно это сохраненное структурирование называется данными, а то, что их хранит, — базой данных. Но пока эти данные просто где-то лежат собранные, они не представляют полезности. Таким образом, данные, которые получены и превращены в информацию представляют интерес для пользователя [1].

Цель баз данных заключается в упрощении работы пользователя, сокращении времени обработки информации, улучшении процесса пользования информацией, уменьшении вероятности возникновения ошибок. Базы данных являются основой большинства современных справочных систем, систем автоматизированного ведения бухгалтерского учета, экспертных систем и многих других программных продуктов. Это обусловлено, прежде всего, простотой и удобством этого типа хранения данных [2].

Для комфортного извлечения данных из базы данных, которые соответствуют параметрам пользователя используются запросы.

Задачей данного исследования является сравнение языков запросов 1C и SQL.

Целью исследования является разработка простейшей базы данных и осуществления запроса к ней для сравнения методики запросов и их продуктивности.

Объектом исследования является декларативный язык программирования, применяемый для создания, модификации и управления данными в реляционной базе данных SQL и язык запросов 1C.

Язык запросов в 1C 8 — это упрощенный аналог SQL. Отличие этих языков только в том, что SQL используется как для чтения, так и для изменения данных, в 1C же язык запросов используется только для чтения данных, для изменения данных применяется объектная модель данных.

Основная часть. В качестве предметной области выбрана узкая направленность «Сотрудники подразделений», которая в последствии может стать частью полноценной базы данных. Логическая модель базы данных представлена на рисунке 1. Схема метаданных для выбранной предметной области системы 1C:Предприятие представлена на рисунке 2.



Рисунок 1 — Логическая модель базы данных

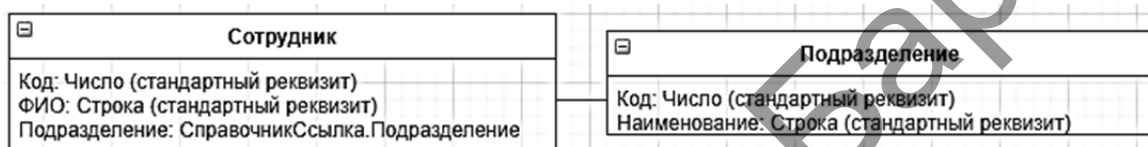


Рисунок 2 — Схема метаданных

Сравнение языков запросов будет проводиться на примере вложенного запроса. Вложенный запрос — запрос, который находится внутри другого запроса и встроен внутри условного оператора WHERE (ГДЕ). Такие запросы обычно используются для получения данных из двух и более таблиц. Они нужны чтобы данные из разных таблиц можно было соотнести и по зависимости осуществить выборку. У вложенных запросов есть и недостаток — зачастую слишком долгое время работы занимает запрос, потому что идет большая нагрузка на сервер.

В качестве примера запроса был выбран запрос вида: отобразить подразделения, в которых число сотрудников больше какого-то количества сотрудников.

Текст запросов на языке SQL и 1C представлен на рисунке 3.

```
SQL:
SELECT Запрос.Подразделение
FROM
  (SELECT Подразделение.Наименование as Подразделение,
   COUNT (Сотрудник.ФИО) AS Количество
   FROM
     Сотрудник, Подразделение
   where Сотрудник.id_подразделения=Подразделения.id_Подразделения
   Group By (Подразделение.Наименование)) AS Запрос
where Запрос.Количество>3

1C:
*ВЫБРАТЬ
| ВложенныйЗапрос.Наименование КАК Подразделение
| ИЗ
| (ВЫБРАТЬ
| | Сотрудник.Подразделение.Наименование КАК Наименование,
| | КОЛИЧЕСТВО(Сотрудник.Наименование) КАК КоличествоСотрудников
| | ИЗ
| | Справочник.Сотрудник КАК Сотрудник
| | СГРУППИРОВАТЬ ПО
| | Сотрудник.Подразделение.Наименование) КАК ВложенныйЗапрос
| ГДЕ
| ВложенныйЗапрос.КоличествоСотрудников > 5;
```

Рисунок 3 — Пример вложенного запроса

При сравнении написания запросов выявлены следующие различия:

1. Язык запросов, но 1C поддерживает также английский язык, что не является существенным различием.
2. Во внутреннем запросе 1C отсутствует ветвь «Где/Where», в отличие от языка SQL. Так как справочники/сущности в 1C связаны по ссылке, что дает возможность лишней не указывать связь между объектами, как это происходит в SQL.

Далее для сравнения производительности принято решение сравнения времени запросов к базе данных с разным количеством записей. Сравнение производилось на одном и том же компьютере с одинаковой конфигурацией. В качестве переменных, которые варьировались в запросе были выбраны: N подразделений и M — количество сотрудников. Распределение сотрудников по подразделениям было хаотичным. Сравнение времени выполнения вложенного запроса в зависимости от количества записей представлено на рисунке 4.

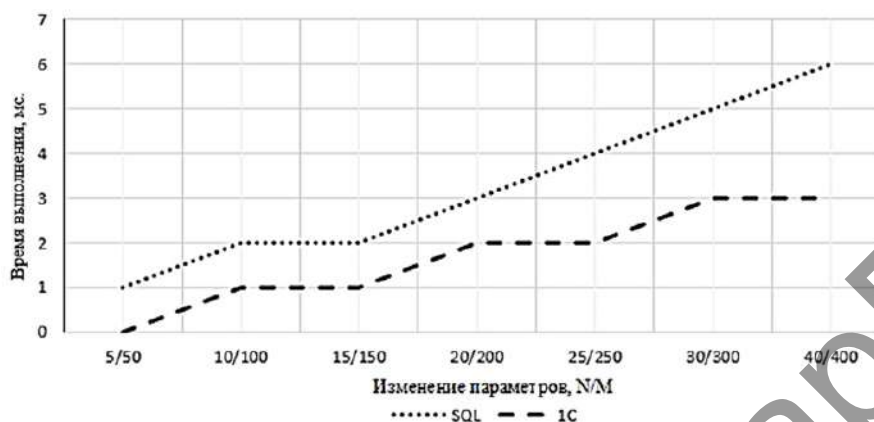


Рисунок 4 — Диаграмма сравнения времени выполнения вложенного запроса

При сравнении производительности запросов выявлено преимущество языка запросов 1C.

Заключение. Результатом исследования являются выявленные отличительные черты каждого из языков запросов 1C и SQL.

Выявлены сходства в следующих операторах: выборки, источников выборки, условий отбора, операторы агрегирования и группировки данных, объединения запросов, вложенных запросов. То есть операторы имеют одинаковое написание, за исключением возможности двуязычного написания в 1C.

Выявлены следующий различия:

- 1C предоставляет возможность обращение к свойствам объектов через точку или несколько точек;
- в 1C есть обращение к табличным частям объектов, что в SQL должно было быть представлено двумя и более сущностями;
- программный запрос 1C не позволяет изменять объекты, в отличие от языка SQL, в котором есть оператор ALTER и UPDATE, которые используются для изменения баз данных и изменения строк сущностей, соответственно. В свою очередь изменение данных в 1C происходит при помощи объектных запросов;
- время выполнения запросов на 1C меньше, чем на SQL, что может объясняться разной конструкцией полей (в SQL поля объединены связями, а в 1C — ссылками, что предоставляет более скорый доступ к полям).

Список цитируемых источников

1. Анализ данных и процессов : учеб. пособие / А. А. Барсегян [и др.]. — 3-е изд., перераб. и доп. — СПб. : БХВ-Петербург, 2009. — 512 с.
2. Кузнецов, С. Д. Основы баз данных : учебное пособие / С. Д. Кузнецов — 2-е изд., испр. — М. : Интернет-Университет Информ. Технологий ; БИНОМ. Лаб. знаний, 2007. — 484 с.

УДК 004.75

Д. С. Кислый, Г. М. Раковцы

Учреждение образования «Барановичский государственный университет», Барановичи, Республика Беларусь

АВТОМАТИЗИРОВАННАЯ СИСТЕМА ПРОВЕДЕНИЯ ШКОЛЬНЫХ ОЛИМПИАД ПО ПРОГРАММИРОВАНИЮ

Введение. Ежегодно проводятся олимпиады по программированию среди школьников и студентов в различных учреждениях образования в несколько этапов. При этом организация такого мероприятия неизбежно связана с обработкой большого количества бумажной информации. Большая часть действий является общей для всех подобных олимпиад: регистрация участников, тестирование их программ и подведение итогов,