

Министерство образования и науки Российской Федерации
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ
ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ
ВЫСШЕГО ОБРАЗОВАНИЯ
«САРАТОВСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ
ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
ИМЕНИ Н.Г. ЧЕРНЫШЕВСКОГО»

Кафедра Математического и компьютерного моделирования

ПОСТРОЕНИЕ ИНФОРМАЦИОННОЙ СИСТЕМЫ

«МАГАЗИНА ЕВРОСЕТЬ»

Автореферат бакалаврской работы

студентки 5 курса 561 группы

направление 09.03.03 - Прикладная информатика

механико-математического факультета

Кузнецовой Надежды Николаевны

Научный руководитель
проф., д.э.н., проф.

Л.В. Кальянов

Зав. кафедрой
д.ф. – м. н.

Ю.А. Блинков

Саратов 2017

ВВЕДЕНИЕ

Под базой данных понимают хранилище структурированных данных, при этом данные должны быть непротиворечивы, минимально избыточны и целостны.

Жизненный цикл любого программного продукта, в том числе и системы управления базой данных, состоит из стадий проектирования, реализации и эксплуатации.

Естественно, наиболее значительным фактором в жизненном цикле приложения, работающего с базой данных, является стадия проектирования. От того, насколько тщательно продумана структура базы, насколько четко определены связи между ее элементами, зависит производительность системы и ее информационная насыщенность, а значит - и время ее жизни.

Обычно БД создается для хранения и доступа к данным, содержащим сведения о некоторой предметной области, то есть некоторой области человеческой деятельности или области реального мира. Всякая БД должна представлять собой систему данных о предметной области. БД, относящиеся к одной и той же предметной области, в различных случаях содержат более или менее детализированную информацию о ней, причем таким способом, который заведомо исключает ненужную избыточность. В хорошо спроектированной базе данных избыточность данных исключается, и вероятность сохранения противоречивых данных минимизируется. Таким образом, создание баз данных преследует две основные цели: понизить избыточность данных и повысить их надежность.

Хорошо спроектированная база данных:

- удовлетворяет всем требованиям пользователей к содержимому базы данных. Перед проектированием базы необходимо провести обширные исследования требований пользователей к функционированию базы данных.
- гарантирует непротиворечивость и целостность данных. При проектировании таблиц нужно определить их атрибуты и некоторые правила, ограничивающие возможность ввода пользователем неверных значений. Для верификации данных перед непосредственной записью их в

- таблицу база данных должна осуществлять вызов правил модели данных и тем самым гарантировать сохранение целостности информации.
- обеспечивает естественное, легкое для восприятия структурирование информации. Качественное построение базы позволяет делать запросы к базе более "прозрачными" и легкими для понимания; следовательно, снижается вероятность внесения некорректных данных и улучшается качество сопровождения базы.
 - удовлетворяет требованиям пользователей к производительности базы данных. При больших объемах информации вопросы сохранения производительности начинают играть главную роль, сразу "высвечивая" все недочеты этапа проектирования.

Логически в современной реляционной СУБД можно выделить наиболее внутреннюю часть – ядро СУБД (часто его называют Data Base Engine), компилятор языка БД (обычно SQL), подсистему поддержки времени выполнения, набор утилит. В некоторых системах эти части выделяются явно, в других – нет, но логически такое разделение можно провести во всех СУБД.

В настоящее время базы данных являются неотъемлемой частью любого предприятия, фирмы. Поэтому нельзя недооценивать важность теории и практики разработки баз данных.

В данной дипломной работе реализуем информационную систему, поддерживающую работу «Магазина Евросеть». В качестве СУБД предлагается использовать БД PostgreSQL. Для разработки интерфейса пользователя в практической работе предлагается использовать библиотеку PyQt4.

1 Обоснование выбора ПО проекта

PostgreSQL — это реляционная база данных. PostgreSQL — это программный продукт с открытым исходным кодом и свободной (в прямом смысле этого слова) лицензией.

Одной из основных целей, которая была поставлена при разработке PostgreSQL является соответствие стандартам. PostgreSQL очень строго следовал ANSI SQL-92, SQL-99 (SQL-2 и SQL-3, соответственно), а теперь и ANSI SQL:2003. Мало кому удаётся похвастаться подобным соответствием стандартам.

В дополнение к стандартам PostgreSQL поддерживает множество полезных расширений. Примером мелкого, но полезного расширения не входящего в стандарт SQL является дополнение к условию для SELECT вида LIMIT/OFFSET, которые позволяют получить только указанные строки из результата запроса. PostgreSQL полностью поддерживает механизм транзакций (transactions), вложенные запросы (subselects), триггеры (triggers), представления (views), функциональные индексы, ссылочную целостность по внешнему ключу (foreign key referential integrity), изощрённые типы блокировок (sophisticated locking) и многое другое.

К названию PostgreSQL обычно прибавляется слово объектная, то есть полное название звучит как объектно-реляционная база данных PostgreSQL. Пользователю предоставляются необходимые инструменты для создания новых типов данных, функций, операторов и своих методов индексирования. Подобные возможности позволяют работать с довольно нестандартными данными, например, с картографическими объектами — PostGIS (<http://postgis.refractor.net/>).

Размер базы данных, управляемой PostgreSQL не ограничен, так же нет ограничения и на число строк в таблице. Таблица не может быть больше чем 32 Тбайта, а число столбцов в таблице не может быть больше 250–1600 в зависимости от типа данных.

Существуют родные интерфейсы для работы с PostgreSQL из языков Java (JDBC), Perl, Python, Ruby, C, C++, PHP, Lisp, Scheme и всего что может связаться через ODBC. PostgreSQL поддерживает хранимые процедуры ко-

торые можно написать на множестве языков программирования, включая Java, Perl, Python, Ruby, Tcl, C/C++ и родном для PostgreSQL PL/pgSQL.

Информация в реляционных базах данных хранится в виде обычных плоских двумерных таблиц. Доступ к данным в таблице можно получить по её имени. В таблице есть именованные столбцы (column) и строки (row) — очень простая и понятная концепция. Пользователю предоставляется набор операторов, результатом действий которых так же являются таблицы. Это особенность реляционной базы данных называется *замкнутость*. Это очень важное свойство, так в результате любых действий порождаются объекты того же типа, что и объект над которым совершались эти самые действия. Следствием замкнутости является возможность применять к результату все имеющиеся в наличии операторы. Иными словами можно пользоваться *вложенными выражениями*¹.

Qt — одна из ведущих платформ (текущая версия Qt4.8) для разработки приложений с графическим пользовательским интерфейсом (GUI) под большинство существующих ныне операционных систем. Также Qt — одноименный набор библиотек, лежащий в основе платформы. Платформа развивается компанией Trolltech и ориентирована на язык программирования C++, однако, ввиду ее удобства и популярности, сторонними разработчиками создаются привязки библиотеки к некоторым иным объектно-ориентированным языкам.

В данном разделе будет рассмотрена работа с PyQt — привязкой библиотеки Qt к языку программирования Python. В этом сочетании нашли соединение многие положительные элементы как библиотеки, так и языка. Достаточно только сказать, что ООП-идеология Qt шире ОО возможностей языка C++, поэтому Qt представляет надстройку над синтаксисом C++. Отсюда следует необходимость использования специального препроцессора для Qt-программы, дабы получить компилирующийся C++ код. Язык же Python интерпретируемый, динамически типизируемый и легко расширяемый. PyQt как раз и предоставляет соответствующее нуждам Qt расширение языка Python, которое позволяет ООП-идеологию Qt использовать в Python-

¹Вложенные выражения, это многоуровневые выражения, причём, использование имён реальных таблиц обязательно только на самом низком уровне. В остальных случаях в качестве объектов действия могут быть вычисляемые выражения.

е как «родную». С другой стороны, можно сказать даже больше: PyQt расширяет ООП-идеологию библиотеки Qt высокоуровневыми возможностями языка Python.

Использование языка Python позволяет на порядок ускорить процесс разработки приложения для Qt по сравнению с альтернативной разработкой на C++. Достигается это за счет интерпретируемости Python-программы (то есть отсутствует цикл компиляции-сборки), также за счет более высокого уровня языка, что позволяет в несколько раз сокращать объем пишущегося кода.

Кроссплатформенность как библиотеки Qt, так и интерпретатора Python, позволяет переносить разработанные на PyQt приложения из одной операционной системы в другую без каких-либо изменений.

2 Реализация БД «Магазин Евросеть»

БД «Магазин Евросеть» моделирует работу организации, занимающейся деятельностью в сфере продажи мобильных телефонов, портативной техники и аксессуаров к ним, продажи контрактов операторов.

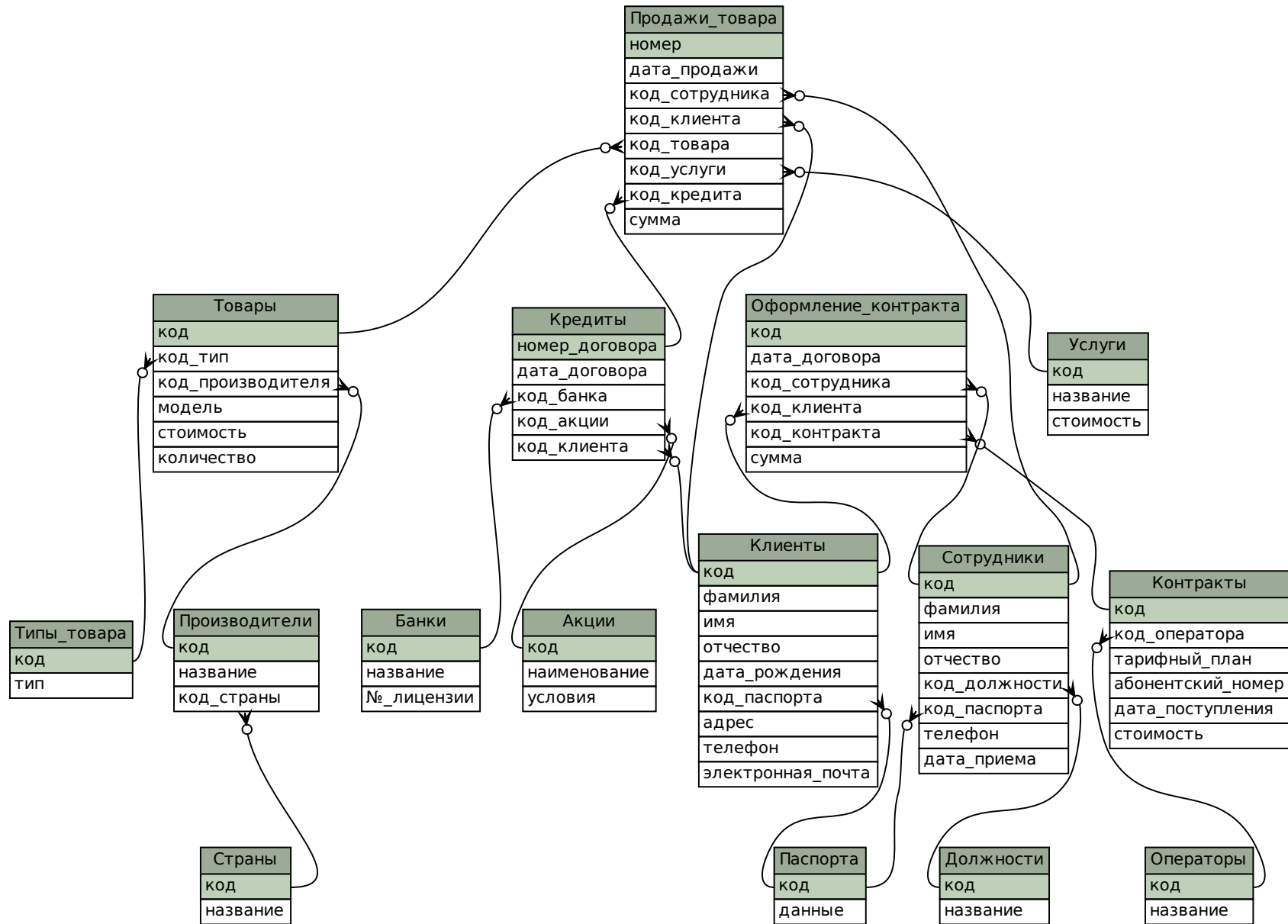
БД «Магазин Евросеть» позволяет пользователю вносить новые данные, вносить изменения в существующие данные, удалять ненужные объекты, формировать отчёты, просматривать и производить поиск необходимых данных.

Использование БД «Магазин Евросеть» предусматривает существенное упрощение и ускорение работы по учёту клиентов магазина, продажи товаров, оформления контрактов операторов, поступления товара на торговую точку, за счёт автоматизации операций производимых при добавлении нового клиента в базу данных магазина, составлении отчетов о поступлении товара, продажи товаров, регистрации контрактов.

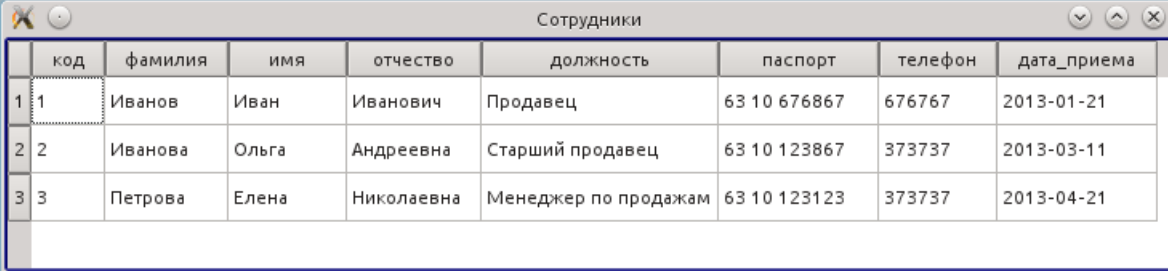
Входной информацией являются данные о сущностях «Клиенты», «Товары», «Сотрудники».

Выходной информацией можно считать отчеты, формируемые по желанию пользователя, а также результаты различных запросов к базе данных.

На рисунке 2.1 приведена полная ER (сущность – связь) диаграмма разработанной базы данных.



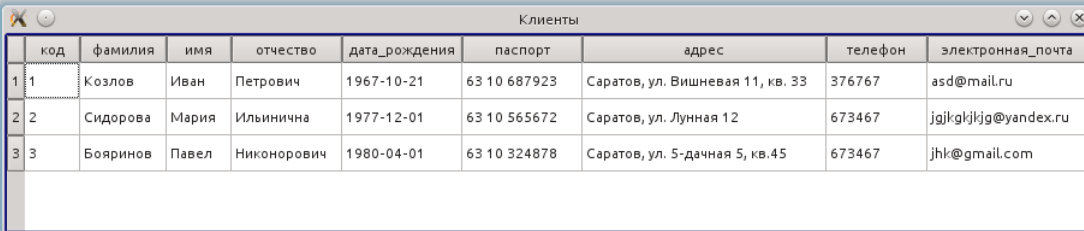
Реализация для таблицы «Сотрудники» приведена на рисунке 2.2.



	код	фамилия	имя	отчество	должность	паспорт	телефон	дата_приема
1	1	Иванов	Иван	Иванович	Продавец	63 10 676867	676767	2013-01-21
2	2	Иванова	Ольга	Андреевна	Старший продавец	63 10 123867	373737	2013-03-11
3	3	Петрова	Елена	Николаевна	Менеджер по продажам	63 10 123123	373737	2013-04-21

Рисунок 2.2 — Реализация для таблицы «Сотрудники»


Реализация для таблицы «Клиенты» приведена на рисунке 2.3.



	код	фамилия	имя	отчество	дата_рождения	паспорт	адрес	телефон	электронная_почта
1	1	Козлов	Иван	Петрович	1967-10-21	63 10 687923	Саратов, ул. Вишневая 11, кв. 33	376767	asd@mail.ru
2	2	Сидорова	Мария	Ильинична	1977-12-01	63 10 565672	Саратов, ул. Лунная 12	673467	jjkkgkjkjg@yandex.ru
3	3	Бояринов	Павел	Никонорович	1980-04-01	63 10 324878	Саратов, ул. 5-дачная 5, кв.45	673467	jhk@gmail.com

Рисунок 2.3 — Реализация для таблицы «Клиенты»

Реализация для таблицы «Операторы» приведена на рисунке 2.4.



	код	название
1	3	Менеджер
2	1	Оператор
3	2	Старший оператор

Рисунок 2.4 — Реализация для таблицы «Операторы»

Реализация для таблицы «Контракты» приведена на рисунке 2.5.

	код	оператор	тарифный_план	онентский_ном	дата_поступлени	стоимость
1	1	Оператор	Молодежный	ERT-763488	2013-02-23	2137,67
2	2	Старший оператор	Молодежный	ERT-768988	2013-03-12	7347
3	3	Менеджер	Умеренный	YTR-763488	2013-03-23	5443

Рисунок 2.5 — Реализация для таблицы «Контракты»

Реализация для таблицы «Продажи_товара» приведена на рисунке 2.6.

	номер	дата_продажи	сотрудник	товар	клиент	услуга	кредит	сумма
1	1	2013-01-23	Иванов	Козлов	RTY-5675	Обычная	1	67869,67
2	2	2013-01-23	Иванова	Бояринов	GHJ-02	Постоянная	2	123129
3	3	2013-01-23	Петрова	Сидорова	IOP-868	Скидочная	3	12869

Рисунок 2.6 — Реализация для таблицы «Продажи_товара»

ЗАКЛЮЧЕНИЕ

В дипломной работе сделан анализ предметной области «Магазин Евросеть». Для нее разработана структура БД в виде диаграмм сущность–связь и для СУБД PostgreSQL написан SQL для ее реализации. В качестве графического интерфейса пользователя для работы с информационной системой «Магазин Евросеть» на PyQt4 реализован интерфейс ко всем таблицам.