

**Флягин Артем Анатольевич**, магистрант,  
Ярославский государственный технический университет,  
г. Ярославль

Научный руководитель:  
**Маевский Вячеслав Константинович**, к.т.н.,  
доцент кафедры "Информационные системы и технологии",  
Ярославский государственный технический университет,  
г. Ярославль

**РАЗРАБОТКА БАЗЫ ДАННЫХ КЛИЕНТСКОГО ПРИЛОЖЕНИЯ  
ДЛЯ ИНФОРМАЦИОННОЙ СИСТЕМЫ ПО УПРАВЛЕНИЮ ЗАКАЗАМИ  
DEVELOPMENT OF A CLIENT APPLICATION DATABASE  
FOR ORDER MANAGEMENT INFORMATION SYSTEM**

**Аннотация:** В данной статье рассматривается разработка базы данных клиентского приложения для информационной системы по управлению заказами для СУБД PostgreSQL.

**Abstract:** This article discusses the development of a client application database for an order management information system for the PostgreSQL DBMS.

**Ключевые слова:** База данных, СУБД, PostgreSQL.

**Keywords:** Database, DBMS, PostgreSQL.

**Введение**

На сегодняшний день без систем управления заказами (OMS) не обходится не одно серьёзное предприятие, они являются неотъемлемой частью малого, среднего и крупного бизнеса занимающиеся доставкой. OMS играют ключевую роль в обеспечении своевременной и точной обработки заказов. Внедрение данной технологии на предприятие позволит:

- сократить время обработки и получения оперативных данных для принятия управленческих решений;
- повысить качество услуг доставки;
- повысить удобство поиска информации по доставке;
- повысить эффективность бизнеса в целом.

Целью научно-исследовательской работы является разработка базы данных клиентского приложения для информационной системы по управлению заказами. Целью автоматизации является решение ряда проблем существующие на предприятии.

Система позволит:

- сократить время выполнения заказа;
- сократить время обработки и получения оперативных данных для принятия управленческих решений;
- повысить качество услуг доставки;
- повысить удобство поиска информации.

Объект исследования – ИП Костицын.

Предмет исследования – клиентского приложения для работы с базой данной, а также технологии, которыми она создавалась.

**Выбор оптимального варианта модели базы**

Модель базы данных имеет решающее значения для успешной работы приложения. Рассмотрим основные используемые модели на сегодняшний момент.

Как указано в [1] иерархическая модель базы данных это связный граф типа дерева, вершины (типы) которого расположены на разных иерархических уровнях.



На основании литературных источников [1] сетевая модель данных – позволяет отображать взаимосвязи элементов данных в виде произвольного графа, обобщая тем самым иерархическую модель данных. Представлены в виде записей и связей. Запись может иметь множество как подчиненных ей записей, так и записей, которым она подчинена.

В настоящее время большинство информационных систем используют Реляционную модель базы данных. На основании источника [1], в реляционной БД - информация хранится в одной или нескольких связанных таблицах. Каждая запись в таблице идентифицирует один. Таблица устроена по принципу матрицы, где строки и столбцы отображаются как записи и поля. Обычно каждая отдельная таблица описывает набор либо реальных объектов, либо абстрактных концепций, либо событий одного типа.

В соответствии с [1], постреляционная модель базы данных – расширенная реляционная модель, снимающую ограничение неделимости данных, хранящихся в ячейках таблицы.

Многомерные базы данных представляют данные в виде кубов, поддерживая неограниченное число значений в поле, и находят свое применение там, где необходима эффективная и простая работа с большими массивами символьной информации.

Таблица 1

	Плюсы	Минусы
Иерархическая	<ul style="list-style-type: none"><li>– скорость поиска нужных данных;</li><li>– эффективность при организации связанных данных.</li></ul>	<ul style="list-style-type: none"><li>– усложняются операции включения и удаления;</li><li>– невозможно реализовать отношения «много-ко-много»;</li><li>– сложность обновления структуры;</li><li>– проблемы с масштабируемостью.</li></ul>
Сетевая	<ul style="list-style-type: none"><li>– возможность иметь у каждой записи несколько родителей;</li><li>– эффективно управляет повторяющимися данными;</li><li>– эффективная реализация обработки данных.</li></ul>	<ul style="list-style-type: none"><li>– сложность обновления структуры;</li><li>– проблемы с целостностью данных.</li></ul>
Реляционная	<ul style="list-style-type: none"><li>– нормализации отношений позволяет получать бд с заданными характеристиками;</li><li>– простота, понятность и удобство физической реализации;</li><li>– независимость данных;</li><li>– позволяет лаконично описать основные операции над данными.</li></ul>	<ul style="list-style-type: none"><li>– сложность описания иерархических и сетевых связей;</li><li>– относительно низкая скорость доступа и большой объем внешней памяти;</li><li>– трудность понимания из-за большого количества таблиц;</li><li>– подходит для описания не всех предметных областей.</li></ul>



Постреляционная	<ul style="list-style-type: none"> <li>– нормализации отношений позволяет получать бд с заданными характеристиками;</li> <li>– простота, понятность и удобство физической реализации;</li> <li>– независимость данных;</li> <li>– позволяет лаконично описать основные операции над данными.</li> <li>– более наглядное представление данных, нет необходимости соединять таблицы;</li> <li>– снимает ограничение неделимости данных, допуская многозначные поля, значения которых состоят из подзначений.</li> </ul>	<ul style="list-style-type: none"> <li>– сложность обеспечения целостности и непротиворечивости данных;</li> <li>– подходит для описания не всех предметных областей.</li> </ul>
Многомерная	<ul style="list-style-type: none"> <li>– эффективны в анализе большого объема данных, имеющих временную связь.</li> </ul>	<ul style="list-style-type: none"> <li>– неэффективное использование памяти;</li> <li>– стандартные операции требуют больших временных затрат.</li> </ul>

В результате анализа моделей в таблице 1 был сделан выбор в пользу реляционной базы данных, по следующим причинам:

- данная область подходит для представления в виде таблиц;
- предыдущая база предприятия была в виде реляционной модели, что упростит разработку;
- низкое время обработки и получения информации.

**Выбор конкретной СУБД**

В данный момент времени на рынке представлено множество реляционных баз данных, их полное сравнение можно увидеть на таблице 2.

Таблица 2

	PostgreSQL	MySQL	Oracle	Microsoft SQL Server
Лицензия	PostgreSQL License	GPL	Проприетарная	Проприетарная
Стоимость	бесплатно	бесплатно	≈ 74041,13 руб	≈ 90622,07 руб
Масштабируемость	Вертикальное и горизонтальное	Вертикальное и горизонтальное	Вертикальное и горизонтальное	Вертикальное и горизонтальное



Поддержка сервера БД	клиент-сервер	клиент-сервер	клиент-сервер	клиент-сервер
Поддерживаемые язык	C, C++, Java, Лисп, Perl, PHP, Python, Ruby, Smalltalk и Tcl	Java, PHP, C++, Python, Ruby, Visual Basic, Delphi, Go R, Perl, Scheme, Tcl, Haskel и Eiffel.	C, C++, Java, COBOL, PL/SQL, and Visual Basic	Java, PHP, C++, Python, Ruby, Visual Basic, Delphi, Go и R
Процедурные	PL/pgSQL, PL/Tcl, PL/Perl, and PL/Python.	SQL, SQL/PSM	PL/SQL, Java и JavaScript	T-SQL, C# или Visual Basic.NET
Документация	++	+++	++	+++
Плагины	Да	Да	Да	Да

Исходя из рассмотренных характеристик баз данных в таблице 2, был сделан выбор в пользу

PostgreSQL. Данная БД подходит следует по следующим:

- в настоящее время это оптимальный вариант по соотношению цена-качество;
- поддерживает практически все современные языки программирования;
- имеет огромное сообщество.

### Проектирование схемы БД

После обследования предметной область и с учетом новых требований, была разработана схема базы данных (рис.1).

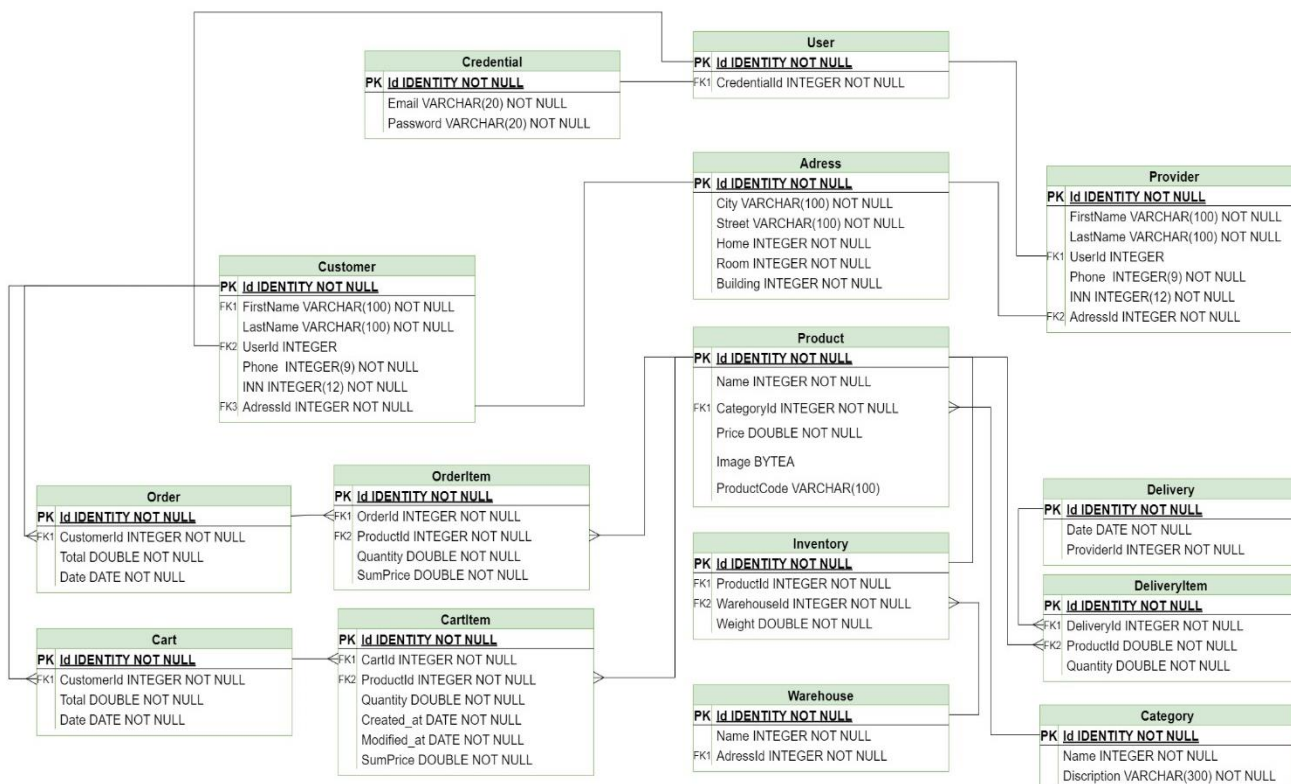


Рис. 1. ER-диаграмма



### **Заключение**

В ходе научно-исследовательской работы была разработана база данных клиентского приложения для информационной системы по управлению заказами, позволяющая автоматизировать процесс принятия заказа товара и его учет. Данная система позволит ускорить работу операторов, руководителя и предприятия в целом.

### *Список литературы:*

1. Скакун В.В. Системы управления базами данных Мн.: БГУ, 2006. – 120 с.

