**Государственное бюджетное профессиональное образовательное учреждение**

**Астраханской области «Астраханский колледж вычислительной техники»**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **К защите допустить:** | | | | | | | |
| **Старший методист** | | | | | | | |
|  |  | | | **Л.О. Земцова** | | | |
| **«** |  | **»** |  | | **20** |  | **г.** |

**курсовая работа**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| по МДК | 02.02 | Технология разработки и защиты баз данных | | | | | |
|  | | | | | | | |
|  | | 6 семестр | | | | | |
| Тема курсовой работы | | Разработка базы данных | | | | | |
|  | | «Грузоперевозки» | | | | | |
|  | |  | | | | | |
| Аквт.09.02.07.Кр00.17ПЗ | | | | | | | |
| Листов: | | | 29 |  | | | |
|  | |  | | | | | |
|  | |  | | | | | |
| Разработчик | |  | | | Студент гр. | | ПБ – 32 |
|  | |  | | | А.Ю. Пиров | | |
|  | |  | | |  |  | |
| Руководитель | |  | | | Преподаватель | | |
|  | |  | | | И.А. Храмцовский | | |

**2023**

|  |
| --- |
| УТВЕРЖДЕНО |
| Цикловой комиссией |
| специальности 09.02.07 |
| ПЦК \_\_\_\_\_\_\_\_\_ Ю.С. Андрианова |
| «\_\_\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_\_\_ 2022 г. |
| Специальность | *09.02.07 Информационные системы и программирование* | | |
| МДК 04.01 | *Технология разработки и защиты баз данных* | | |
| Группа | *ПБ– 32* | | |
| **ЗАДАНИЕ** | | | |
| **на курсовое проектирование** | | | |
| ***Пиров Амик Юрикович*** | | | |
| (фамилия, имя, отчество студента) | | | |
| Тема курсового проекта | | ***Разработка БД*** | |
| ***«Грузоперевозки»*** | | | |

Содержание пояснительной записки курсового проекта

Содержание

Введение

1. Проектирование базы данных
2. Анализ и описание предметной области информационной системы
3. Проектирование концептуальной модели
4. Построение логической модели, схемы базы данных
5. Выбор СУБД
6. Реализация базы данных
7. Физическое проектирование
8. Написание исходного кода БД
9. Написание запросов, функций и хранимых процедур

Заключение

Список использованных источников

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Дата выдачи задания | « |  | » |  | |  | 20 |  | г. |
| Руководитель |  | | | | И.А. Храмцовский | | | | |
| Студент |  | | | | А.Ю. Пиров | | | | |

**Аннотация**

Данная курсовая работа посвящена изучению транспортной отрасли по грузоперевозкам. В работе изучается спрос на грузоперевозки, структура и характеристика грузов, способы доставки груза и их преимущества, а также особенности грузоперевозок.

Кроме того, в работе будут изучены способы доставок груза, такие как дверь-терминал, терминал-терминал, терминал дверь, дверь-дверь, их преимущества и недостатки.

В итоге курсовая работа по грузоперевозкам позволит получить более четкое представление о транспортной отрасли и разработать решения, направленные на повышение эффективности грузоперевозок, снижение расходов и увеличение прибыли.

**Содержание**

[Введение 5](#_Toc136788955)

[1 Проектирование базы данных 6](#_Toc136788956)

[1.1 Анализ и описание предметной области информационной системы 6](#_Toc136788957)

[1.2 Проектирование концептуальной модели 7](#_Toc136788958)

[1.3 Построение логической модели, схемы базы данных 8](#_Toc136788959)

[1.4 Выбор СУБД 9](#_Toc136788960)

[2 Реализация базы данных 10](#_Toc136788961)

[2.1 Физическое проектирование 10](#_Toc136788962)

[2.2 Написание исходного кода БД 16](#_Toc136788963)

[2.3 Написание запросов, функций и хранимых процедур 23](#_Toc136788964)

[Заключение 27](#_Toc136788965)

[Список использованных источников 28](#_Toc136788966)

**Введение**

Цель этой курсовой работы - изучить, как использование баз данных в грузоперевозках может увеличить эффективность логистических процессов, сократить время перевозки, уменьшить стоимость транспортировки грузов и увеличить удовлетворенность клиентов.

В работе рассмотрены основные понятия и определения, подходы и модели, которые используются при работе с базами данных в грузоперевозках. Будут исследованы особенности баз данных в транспортной логистике, а также рассмотрены возможности и преимущества использования баз данных при управлении грузоперевозками.

Итак, данная курсовая работа будет полезной для тех, кто интересуется вопросами использования баз данных в грузоперевозках и стремится к повышению эффективности управления логистическими процессами.

# **Проектирование базы данных**

## **Анализ и описание предметной области информационной системы**

Анализ и описание предметной области информационной системы грузоперевозок - важный этап в проектировании такой системы, который позволяет определить функциональные возможности системы и требования к ее разработке и внедрению.

Грузоперевозки - это доставка грузов из одного места в другое, которая может возникнуть как внутри предприятия, так и между различными фирмами. Для оптимизации этого процесса часто используются информационные системы, которые позволяют оперативно контролировать перемещение грузов и управлять логистическими процессами.

В предметной области информационной системы грузоперевозок нередко может возникнуть проблема нехватки информации о перемещении грузов, а также проблема управления качеством услуг, транспортных расходов и сроков доставки. Информационная система может решить эти проблемы, предоставляя оперативную информацию о грузах, транспортных средствах и регулируя логистические процессы по доставке их конечному месту назначения.

Основной целью информационной системы грузоперевозок является обеспечение эффективной и точной доставки грузов, минимизация услуг, связанных с перевозками, рациональное использование транспортных ресурсов, обеспечение контроля над логистическими процессами и повышение качества обслуживания клиентов.

Таким образом, основной задачей информационной системы грузоперевозок является обеспечение эффективного управления логистическими процессами, увеличение производительности и контроля над обслуживанием клиентов, что может повысить конкурентоспособность компании на рынке.

## **Проектирование концептуальной модели**

Проектирование концептуальной модели - это этап создания информационной системы, на котором определяется ее структура, основные функции и связи между компонентами системы. Концептуальная модель является высокоуровневой абстракцией, которая отображает общие идеи и принципы информационной системы.

В процессе проектирования концептуальной модели в системе грузоперевозок необходимо определить основные объекты предметной области, представить их в виде сущностей, а также связи между ними. В качестве основных объектов предметной области могут выступать грузы, заказчики, транспортные средства, конечные адреса доставки и др.

Еще одним важным аспектом при проектировании концептуальной модели информационной системы грузоперевозок являются связи между сущностями. Связи могут быть однозначными или многозначными и могут использоваться для определения отношений между объектами предметной области.

Таким образом, проектирование концептуальной модели для информационной системы грузоперевозок - это ключевой этап, который позволяет определить основные объекты предметной области, их свойства и взаимосвязи. Это позволяет разработчикам создать эффективную и производительную информационную систему.

## **Построение логической модели, схемы базы данных**

Построение логической модели и схемы базы данных - это следующий этап после проектирования концептуальной модели информационной системы грузоперевозок. Он позволяет конкретизировать предметную область и осуществить перевод концептуальной модели в язык конкретной базы данных.

На этом этапе создаются таблицы и связи между ними, определяются поля таблиц и их типы. Для информационной системы грузоперевозок будут созданы таблицы для сущностей, таких как “груз”, “заказчик”, “способ доставки”, “адрес доставки” и другие.

Для каждой таблицы определяются ключи (основной ключ и внешние ключи), которые позволяют связывать таблицы и создавать отношения между ними. Например, таблица “Order” может связываться с таблицей “Method” через внешний ключ “ method\_of\_transportation”.

Также на этом этапе возможно определение процедур (хранимых процедур), которые будут использоваться для выполнения транзакций в базе данных, например, сохранение нового груза в базу данных или изменение статуса доставки.

Для построения схемы базы данных, визуальное представление логической модели, могут использоваться средства проектирования базы данных, такие как ER-диаграммы. Они позволяют визуализировать таблицы, ключи, связи и поля в более понятном виде.

Таким образом, построение логической модели и схемы базы данных в информационной системе грузоперевозок является важным этапом, который позволяет определить структуру базы данных, создать связи и поля для таблиц и обеспечить целостность данных в системе.

## **Выбор СУБД**

Выбор СУБД (системы управления базами данных) - это важный шаг при разработке информационной системы грузоперевозок. При выборе СУБД необходимо учитывать множество факторов, таких как объем и структуру данных, требования к производительности, уровень безопасности и надежности, наличие готовых решений для разработки, доступность технической поддержки и многие другие.

Рассмотрим наиболее популярные СУБД, которые могут быть использованы для разработки системы грузоперевозок:

1. MySQL - это свободная реляционная СУБД, широко используемая для разработки веб-приложений и масштабируемых систем. MySQL обеспечивает высокую производительность и удобный интерфейс для разработчиков.
2. PostgreSQL - это мощная реляционная СУБД с открытым исходным кодом. PostgreSQL поддерживает многие функции, включая триггеры, хранимые процедуры и представления, и предоставляет высокий уровень безопасности и надежности.
3. Microsoft SQL Server - это коммерческая реляционная СУБД, разработанная компанией Microsoft. Она обеспечивает высокую производительность и безопасность, имеет удобный интерфейс управления и интегрируется с другими продуктами Microsoft.
4. Oracle Database - это мощная коммерческая реляционная СУБД, которая поддерживает многие функции, такие как транзакции, многопоточность и хранимые процедуры. Oracle Database имеет высокую производительность и надежность, но требует определенных навыков для настройки и управления.
5. MongoDB - это документо-ориентированная СУБД, которая предоставляет более гибкий подход к хранению и обработке данных. MongoDB поддерживает JSON-документы и обеспечивает высокую производительность, но меньшую надежность и безопасность по сравнению с реляционными СУБД.

В целом, выбор СУБД в системе грузоперевозок зависит от ряда факторов, таких как типы запросов, виды данных, требования к производительности и удобство использования. В любом случае, выбранная СУБД должна обеспечивать высокую надежность и производительность, а также иметь поддержку со стороны разработчиков и сообщества пользователей.

В моем случае была выбрана СУБД PostgreSQL, на которой создана база данных по грузоперевозкам. PostgreSQL (или просто Postgres) - это мощная объектно-реляционная система управления базами данных (СУБД), которая предлагает множество возможностей и функций. Она является одной из самых популярных открытых СУБД и широко используется в различных проектах, от небольших веб-приложений до крупных предприятий.

Ниже я приведу некоторые основные характеристики и функции PostgreSQL, которые могут быть полезными при написании курсовой работы:

1. Открытый исходный код: PostgreSQL является проектом с открытым исходным кодом, что означает, что его исходный код доступен для свободного использования, изучения и модификации. Это позволяет пользователям настраивать и расширять функциональность СУБД под свои потребности.

2. Расширяемость: PostgreSQL обладает мощной системой расширений, которая позволяет добавлять дополнительные функции и возможности. Вы можете создавать пользовательские типы данных, функции, агрегатные функции, операторы, индексы и многое другое.

3. Поддержка SQL: PostgreSQL полностью поддерживает стандарт SQL и предоставляет широкий набор функций и операторов для выполнения различных операций с данными, таких как выборка, вставка, обновление, удаление, агрегирование и т.д.

4. Расширенная поддержка типов данных: PostgreSQL предлагает широкий спектр встроенных типов данных, включая целочисленные типы, числа с плавающей точкой, символьные строки, даты и времена, булевы значения, массивы, JSON, XML и многое другое. Кроме того, вы также можете определять собственные пользовательские типы данных.

5. Транзакционность и целостность данных: PostgreSQL обеспечивает полную поддержку транзакций, что позволяет вам выполнять несколько операций как единое логическое действие. Он также обеспечивает целостность данных с помощью ограничений, триггеров, внешних ключей и других механизмов.

6. Многопользовательская и многопоточная поддержка: PostgreSQL поддерживает одновременную работу нескольких пользователей, предоставляя им доступ к базе данных с различными уровнями привилегий. Он также обеспечив

# **Реализация базы данных**

## **Физическое проектирование**

В базе данных грузоперевозок существуют следующие таблицы:

1. Client (клиент) - в данной таблице хранится информация о заказчике (рисунок 1):

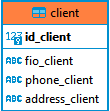


Рисунок 1 – Таблица Client

1. Sender (отправитель) - в данной таблице хранится информация об отравителе (рисунок 2):

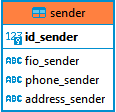


Рисунок 2 – Таблица Sender

1. Recipient (получатель) - в данной таблице хранится информация о получателе (рисунок 3):

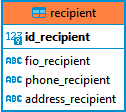


Рисунок 3 – Таблица Recipient

1. Persons (лица), в которой созданы внешние ключи, связанные таблицами Client, Sender, Recipient и Order (рисунок 4):

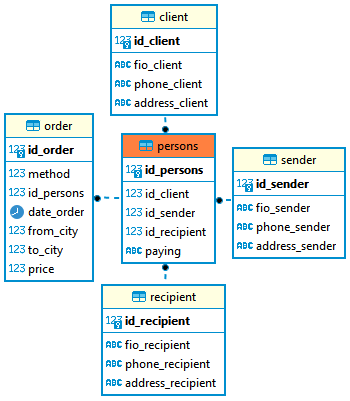


Рисунок 4 – Таблица Persons

1. Method (способ доставки) – таблица-справочник (рисунок 5), заполненная четырьмя способами доставки (рисунок 6):

* доставка от двери до терминала (Door-Terminal) - это способ доставки, при котором груз забирается у отправителя из двери его дома (или офиса) и доставляется в терминал назначения (например, склад перевозчика), где получатель самостоятельно получает свой груз и отвозит его;
* доставка от терминала до терминала (Terminal-Terminal) - это способ доставки, при котором груз отправляется из терминала отправителя в терминал назначения, где он хранится до получения получателем. При этом отправитель и получатель должны самостоятельно забирать и отвозить свой груз;
* доставка от терминала до двери (Terminal-Door) - это способ доставки, при котором груз отправляется из терминала отправителя и доставляется к двери получателя. При этом отправитель самостоятельно доставляет груз в терминал, а получатель получает груз у себя дома или в офисе;
* доставка от двери до двери (Door-Door) - это способ доставки, при котором груз забирается у отправителя с его двери и доставляется непосредственно к двери получателя. В этом случае отправитель и получатель не должны забирать и доставлять свой груз самостоятельно.



Рисунок 5 – Таблица Method

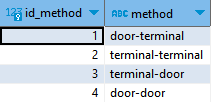


Рисунок 6 – Данные в таблице Method

1. Status (статус заказа) – таблица-справочник (рисунок 7), заполненная статусами заказа (рисунок 8):

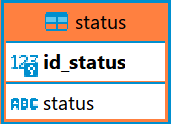


Рисунок 7 – Таблица Status

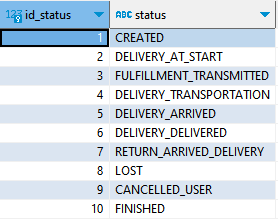


Рисунок 8 – Данные в таблице Status

1. Journal (журнал) – таблица для ведения отчетов о заказах, связанная с таблицей Order и Status (рисунок 9):

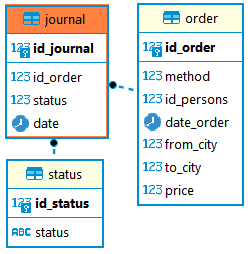


Рисунок 9 – Таблица Journal

1. Character (категория товара) – таблица-справочник (рисунок 10), в которой описаны основные категории товаров (рисунок 11):



Рисунок 10 – Таблица Character



Рисунок 11 – Данные в таблице Character

1. Order\_Details (детали заказа) – таблица, связанная с таблицей Character и Order, в которой хранятся интересующие для компании грузоперевозок детали о заказе (рисунок 12):

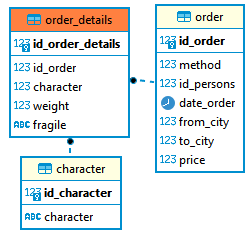


Рисунок 12 – Таблица Order\_Details

1. City (город) - таблица-справочник (рисунок 13), в которой расписаны города для информации о заказе (рисунок 14):



Рисунок 13 – Таблица City



Рисунок 14 – Данные в таблице City

1. Order (заказ) – основная таблица, в которой хранится вся информация о заказе, методе транспортировки (таблиц Method), информация о клиентах (таблица Persons), дате заказа, информация о городе отправки и получения (таблица City) и цена заказа (рисунок 15):

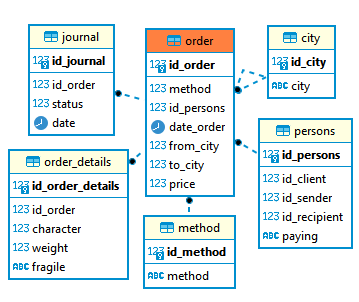


Рисунок 15 – Таблица Order

## **Написание исходного кода БД**

Для создания таблиц был написан код, представленный ниже.

Код для таблицы Client (рисунок 16):

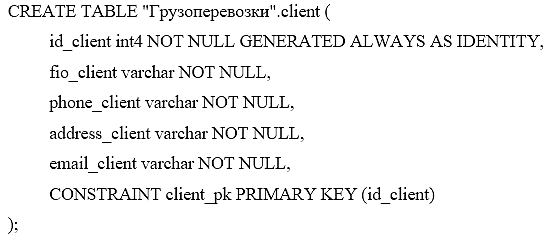


Рисунок 16 – Код для таблицы Client

Код для таблицы Sender (рисунок 17):

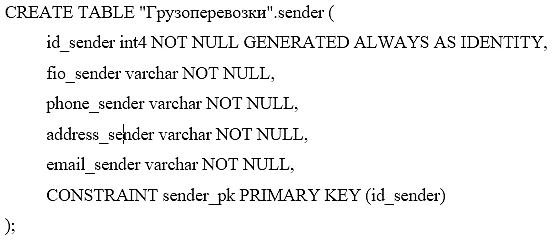


Рисунок 17 – Код для таблицы Sender

Код для таблицы Recipient (рисунок 18):

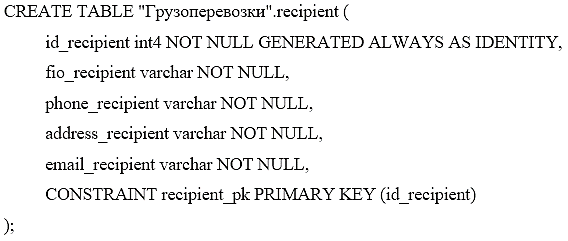


Рисунок 18 – Код для таблицы Recipient

Код для таблицы Method (рисунок 19):

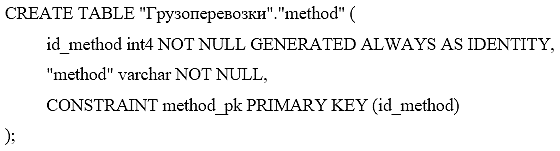


Рисунок 19 – Код для таблицы Method

Код для таблицы Order\_Details (рисунок 20):

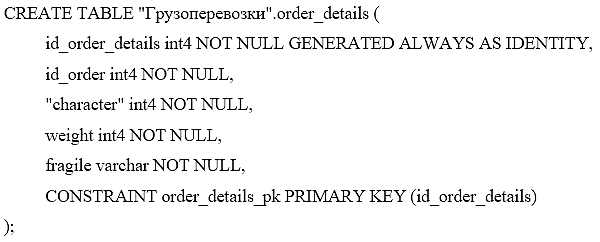


Рисунок 20 – Код для таблицы Order\_Details

Код для таблицы Journal (рисунок 21):

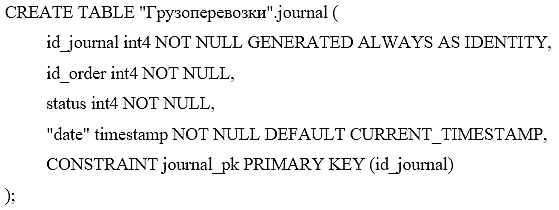


Рисунок 21 – Код для таблицы Journal

Код для таблицы Character (рисунок 22):

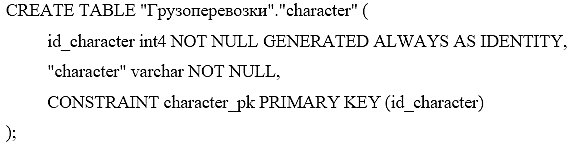


Рисунок 22 – Код для таблицы Character

Код для таблицы Status (рисунок 23):

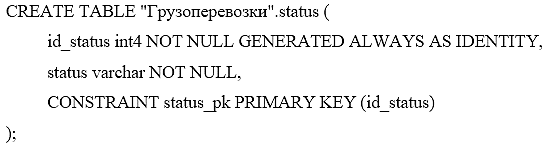


Рисунок 23 – Код для таблицы Status

Код для таблицы City (рисунок 24):

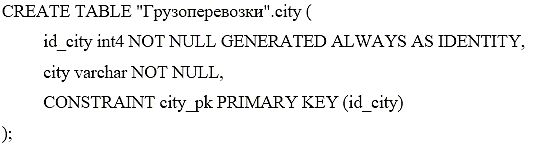


Рисунок 24 – Код для таблицы City

Код для таблицы Order (рисунок 25):

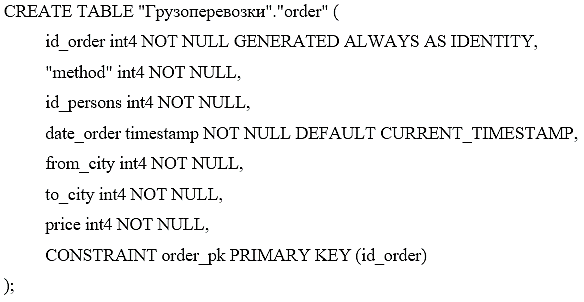


Рисунок 25 – Код для таблицы Order

Код для таблицы Persons (рисунок 26):

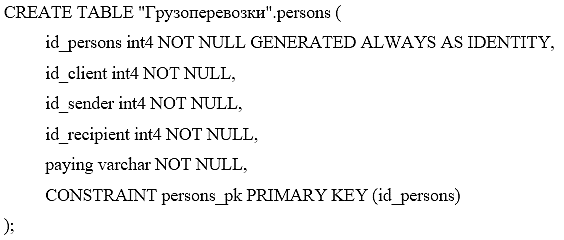


Рисунок 26 – Код для таблицы Persons

Для таблиц-справочников был написан код заполнения данными, представленный ниже.

Код для заполнения таблицы Status (рисунок 27):

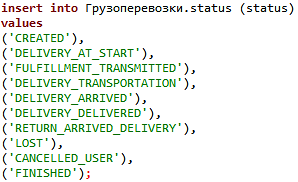


Рисунок 27 – Код для заполнения таблицы Status

Код для заполнения таблицы City (рисунок 28):

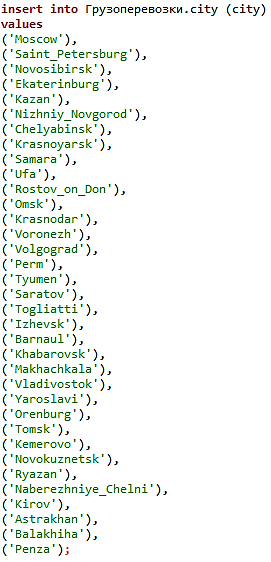


Рисунок 28 – Код для заполнения таблицы City

Код для заполнения таблицы Character (рисунок 29):

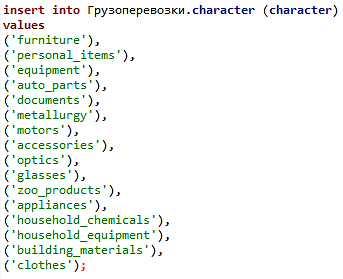


Рисунок 29 – Код для заполнения таблицы Character

Код для заполнения таблицы Method (рисунок 30):

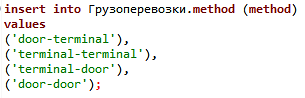


Рисунок 30 – Код для заполнения таблицы Method

Также для определенных таблиц был расписан код для связки внешних ключей с первичными ключами.

Внешний ключ character таблицы Order\_Details, связывающий первичный ключ из таблицы Character (рисунок 31):



Рисунок 31 – Внешний ключ character

* Внешний ключ id\_order таблицы Order\_Details, связывающий первичный ключ из таблицы Order (рисунок 32):



Рисунок 32 – Внешний ключ id\_order

* Внешний ключ id\_order таблицы Journal, связывающий первичный ключ из таблицы Order (рисунок 33):



Рисунок 33 – Внешний ключ id\_order

* Внешний ключ status таблицы Journal, связывающий первичный ключ из таблицы Status (рисунок 34):



Рисунок 34 – Внешний ключ status

* Внешний ключ from\_ctiy таблицы Order, связывающий первичный ключ из таблицы City (рисунок 35):



Рисунок 35 – Внешний ключ from\_ctiy

* Внешний ключ to\_ctiy таблицы Order, связывающий первичный ключ из таблицы City (рисунок 36):



Рисунок 36 – Внешний ключ to\_ctiy

* Внешний ключ method таблицы Order, связывающий первичный ключ из таблицы Method (рисунок 37):



Рисунок 37 – Внешний ключ method

Внешний ключ id\_persons таблицы Order, связывающий первичный ключ из таблицы Persons (рисунок 38):



Рисунок 38 – Внешний ключ id\_persons

Внешний ключ id\_client таблицы Persons, связывающий первичный ключ из таблицы Client (рисунок 39):



Рисунок 39 – Внешний ключ id\_client

Внешний ключ id\_sender таблицы Persons, связывающий первичный ключ из таблицы Sender (рисунок 40):



Рисунок 40 – Внешний ключ id\_sender

Внешний ключ id\_recipient таблицы Persons, связывающий первичный ключ из таблицы Recipient (рисунок 41):



Рисунок 41 – Внешний ключ id\_recipient

## **Написание запросов, функций и хранимых процедур**

Для работы с базой данных были написаны следующие запросы:

1. «SELECT \* FROM "order" o WHERE o.id\_order IN (SELECT id\_order FROM journal j WHERE j.status = 10)»

Этот запрос выбирает информацию с таблицы «order», где значение «id\_order» содержится в результате подзапроса. Подзапрос выбирает все «id\_order» из таблицы «journal», где значение столбца «status» равно значению 10, что означает статус заказа “FINISHED”, то есть «Завершено». В итоге было выведено 285 записей о заказах (рисунок 42).

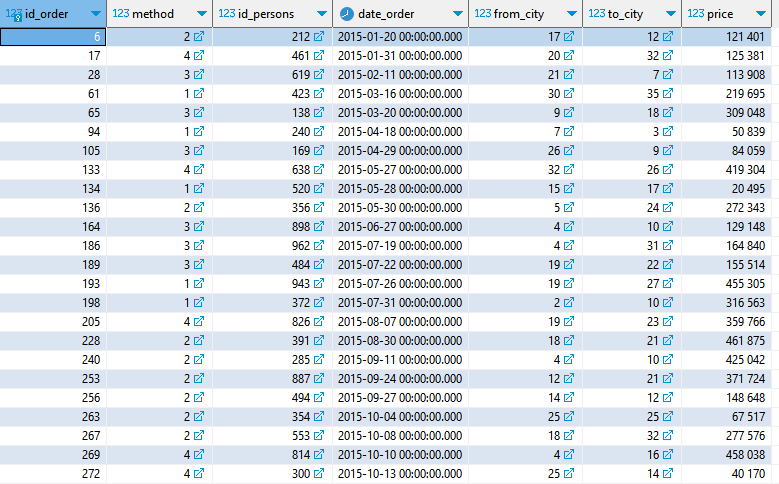


Рисунок 42 – Резульат запроса по заказам

Таким образом, основной запрос возвращает все строки из таблицы “order”, которые имеют одинаковые значения “id\_order”, что и в подзапросе. Использование оператора IN вместо = позволяет использовать подзапрос, который может возвращать несколько строк, без возникновения ошибки “more than one row returned by a subquery used as an expression”.

Результат запроса может быть использован для анализа тенденций выполнения заказов и определения времени, затраченного на исполнение заказов в системе.

1. «SELECT c.character, COUNT(od.id\_character) as count

FROM order\_details od JOIN character c ON c.id\_character = od.id\_character

GROUP BY c.character ORDER BY count DESC;»

Этот запрос позволяет получить наиболее часто встречающиеся категории из таблицы “character”, связанной с таблицей “order\_details” через внешний ключ “id\_character”.

Оператор JOIN используется для объединения двух таблиц по столбцу “id\_character”. Затем результат группируется по наименованию каждой категории “character” с помощью функции GROUP BY.

Функция COUNT() подсчитывает количество записей в таблице “order\_details” для каждой категории в “character”. Результаты (рисунок 43) сортируются в порядке убывания количества записей с помощью оператора ORDER BY с дескриптором DESC.

Этот запрос может использоваться в качестве инструмента для анализа заказов в системе и определения топ-категорий, которые наиболее часто доставляются.

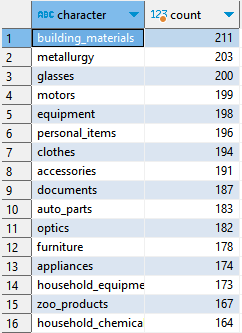


Рисунок 43 – Резульат запроса по категориям

1. «SELECT m.method, COUNT(o.method) as count

FROM "order" o JOIN method m ON m.id\_method = o.method

GROUP BY m.method ORDER BY count DESC»

Этот запрос позволяет получить наиболее часто используемые способы доставки из таблицы “method”, связанной с таблицей “order” через внешний ключ “method”.

Оператор JOIN используется для объединения двух таблиц по столбцу “id\_method”. Затем результат группируется по названию каждого способа доставки “method” с помощью функции GROUP BY.

Функция COUNT() подсчитывает количество записей в таблице “order” для каждого способа доставки в “method”. Результаты сортируются в порядке убывания количества записей с помощью оператора ORDER BY с дескриптором DESC.

Этот запрос может использоваться в качестве инструмента для анализа заказов в системе и определения топ-способов доставки, которые наиболее часто используются. Результаты (рисунок 44) запроса могут помочь руководству компании принимать решения об улучшении системы доставки, выборе наиболее эффективных способов доставки товаров.

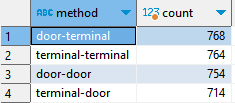


Рисунок 44 – Резульат запроса по способам траспортировки

1. SELECT DATE\_TRUNC('year', date\_order) as year, COUNT(\*) as count

FROM "order" GROUP BY year ORDER BY year ASC;

Этот запрос позволяет получить количество заказов за каждый год из столбца “date\_order” таблицы “order”.

В запросе используется функция DATE\_TRUNC(), которая округляет дату до начала года, тем самым группируя заказы по годам. Затем, используется функция COUNT() для подсчета количества заказов за каждый год.

Результаты объединяются по году и сортируются в порядке возрастания года с помощью оператора ORDER BY.

Такой запрос может быть полезен для анализа динамики заказов в течение нескольких лет. Результаты (рисунок 45) могут помочь руководству компании принимать решения в зависимости от количества заказов каждый год, что может помочь определить популярные периоды заказов и построить стратегию развития компании с учетом этих данных.

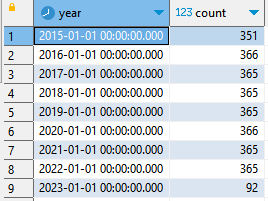


Рисунок 45 – Резульат запроса заказов за каждый год

**Заключение**

В курсовой работе по Базам Данных было выполнено создание базы данных грузоперевозок, которая была успешно спроектирована и отвечает всем требованиям функциональности и структуры. Проект включает в себя несколько таблиц, которые описывают важные аспекты операции грузоперевозок, такие как заказы, инвентарь, перевозки и другие.

В результате работы были обработаны и загружены данные, включая написание запросов и использование SQL-команд для заполнения таблиц. Настройка и использование запросов помогло улучшить производительность базы данных, ускорить обработку запросов и сделать данные более полезными и доступными.

Результатом курсовой работы является установка Базы Данных Грузоперевозок, которая отвечает всем стандартам и требованиям операции грузоперевозок. База данных представляет собой мощный инструмент для управления данными, оптимизации операций и улучшения качества услуг грузоперевозок.

В целом, выполнение данной курсовой работы помогло углубить знания в области баз данных и изучить различные методы работы с данными, такие как SQL-запросы, в практическом применении. Также были изучены основы проектирования баз данных, в том числе создание таблиц и взаимосвязей для управления и обработки данных.

**Список использованных источников**

[PostgreSQL : Документация: 15: Глава 7. Запросы : Компания Postgres Professional](https://postgrespro.ru/docs/postgresql/15/queries)

[PostgreSQL: Documentation: 15: 2.3. Creating a New Table](https://www.postgresql.org/docs/current/tutorial-table.html)

[PostgreSQL : Документация: 15: 6.1. Добавление данных : Компания Postgres Professional](https://postgrespro.ru/docs/postgresql/15/dml-insert)