МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ НИЖЕГОРОДСКОЙ ОБЛАСТИ

Государственное бюджетное профессиональное образовательное учреждение НИЖЕГОРОДСКИЙ РАДИОТЕХНИЧЕСКИЙ КОЛЛЕДЖ

Специальность 09.02.07 Информационные системы и программирование Квалификация Программист

# КУРСОВАЯ РАБОТА

МДК 11.01 Технология разработки и защиты баз данных

Тема: «Проектирование и разработка базы данных для магазина компьютерной техники»

|  |  |
| --- | --- |
| Выполнил студент группы  3ИСиП-22-2 | Проверил преподаватель Гутянская Е.М. |
| Пронин Дмитрий Алексеевич | Проект защищен с оценкой  Дата защиты Подпись |

Нижний Новгород, 2024

# Оглавление

[КУРСОВАЯ РАБОТА 1](#_Toc1)

[Оглавление 2](#_Toc2)

[Введение 3](#_Toc3)

[Задачи работы: 3](#_Toc4)

[Ожидаемый результат работы: 3](#_Toc5)

# **Введение**

В современном мире компьютерная техника занимает важное место в жизни пользователей и организаций. С каждым годом увеличивается спрос на компьютеры, что ставит перед магазинами задачу эффективного управления товарными запасами, учётом продуктов и обслуживанием клиентов. Ключевым элементом для достижения этих целей является создание и функционирование базы данных, специально разработанной для магазина компьютерной техники. Целью данной курсовой работы является проектирование и разработка базы данных и информационной системы для магазина компьютерной техники. Это позволит оптимизировать управление складскими операциями, автоматизировать процессы учёта товаров и обслуживания клиентов, а также повысить общую эффективность работы магазина.

## Задачи работы:

1. Анализ требований и потребностей магазина компьютерной техники.
2. Проектирование структуры базы данных с учётом специфики магазина.
3. Разработка функциональной части информационной системы для автоматизации бизнес-процессов.
4. Реализация и интеграция базы данных и информационной системы.
5. Оценка эффективности использования базы данных и информационной системы в работе магазина.

## Ожидаемый результат работы:

Готовая функционирующая база данных и информационная система для магазина компьютерной техники.В процессе выполнения данной курсовой работы будут применены современные методы и инструменты проектирования баз данных и информационных систем, что позволит создать удобную, надёжную и эффективную систему управления информацией для магазина компьютерной техники. Исследование лучших практик в области проектирования баз данных поможет учесть все необходимые функции и требования, обеспечивая оптимальное использование ресурсов и повышая конкурентоспособность магазина на рынке.

# Теоретические основы разрабатываемой темы

## Анализ проектируемой системы

В рамках разработанной системы для магазина компьютеров будут реализованы следующие функции:

* Многопользовательский доступ
* Разграничение функционала по ролям
* Клиент и сотрудник магазина могут осуществлять следующие действия:
  + получать оперативную информацию о наличии, описании, фото и стоимости техники;
  + подбирать технику по заявленным критериям;
  + осуществлять отбор техники по категориям, производителям;
  + находить необходимую информацию, содержащую описание техники и комплектующих;
* Сотрудник магазина может осуществлять следующие действия:
  + вести справочники (добавление, удаление, редактирование);
  + оформлять заказ на товар (в одном заказе может быть несколько различных товаров в разном количестве);
  + стоимость заказа рассчитывается динамически.

Данные в системе будут храниться в структурированном формате в базе данных:

* Пользователи. В этой таблице хранится информация о пользователях системы, включая их имя пользователя, адрес электронной почты, имя, фамилию, пароль и роль.
* Процессоры. В этой таблице хранится информация о процессорах, включая их название, бренд, количество ядер и потоков, базовую и турбочастоты.
* Видеокарты. В этой таблице хранится информация о графических картах, включая название, бренд, объем памяти и тип памяти.
* Блоки питания. В этой таблице хранится информация о блоках питания, включая название, бренд, мощность и рейтинг эффективности.
* Материнские платы. В этой таблице хранится информация о материнских платах, включая название, бренд, тип сокета и форм-фактор.
* Охладители. В этой таблице хранится информация об охладителях, включая название, бренд, тип и мощность охлаждения.
* Корпуса. В этой таблице хранится информация о корпусах компьютеров, включая название, бренд, форм-фактор и цвет.
* Оперативная память (RAM). В этой таблице хранится информация об оперативной памяти, включая название, бренд, объем и скорость.
* Компьютеры. В этой таблице хранится информация обо всех компьютерах, доступных для продажи, включая описание, цену и ссылки на компоненты (процессор, видеокарта и т.д.).
* Корзина покупок. В этой таблице хранится информация о товарах в корзине пользователей, включая владельца корзины и количество каждого компьютера.
* Заказы. В этой таблице хранится информация о заказах клиентов, включая дату заказа, общую сумму и статус заказа.
* Элементы заказа. В этой таблице хранится информация о конкретных товарах в каждом заказе, включая количество и цену каждого товара.

Рассмотрим функции, которые будут реализованы в нашем приложении:

* Многопользовательский доступ:

Это возможность одновременного доступа нескольких пользователей к сервису.

* Разграничение функционала по ролям:  
  Это метод управления доступом, при котором права доступа к различным функциям системы предоставляются на основе ролей, которые назначаются пользователям. Каждая роль определяет свой набор прав доступа. Этот подход позволяет эффективно управлять правами доступа и обеспечивает безопасность информации в системе.

В нашей программе сотрудник магазина будет иметь доступ ко всем функциям и данным, в то время как клиенты будут иметь ограниченный доступ к определенным функциям или данным.

Функции, доступные только для сотрудника магазина:

* + Получение любой информации о товаре;
  + Ведение справочников (добавление, удаление, редактирование);
  + Оформление заказа за пользователя (доступно поле комментария при оформлении заказа, в которое можно указать контактную информацию о заказчике);

Функции, доступные для всех:

* + Получение информации о названии, описании, стоимости компьютера;
  + Получение информации о комплектующих, установленных в ПК;
  + Осуществлять фильтрацию продуктов в каталоге;
  + Составление своей корзины покупок;
  + Просмотр информации о всех своих заказах

# Проектирование базы данных информационной системы

## Информационно-логическая модель базы данных

Информационно-логическая модель отображает данные предметной области в виде совокупности информационных объектов и связей между ними. Эта модель представляет данные, подлежащие хранению в базе данных.

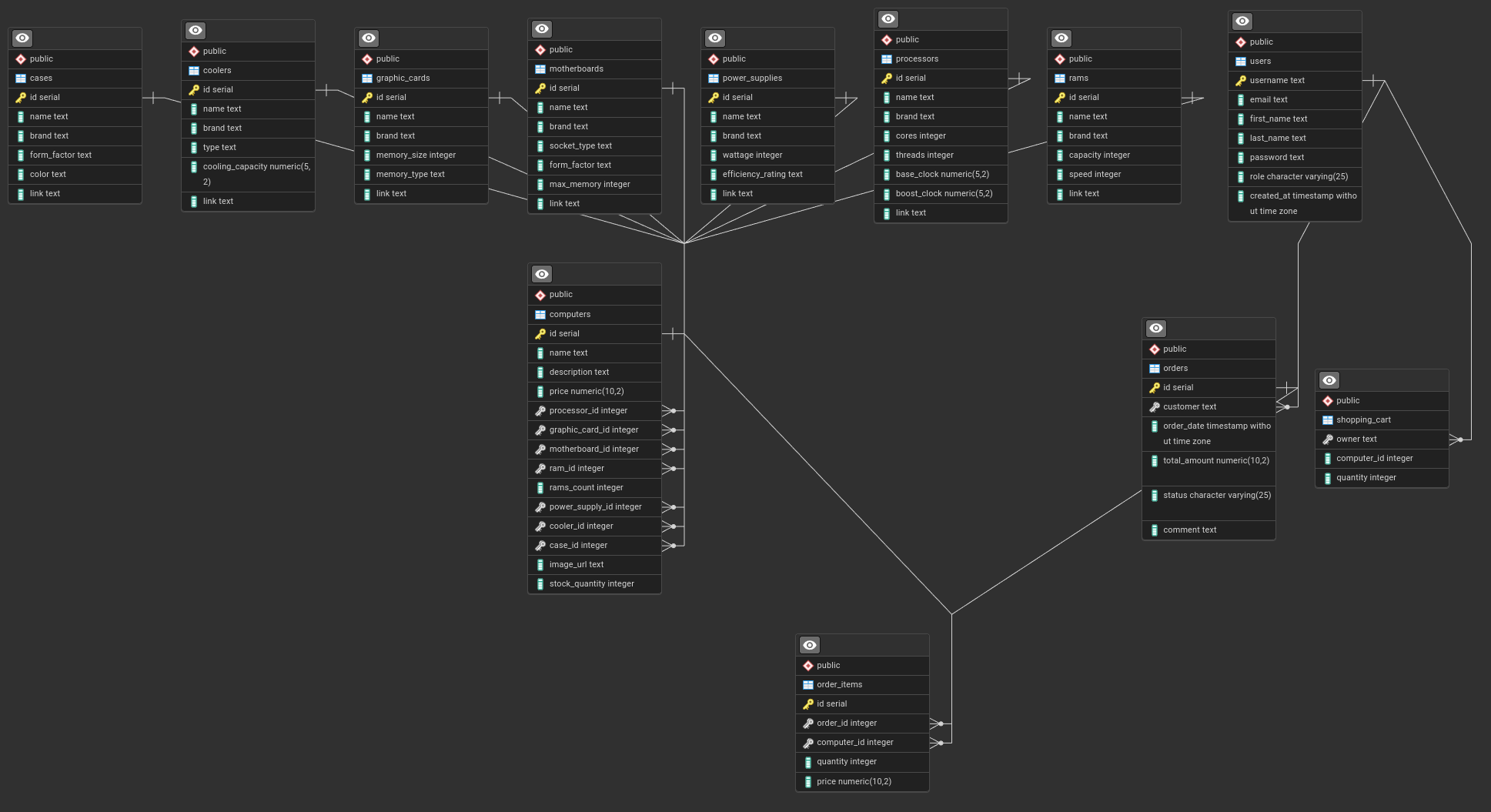


Рисунок 1. Информационно-логическая модель базы данных

## Словарь данных

В приведённых ниже таблицах описаны поля всех таблиц базы данных, используемых в разработанной информационной системе.

| **users** | | | |
| --- | --- | --- | --- |
| **Имя атрибута** | **Значение атрибута** | **Тип данных** | **Ключ (ссылка)** |
| username | Имя пользователя | TEXT | PK |
| email | Электронная почта | TEXT |  |
| first\_name | Имя | TEXT |  |
| last\_name | Фамилия | TEXT |  |
| password | Пароль | TEXT |  |
| role | Роль | VARCHAR(25) |  |
| created\_at | Дата создания | TIMESTAMP |  |

| **processors** | | | |
| --- | --- | --- | --- |
| **Имя атрибута** | **Значение атрибута** | **Тип данных** | **Ключ (ссылка)** |
| id | Идентификатор | SERIAL | PK |
| name | Название | TEXT |  |
| brand | Бренд | TEXT |  |
| cores | Количество ядер | INT |  |
| threads | Количество потоков | INT |  |
| base\_clock | Базовая частота | DECIMAL(5,2) |  |
| boost\_clock | Турбочастота | DECIMAL(5,2) |  |
| link | Ссылка | TEXT |  |

| **graphic\_cards** | | | |
| --- | --- | --- | --- |
| **Имя атрибута** | **Значение атрибута** | **Тип данных** | **Ключ (ссылка)** |
| id | Идентификатор | SERIAL | PK |
| name | Название | TEXT |  |
| brand | Бренд | TEXT |  |
| memory\_size | Объем памяти | INT |  |
| memory\_type | Тип памяти | TEXT |  |
| link | Ссылка | TEXT |  |

| **power\_supplies** | | | |
| --- | --- | --- | --- |
| **Имя атрибута** | **Значение атрибута** | **Тип данных** | **Ключ (ссылка)** |
| id | Идентификатор | SERIAL | PK |
| name | Название | TEXT |  |
| brand | Бренд | TEXT |  |
| wattage | Мощность | INT |  |
| efficiency\_rating | Рейтинг эффективности | TEXT |  |
| link | Ссылка | TEXT |  |

|  |  |  |  |
| --- | --- | --- | --- |
| **Имя атрибута** | **Значение атрибута** | **Тип данных** | **Ключ (ссылка)** |
| id | Идентификатор | SERIAL | PK |
| name | Название | TEXT |  |
| brand | Бренд | TEXT |  |
| socket\_type | Тип сокета | TEXT |  |
| form\_factor | Форм-фактор | TEXT |  |
| max\_memory | Максимальный объем памяти | INT |  |
| link | Ссылка | TEXT |  |

| **coolers** | | | |
| --- | --- | --- | --- |
| **Имя атрибута** | **Значение атрибута** | **Тип данных** | **Ключ (ссылка)** |
| id | Идентификатор | SERIAL | PK |
| name | Название | TEXT |  |
| brand | Бренд | TEXT |  |
| type | Тип охладителя | TEXT |  |
| cooling\_capacity | Мощность охлаждения | DECIMAL(5,2) |  |
| link | Ссылка | TEXT |  |

| **cases** | | | |
| --- | --- | --- | --- |
| **Имя атрибута** | **Значение атрибута** | **Тип данных** | **Ключ (ссылка)** |
| id | Идентификатор | SERIAL | PK |
| name | Название | TEXT |  |
| brand | Бренд | TEXT |  |
| form\_factor | Форм-фактор | TEXT |  |
| color | Цвет | TEXT |  |
| link | Ссылка | TEXT |  |

| **rams** | | | |
| --- | --- | --- | --- |
| **Имя атрибута** | **Значение атрибута** | **Тип данных** | **Ключ (ссылка)** |
| id | Идентификатор | SERIAL | PK |
| name | Название | TEXT |  |
| brand | Бренд | TEXT |  |
| capacity | Объем памяти в ГБ | INT |  |
| speed | Скорость в МГц | INT |  |
| link | Ссылка | TEXT |  |

| **computers** | | | |
| --- | --- | --- | --- |
| **Имя атрибута** | **Значение атрибута** | **Тип данных** | **Ключ (ссылка)** |
| id | Идентификатор | SERIAL | PK |
| name | Название | TEXT |  |
| description | Описание | TEXT |  |
| price | Цена | DECIMAL(10,2) |  |
| processor\_id | Идентификатор процессора | INT NOT NULL REFERENCES processors(id) | FK |
| graphic\_card\_id | Идентификатор видеокарты | INT NOT NULL REFERENCES graphic\_cards(id) | FK |
| motherboard\_id | Идентификатор материнской платы | INT NOT NULL REFERENCES motherboards(id) | FK |
| ram\_id | Идентификатор оперативной памяти | INT NOT NULL REFERENCES rams(id) | FK |
| rams\_count | Количество модулей оперативной памяти | INT NOT NULL |  |
| power\_supply\_id | Идентификатор блока питания | INT NOT NULL REFERENCES power\_supplies(id) | FK |
| cooler\_id | Идентификатор охладителя | INT NOT NULL REFERENCES coolers(id) | FK |
| case\_id | Идентификатор корпуса | INT NOT NULL REFERENCES cases(id) | FK |
| image\_url | URL изображения | TEXT |  |
| stock\_quantity | Количество на складе | INT NOT NULL DEFAULT 0 |  |

| **shopping\_cart** | | | |
| --- | --- | --- | --- |
| **Имя атрибута** | **Значение атрибута** | **Тип данных** | **Ключ (ссылка)** |
| owner | Владелец корзины (username) | TEXT NOT NULL REFERENCES users(username) | FK |
| computer\_id | Идентификатор компьютера | INT NOT NULL REFERENCES computers(id) | FK |
| quantity | Количество товара в корзине | INT NOT NULL DEFAULT 1 |  |

| **orders** | | | |
| --- | --- | --- | --- |
| **Имя атрибута** | **Значение атрибута** | **Тип данных** | **Ключ (ссылка)** |
| id | Идентификатор заказа | SERIAL | PK |
| customer | Клиент (username) | TEXT NOT NULL REFERENCES users(username) | FK |
| order\_date | Дата заказа | TIMESTAMP NOT NULL DEFAULT CURRENT\_TIMESTAMP |  |
| total\_amount | Общая сумма заказа | DECIMAL(10,2) NOT NULL |  |
| status | Статус заказа | VARCHAR(25) NOT NULL DEFAULT 'pending' |  |
| comment | Комментарий к заказу | TEXT |  |

| **order\_items** | | | |
| --- | --- | --- | --- |
| **Имя атрибута** | **Значение атрибута** | **Тип данных** | **Ключ (ссылка)** |
| id | Идентификатор элемента заказа | SERIAL | PK |
| order\_id | Идентификатор заказа | INT NOT NULL REFERENCES orders(id) | FK |
| computer\_id | Идентификатор компьютера | INT NOT NULL REFERENCES computers(id) | FK |
| quantity | Количество товара в заказе | INT NOT NULL |  |
| price | Цена товара на момент заказа | DECIMAL(10,2) NOT NULL |  |

## **Ограничения ссылочной целостности**

1. **Ограничение на таблице «users»**: Идентификатор пользователя (username) является первичным ключом. Это ограничение гарантирует, что каждый пользователь будет иметь уникальное имя пользователя, что позволяет избежать дублирования записей и обеспечивает целостность данных в системе.
2. **Ограничение на таблице «processors»**: Идентификатор процессора (id) является первичным ключом. Это ограничение обеспечивает уникальность каждой записи о процессоре и позволяет точно идентифицировать каждый процессор в базе данных.
3. **Ограничение на таблице «graphic\_cards»**: Идентификатор видеокарты (id) является первичным ключом. Это ограничение гарантирует, что каждая видеокарта будет иметь уникальный идентификатор, что позволяет точно отслеживать информацию о каждой видеокарте.
4. **Ограничение на таблице «power\_supplies»**: Идентификатор блока питания (id) является первичным ключом. Это ограничение обеспечивает уникальность каждой записи о блоке питания и позволяет избежать путаницы между различными моделями блоков питания.
5. **Ограничение на таблице «motherboards»**: Идентификатор материнской платы (id) является первичным ключом. Это ограничение гарантирует, что каждая материнская плата будет иметь уникальный идентификатор, что позволяет точно отслеживать информацию о каждой материнской плате.
6. **Ограничение на таблице «coolers»**: Идентификатор охладителя (id) является первичным ключом. Это ограничение обеспечивает уникальность каждой записи об охладителе и позволяет точно идентифицировать каждый охладитель в базе данных.
7. **Ограничение на таблице «cases»**: Идентификатор корпуса (id) является первичным ключом. Это ограничение гарантирует, что каждый корпус будет иметь уникальный идентификатор, что позволяет избежать дублирования записей и обеспечивает целостность данных.
8. **Ограничение на таблице «rams»**: Идентификатор оперативной памяти (id) является первичным ключом. Это ограничение обеспечивает уникальность каждой записи об оперативной памяти и позволяет точно отслеживать информацию о каждом модуле RAM.
9. **Ограничение на таблице «computers»**:

* Идентификатор процессора (processor\_id) является внешним ключом, связанным с таблицей «processors» по полю идентификатора (id). Это ограничение гарантирует, что каждый компьютер будет иметь действительный процессор, представленный в таблице «processors». Такая связь помогает контролировать совместимость компонентов.
* Идентификатор видеокарты (graphic\_card\_id) является внешним ключом, связанным с таблицей «graphic\_cards» по полю идентификатора (id). Это ограничение гарантирует, что каждый компьютер будет иметь действительную видеокарту, представленную в таблице «graphic\_cards».
* Идентификатор материнской платы (motherboard\_id) является внешним ключом, связанным с таблицей «motherboards» по полю идентификатора (id). Это ограничение гарантирует, что каждый компьютер будет иметь действительную материнскую плату.
* Идентификатор оперативной памяти (ram\_id) является внешним ключом, связанным с таблицей «rams» по полю идентификатора (id). Это ограничение гарантирует, что каждый компьютер будет иметь действительную оперативную память.
* Идентификатор блока питания (power\_supply\_id) является внешним ключом, связанным с таблицей «power\_supplies» по полю идентификатора (id). Это ограничение гарантирует, что каждый компьютер будет иметь действительный блок питания.
* Идентификатор охладителя (cooler\_id) является внешним ключом, связанным с таблицей «coolers» по полю идентификатора (id). Это ограничение гарантирует, что каждый компьютер будет иметь действительное охлаждение.
* Идентификатор корпуса (case\_id) является внешним ключом, связанным с таблицей «cases» по полю идентификатора (id). Это ограничение гарантирует, что каждый компьютер будет иметь действительный корпус.

1. **Ограничение на таблице «shopping\_cart»**: Идентификатор владельца корзины (owner) является внешним ключом, связанным с таблицей «users» по полю идентификатора пользователя (username). Это ограничение гарантирует, что каждая корзина принадлежит действительному пользователю. Идентификатор компьютера (computer\_id) является внешним ключом, связанным с таблицей «computers» по полю идентификатора (id). Это ограничение гарантирует, что каждая запись в корзине содержит действительный компьютер.

1. **Ограничение на таблице «orders»**: Идентификатор клиента (customer) является внешним ключом, связанным с таблицей «users» по полю идентификатора пользователя (username). Это ограничение гарантирует, что каждый заказ принадлежит действительному клиенту.
2. **Ограничение на таблице «order\_items»**: Идентификатор заказа (order\_id) является внешним ключом, связанным с таблицей «orders» по полю идентификатора заказа (id). Это ограничение гарантирует, что каждый элемент заказа относится к действительному заказу. Идентификатор компьютера (computer\_id) является внешним ключом, связанным с таблицей «computers» по полю идентификатора компьютера (id). Это ограничение гарантирует, что каждый элемент заказа содержит действительный компьютер.

## **Обоснование выбора СУБД**

Система управления базами данных (СУБД) – это набор программ, которые управляют структурой БД и контролируют доступ к данным, хранящимся в ней. СУБД служит посредником между пользователем и БД. В качестве СУБД для моей информационной системы я выбрал PostgreSQL. Это решение обусловлено рядом преимуществ, которые PostgreSQL предлагает по сравнению с другими СУБД:

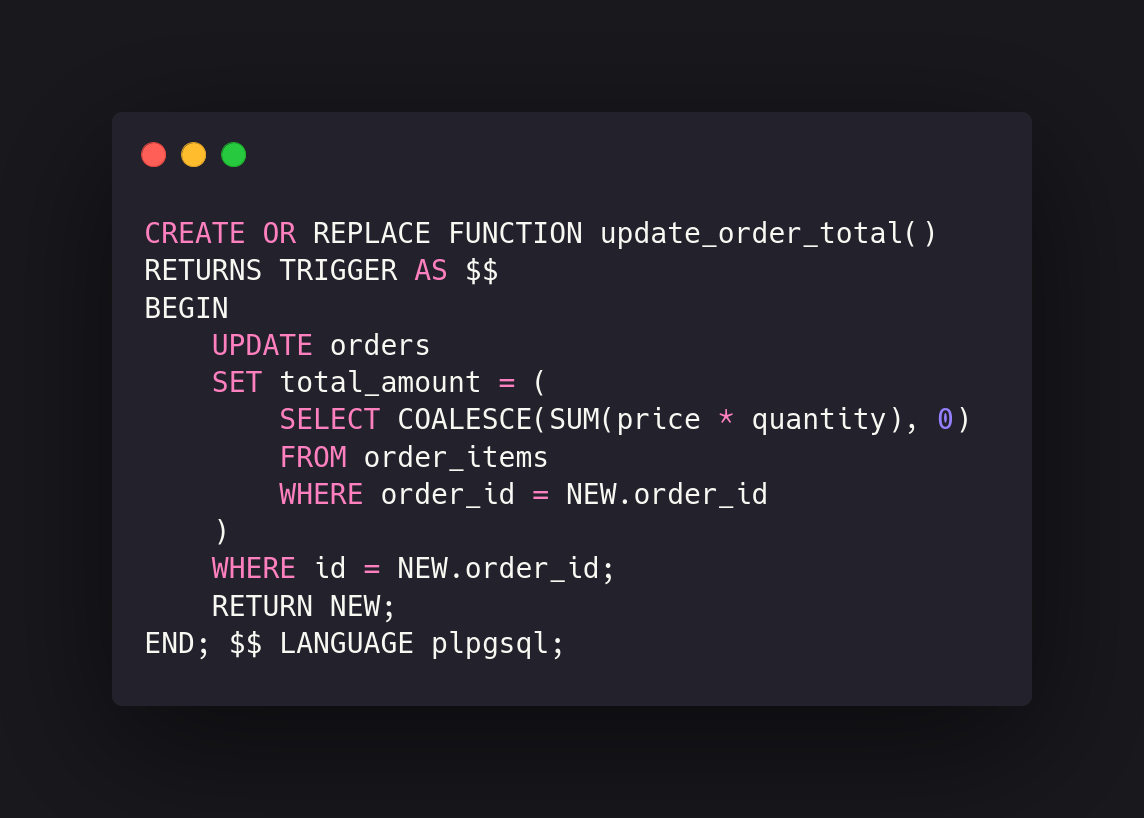
1. **Расширяемость**: PostgreSQL поддерживает пользовательские типы данных, функции и операторы, что позволяет адаптировать систему под специфические требования приложения. Это делает PostgreSQL особенно подходящим для сложных и нестандартных проектов.
2. **Поддержка стандартов SQL**: PostgreSQL строго следует стандартам SQL, что обеспечивает совместимость с другими системами и упрощает миграцию данных. Это также облегчает обучение пользователей, знакомых с SQL.
3. **Высокая производительность**: PostgreSQL использует эффективные механизмы индексации и оптимизации запросов, что обеспечивает высокую производительность при работе с большими объемами данных и сложными запросами.
4. **Надежность и целостность данных**: PostgreSQL поддерживает ACID-транзакции (атомарность, согласованность, изолированность, долговечность), что гарантирует надежность и целостность данных. Это особенно важно для критически важных приложений.
5. **Многоуровневая безопасность**: PostgreSQL предлагает расширенные функции безопасности, включая аутентификацию на уровне базы данных, шифрование данных и управление правами доступа на уровне строк и столбцов. Это обеспечивает защиту данных от несанкционированного доступа.
6. **Активное сообщество и поддержка**: PostgreSQL имеет большое и активное сообщество разработчиков и пользователей, что обеспечивает доступ к обширной документации, руководствам и онлайн-ресурсам. Пользователи могут легко получать поддержку от опытных специалистов.
7. **Частые обновления и новые функции**: PostgreSQL активно разрабатывается и регулярно получает обновления, которые включают новые функции, исправления ошибок и улучшения производительности. Это гарантирует, что система будет оставаться актуальной и безопасной.
8. **Личный опыт использования:** Я постоянно использую PostgreSQL в своих проектах, так как он мне хорошо знаком. Его удобный интерфейс и многофункциональность позволяют эффективно решать задачи различной сложности. Например, поддержка массивов и JSON-данных предоставляет возможности для работы с более сложными структурами данных, которые отсутствуют в других СУБД. Это делает его идеальным выбором для разработки современных приложений.

## **Триггеры и хранимые процедуры**

Триггеры и хранимые процедуры для базы данных магазина компьютерной техники могут быть созданы, чтобы автоматизировать определенные операции или проверки при выполнении изменений в базе данных.

## **Хранимая функция для обновления общей суммы заказа**

Хранимая функция **update\_order\_total** предназначена для автоматического обновления общей суммы заказа в таблице «**orders**». Она вычисляет новую общую сумму заказа, суммируя стоимость всех элементов заказа из таблицы «**order\_items**» и обновляет соответствующую запись в таблице «**orders**». Функция будет вызываться триггером после вставки, обновления или удаления элементов заказа.



## **Триггер для обновления общей суммы заказа**

Триггер **order\_total\_update** будет активироваться после каждой вставки, обновления или удаления записи в таблице «**order\_items**». Он будет вызывать функцию **update\_order\_total**, чтобы обеспечить автоматическое обновление общей суммы заказа при любых изменениях в связанных элементах заказа, что позволяет поддерживать целостность данных и актуальность информации о заказах.

