

Федеральное государственное автономное образовательное учреждение высшего
образования

НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ

«ВЫСШАЯ ШКОЛА ЭКОНОМИКИ»

Высшая школа бизнеса

Контрольное Домашнее Задание

по предмету Базы Данных

«Разработка Базы Данных для магазина спортивных товаров
GlobalDrive»

по направлению подготовки 38.03.05 Бизнес-информатика
образовательная программа «Бизнес-информатика»

Проект выполнили:

Бекнеева Гиляна Мергеновна ББИ231

Текеев Баир Валерьевич ББИ231

Трескин Владислав Сергеевич ББИ231

Рябов Семён Кириллович ББИ238

Москва 2025

СОДЕРЖАНИЕ

Анализ предметной области.....	3
Тема.....	3
Описание компании.....	3
Цель.....	3
Анализ рынка.....	4
Потенциальные пользователи и их потребности.....	4
Разработка.....	5
Основные сущности и их связи.....	5
База Данных.....	12
Хранимые процедуры.....	12
Триггер.....	15
Запросы.....	16
Визуализация.....	22
Планируемый перечень отчётных форм.....	24
Отчет Revenue By Category.....	24
Отчет “Оплата по клиентам”.....	24
Отчет ”Клиенты и их покупки”.....	25
Отчет “Выручка по месяцам”.....	26
Вклад участников в проект.....	27

Анализ предметной области

Тема

Для проекта нашей командой была выбрана предметная область - магазин спортивных товаров, управление товарными запасами, заказами и клиентским сервисом для компании GlobalDrive.

Описание компании

GlobalDrive – это российский магазин, специализирующийся на продаже спортивных товаров, техники и аксессуаров для активного образа жизни. Здесь можно найти тренажеры, спортивную экипировку, электротранспорт (гироскутеры, моноколеса), а также гаджеты для фитнеса, такие как умные часы и трекеры. Магазин предлагает товары через розничные точки, с доставкой по России. GlobalDrive сотрудничает с известными брендами, обеспечивая качество и разнообразие ассортимента.

Цель

Создание и внедрение базы данных для автоматизации процессов компании: учета ассортимента, остатков, поставок, обработки заказов, учет перемещения товара, анализа поведения клиентов. Это позволит сократить операционные издержки за счет исключения ручных ошибок и оптимизации учета и логистики, что повысит конкурентоспособность компании и упростит взаимодействие между сотрудниками. Рассмотрим подробнее, какие бизнес-функции могут быть реализованы с помощью нашей СУБД.

- 1 Автоматизация ключевых бизнес-процессов:
 - Учет товарных запасов в реальном времени.
 - Обработка заказов, возвратов и перемещений товаров между складами.
 - Управление поставками и взаимоотношениями с поставщиками.
- 2 Аналитика и отчетность:
 - Анализ продаж по категориям, клиентам и периодам.
 - Выявление популярных товаров и сезонных трендов.
 - Оценка эффективности сотрудников и маркетинговых кампаний.
- 3 Повышение качества обслуживания:
 - Персонализация предложений для клиентов на основе истории покупок.
 - Сокращение времени обработки заказов и возвратов.
- 4 Контроль и безопасность:
 - Отслеживание изменений цен и себестоимости товаров.
 - Автоматизация проверок (например, соответствие остатков перед продажами).

Анализ рынка

Рынок крупной спортивной техники в России характеризуется высокой средним чеком (от 500 тыс. до 3 млн руб.) а также длинным циклом принятия решения (30-90 дней), ключевой ролью экспертного консультирования и необходимостью сервисного обслуживания.

Основные игроки и наши прямые конкуренты: Motoland (официальный дилер Honda, Yamaha), MotoShock (крупнейший онлайн-маркетплейс), Квадросила (дилер BRP), необходимо будет ориентироваться в том числе на предоставляемый ими сервис и функционал. Ниже мы составили таблицу КФУ для данного рыночного сегмента.

Критерии КФУ	Motoland	MotoShock	Квадросила	GlobalDrive
Качество консультирования	+++	+	+	+
Качество обработки заказов	++	++	+++	++
Понятный сайт	+++	++	+	++
Качество сервисного обслуживания	++	+++	++	+
Скорость обработки заказов	++	+++	+	+
Программы лояльности в разных категориях	++	++	+	+
Trade-in	+	+	+	+

Таблица 1. КФУ

Потенциальные пользователи и их потребности

Потенциальными пользователями нашей СУБД будут являться практически все люди, начиная с клиента и заканчивая руководителем. Приведем примеры, чем именно может быть полезна наш проект для сотрудников компании.

❖ Менеджеры по продажам:

- Доступ к данным о клиентах, истории покупок, остатках товаров.

- Функция: Проверка наличия товара перед оформлением покупки.
- ❖ Специалисты по логистике:
 - Управление перемещениями товаров между складами.
 - Контроль сроков поставок и остатков.
 - Функция: Автоматическое обновление остатков при перемещениях.
- ❖ Финансовый отдел:
 - Отчеты по выручке, себестоимости, прибыли.
 - Анализ платежей и возвратов.
- ❖ Маркетологи:
 - Анализ популярности товаров и эффективности акций.
 - Сегментация клиентов для персонализированных рассылок.
- ❖ Клиенты (через веб-интерфейс):
 - Просмотр каталога, статуса заказов, истории покупок.
 - Функция: Проверка статуса возврата товара.

Разработка

Инфологическая модель

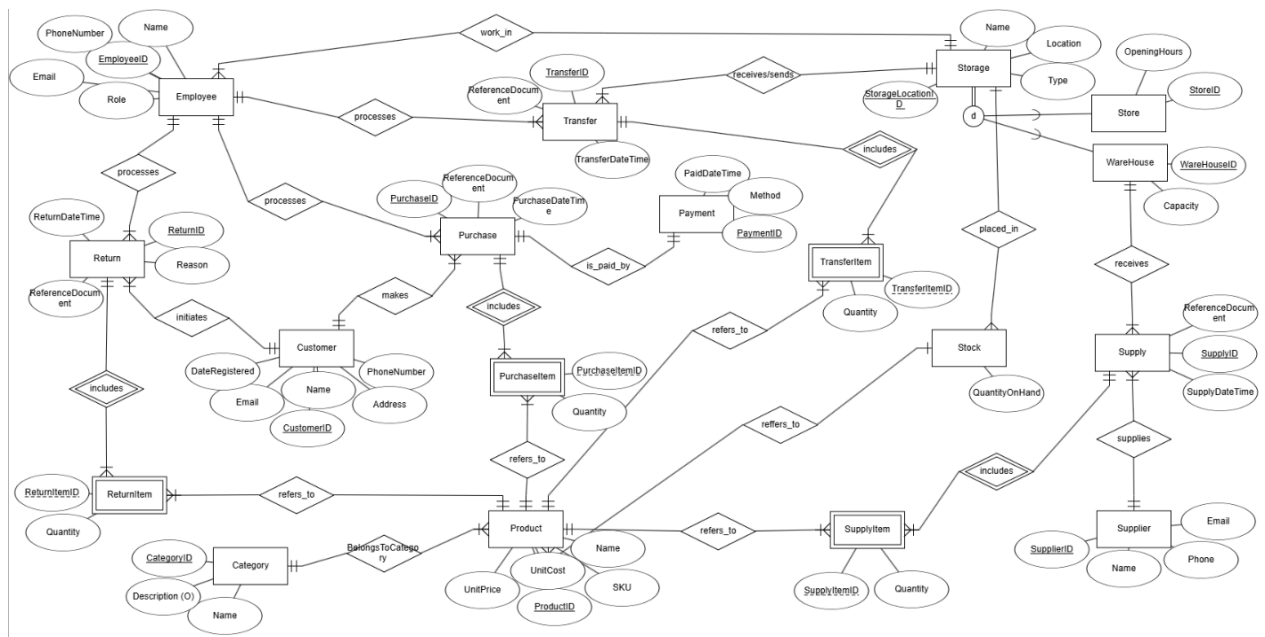


Рис.1 ER - диаграмма

Основные сущности и их связи

Сущность	Атрибут	Тип данных	Описание	Ключ
----------	---------	------------	----------	------

Customer	CustomerID	int	ID клиента	PK
	Name	varchar	ФИО клиента	
	PhoneNumber	varchar	Номер телефона клиента	
	Address	varchar	Адрес проживания клиента	
	Email	varchar	Адрес электронной почты клиента	
	DateRegistered	datetime2	Дата регистрации клиента	
Employee	EmployeeID	int	ID работника	PK
	Name	varchar	ФИО работника	
	Role	varchar	Должность работника	
	PhoneNumber	varchar	Номер телефона работника	
	Email	varchar	Адрес электронной почты работника	
	StorageLocationID	int	ID Магазина/склада, где работает сотрудник	FK
Product	ProductID	int	ID товара	PK
	Name	varchar	Название товара	
	UnitPrice	decimal	Цена продажи товара	
	UnitCost	decimal	Закупочная цена товара	
	SKU	varchar	Уникальный идентификатор товара	
	CategoryID	int	ID категории, к которой относится товар	FK
Category	CategoryID	int	ID категории	PK
	Name	varchar	Название категории	
	Description	varchar	Описание категории	
Payment	PaymentID	int	ID оплаты	PK

	PaidDateTime	datetime2	Время оплаты	
	Method	varchar	Способ оплаты	
	PurchaseID	int	Покупка, по которой была совершена оплата	FK
Purchase	PurchaseID	int	ID покупки	PK
	PurchaseDateTime	datetime2	Время покупки	
	ReferenceDocument	varchar	Описание документа о покупке	
	CustomerID	int	Покупатель, совершивший покупку	FK
	EmployeeID	int	Сотрудник, оформивший покупку	FK
PurchaseItem	PurchaseItemID	int	ID транзакции	PK
	PurchaseID	int	ID покупки к которой относится товар	FK
	ProductID	int	ID продукта в совершенной покупке	FK
	Quantity	int	Количество покупаемого продукта	
Return	ReturnID	int	ID возврата товара	PK
	ReturnDateTime	datetime2	Время возврата товара	
	Reason	varchar	Причина возврата	PK
	ReferenceDocument	varchar	Описание документа о возврате	
	CustomerID	int	ID покупателя, совершившего возврат	FK
	EmployeeID	int	ID сотрудника, принявшего возврат	FK
ReturnItem	ReturnItemID	int	ID транзакции	PK
	ReturnID	int	ID записи о возврате	FK
	ProductID	int	ID возвращаемого	FK

			товара	
	Quantity	int	Количество возвращаемого товара	
Storage	StorageLocationID	int	ID склада	PK
	Name	varchar	Название склада	
	Location	varchar	Местоположение склада	
	Type	varchar	Магазин или склад	
Store	StoreID	int	ID магазина	PK
	OpeningHours	varchar	Часы работы магазина	
WareHouse	WareHouseID	int	ID склада	PK
	Capacity	int	Вместимость склада	
Supply	SupplyID	int	ID поставки	PK
	SupplyDateTime	datetime2	Время поставки	
	ReferenceDocument	varchar	Описание документа о поставке	
	SupplierID	int	ID поставщика	FK
	StorageLocationID	int	ID куда поставили товар	FK
Supplier	SupplierID	int	ID поставщика	PK
	Name	varchar	Наименование поставщика	
	Phone	varchar	Номер телефона поставщика	
	Email	varchar	Адрес электронной почты поставщика	
SupplyItem	SupplyItemID	int	ID транзакции	PK
	SupplyID	int	ID записи о поставке	FK
	ProductID	int	ID полученного товара	FK
	Quantity	int	Количество поставляемого товара	

Transfer	TransferID	int	ID перемещения товара	PK
	TransferDateTime	datetime2	Время перевозки	
	ReferenceDocument	varchar	Описание документа о перевозке	
	EmployeeID	int	ID сотрудника, ответственного за перевозку	FK
	FromStorageID	int	ID магазина/склада куда поставляется товар	FK
	ToStorageID	int	ID магазина/склада откуда поставляется товар	FK
TransferItem	TransferItemID	int	ID транзакции	PK
	TransferID	int	ID записи о перемещении товара	FK
	ProductID	int	ID перемещаемого товара	FK
	Quantity	int	Количество перевезенных товаров	
Stock	StorageID	int	Место хранения товара	FK
	ProductID	int	Какой товар хранится	FK
	QuantityOnHand	int	Количество товара	

Таблица 2. Сущности

Связи сущностей

Связь	Тип связи	Описание
Employee - Storage	1:N	Один склад может быть закреплён за несколькими сотрудниками, каждый сотрудник работает на одном складе.
Employee - Transfer	N:1	Один сотрудник может оформлять множество

		перемещений.
Transfer - TransferItem	N:1	Одно перемещение включает множество товарных позиций.
Storage - Transfer	N:1	Одно хранилище может участвовать во множестве перемещений как отправитель или получатель.
Employee - Return	N:1	Один сотрудник может оформлять множество возвратов.
Employee - Purchase	N:1	Один сотрудник может оформлять множество покупок (продаж).
Customer - Return	N:1	Один клиент может оформить несколько возвратов.
Return - ReturnItem	N:1	Один возврат может содержать несколько позиций товаров.
Customer - Purchase	N:1	Один клиент может сделать несколько покупок.
Purchase - Payment	1:1	Каждая покупка имеет ровно один платёж, и каждый платёж связан с одной покупкой.
WareHouse - Supply	N:1	В один склад может быть сделано много поставок.
Supplier - Supply	N:1	Один поставщик может совершать множество поставок.
Supply - SupplyItem	N:1	Одна поставка содержит множество товарных позиций.
TransferItem - Product	1:N	Один товар может

		участвовать в множестве позиций перемещений
SupplyItem - Product	1:N	Один товар может участвовать в множестве позиций поставок.
PurchaseItem - Product	1:N	Один товар может быть в множестве позиций покупок
ReturnItem - Product	1:N	Один товар может быть в множестве возвратов
Product - Category	1:N	Один товар может быть в множестве возвратов
Purchase - PurchaseItem	N:1	Каждая покупка содержит несколько позиций товаров
Product - Stock	1:N	В стоке может находиться много товаров
Stock - Storage	1:N	Позиции товаров могут находиться в различных местах

Таблица 3. Связи сущностей

Даталогическая модель

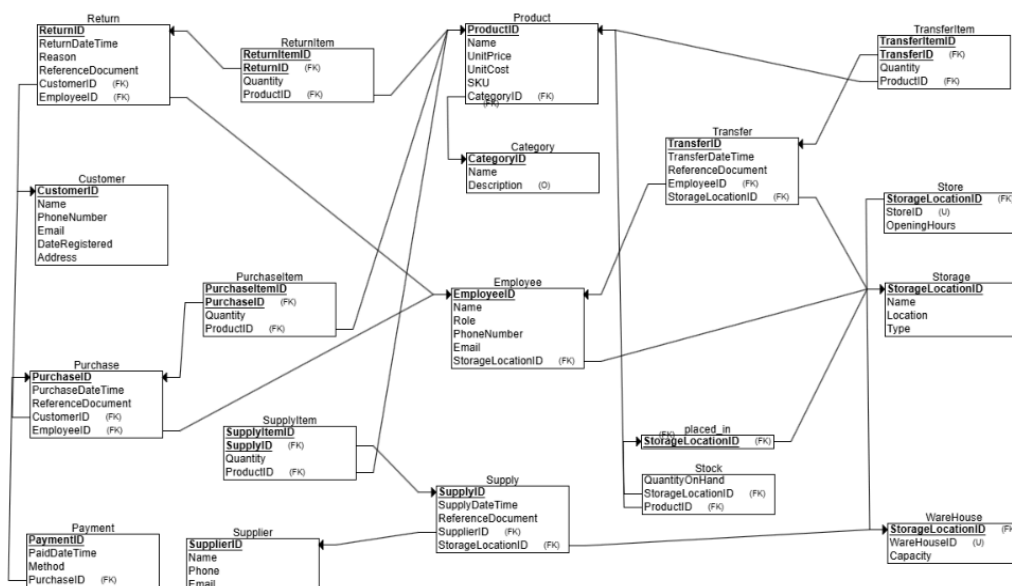


Рис.3 Даталогическая модель

База Данных

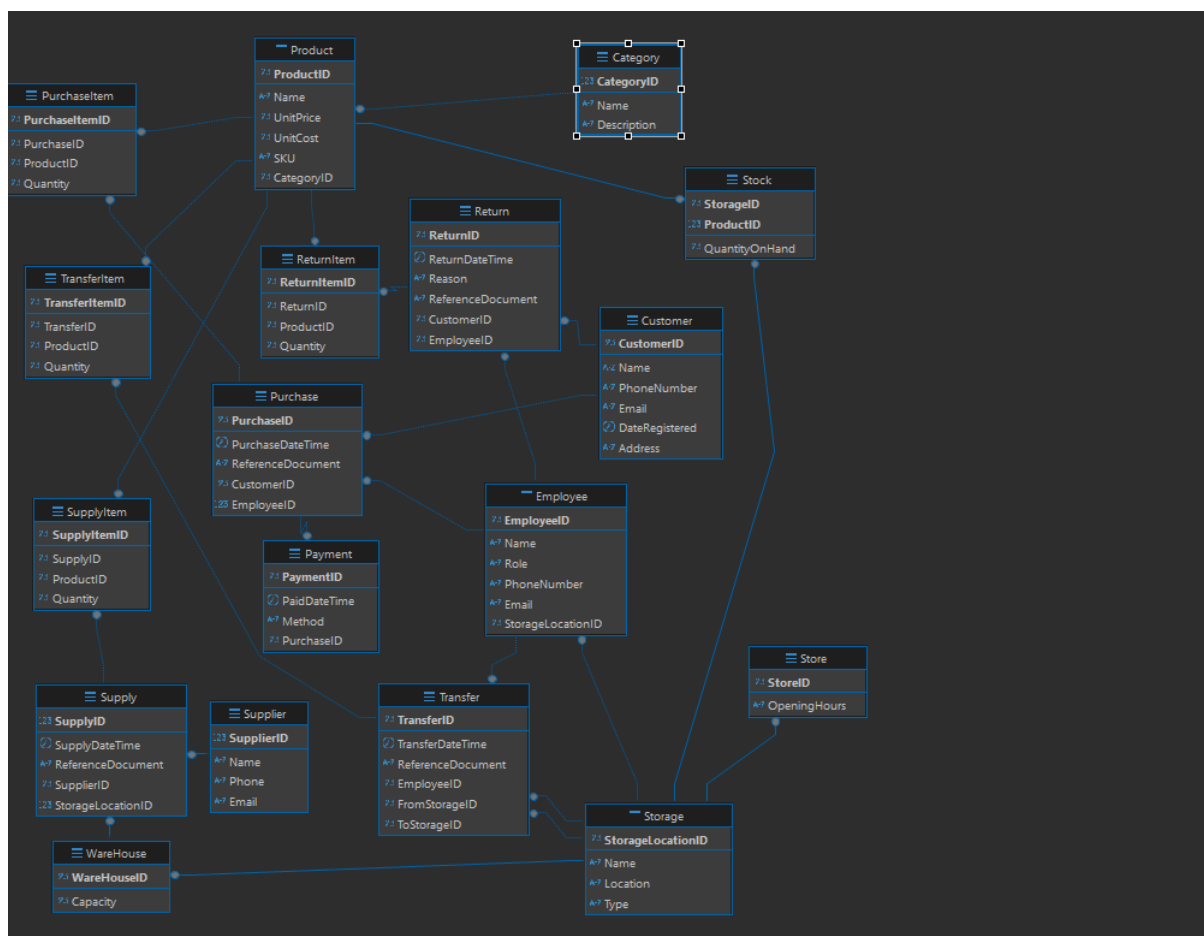


Рис.4 Диаграмма БД

Данная база данных находится в третьей нормальной форме (3NF), поскольку каждая таблица имеет только атомарные атрибуты (не содержащие множественных значений), каждый неключевой атрибут полностью зависит от первичного ключа, а в таблицах нет транзитивных зависимостей, где неключевые атрибуты зависят от других неключевых атрибутов. Все связи между таблицами правильно реализованы через внешние ключи, обеспечивая целостность данных и минимизируя избыточность.

Хранимые процедуры

Для упрощения работы с данными и автоматизации ключевых операций мы разработали несколько хранимых процедур - они охватывают основные процессы: продажи, возвраты и перемещения товаров. Это позволяет ускорить обработку и

повысить точность.

Процедура 1.

Процедура `sp_InsertCustomer` предназначена для добавления нового клиента в таблицу `Customer`. Она принимает четыре обязательных параметра: имя клиента, номер телефона, адрес электронной почты и адрес проживания. После вызова процедура выполняет вставку этих данных в соответствующие поля таблицы.

```
CREATE OR ALTER PROCEDURE dbo.sp_InsertCustomer
    @Name          VARCHAR(100),
    @PhoneNumber    VARCHAR(20),
    @Email          VARCHAR(100),
    @Address        VARCHAR(200)
AS
BEGIN
    INSERT INTO dbo.Customer(Name, PhoneNumber, Email, Address)
    VALUES(@Name, @PhoneNumber, @Email, @Address);
END;
```

Рис.5 Процедура 1

Процедура 2.

Процедура `sp_RegisterReturn` оформляет возврат товаров по ранее совершённой покупке. В качестве входных параметров она принимает идентификатор покупки, табличный список возвращаемых товаров с их количеством, а также причину возврата. Сначала процедура проверяет, существует ли указанная покупка, не истёк ли срок возврата (более 30 дней), и что количество возвращаемых товаров положительное. Если все условия соблюдены, создаётся новая запись о возврате в таблице `Return`, после чего ей присваивается уникальный код `ReferenceDocument` в формате `RET-ACT-<ReturnID>-<Year>`. Далее все указанные товары добавляются в таблицу `ReturnItem`, привязываясь к созданному возврату.

```

CREATE PROCEDURE dbo.sp_RegisterReturn
    @PurchaseID INT,
    @ProductID INT,
    @Quantity INT,
    @Reason NVARCHAR(200)
AS
BEGIN
    SET NOCOUNT ON;

    DECLARE
        @PurchaseDateTime DATETIME2,
        @CustomerID INT,
        @EmployeeID INT,
        @DaysSince INT,
        @ReturnID INT;

    SELECT
        @PurchaseDateTime = p.PurchaseDateTime,
        @CustomerID = p.CustomerID,
        @EmployeeID = p.EmployeeID
    FROM dbo.Purchase p
    WHERE PurchaseID = @PurchaseID;

    IF @PurchaseDateTime IS NULL
    BEGIN
        RAISERROR('PurchaseID %d not found.', 16, 1, @PurchaseID);
        RETURN;
    END

    SET @DaysSince = DATEDIFF(DAY, @PurchaseDateTime, SYSUTCDATETIME());

    IF @DaysSince > 30
    BEGIN
        RAISERROR('Return period expired: %d days > 30.', 16, 1, @DaysSince);
        RETURN;
    END

    IF @Quantity <= 0
    BEGIN
        RAISERROR('Quantity must be positive.', 16, 1);
        RETURN;
    END

    INSERT INTO dbo.[Return] (
        ReturnDateTime,
        Reason,
        ReferenceDocument,
        CustomerID,
        EmployeeID
    )
    VALUES (
        CAST(SYSDATETIME() AS datetime),
        @Reason,
        '',
        @CustomerID,
        @EmployeeID
    );

    SET @ReturnID = SCOPE_IDENTITY();

    UPDATE dbo.[Return]
    SET ReferenceDocument = 'RET-ACT-' + CAST(@ReturnID AS NVARCHAR) + '-' + CAST(YEAR(SYSDATETIME()) AS NVARCHAR)
    WHERE ReturnID = @ReturnID;

    INSERT INTO dbo.ReturnItem (
        ReturnID,
        ProductID,
        Quantity
    )
    VALUES (
        @ReturnID,
        @ProductID,
        @Quantity
    );
END;

```

Рис.6 Процедура 2

Процедура 3

Процедура sp_UpdateCustomerContact предназначена для обновления контактной информации клиента в таблице Customer. В неё передаются идентификатор клиента и новые значения полей: имя, номер телефона, email и адрес. Все параметры, кроме CustomerID, являются необязательными. Сначала процедура проверяет, существует ли клиент с указанным CustomerID. Если клиент найден, выполняется обновление: каждое из полей изменяется только в том случае, если для него было передано новое значение, в противном случае сохраняется старое.

```
CREATE PROCEDURE dbo.sp_UpdateCustomerContact
    @CustomerID INT,
    @Name NVARCHAR(100) = NULL,
    @PhoneNumber NVARCHAR(20) = NULL,
    @Email NVARCHAR(100) = NULL,
    @Address NVARCHAR(200) = NULL
AS
BEGIN
    SET NOCOUNT ON;

    IF NOT EXISTS (SELECT 1 FROM dbo.Customer WHERE CustomerID = @CustomerID)
    BEGIN
        RAISERROR('Customer %d not found.', 16, 1, @CustomerID);
        RETURN;
    END

    UPDATE dbo.Customer
    SET
        Name = COALESCE(@Name, Name),
        PhoneNumber = COALESCE(@PhoneNumber, PhoneNumber),
        Email = COALESCE(@Email, Email),
        Address = COALESCE(@Address, Address)
    WHERE CustomerID = @CustomerID;
END
```

Рис.7 Процедура 3

Триггер

Триггер trg_UpdateStockOnSupplyInsert автоматически обновляет таблицу Stock при добавлении новых строк в SupplyItem: если для товара в указанном хранилище уже есть запись, увеличивает QuantityOnHand, иначе создаёт новую строку с

начальным количеством. Это гарантирует актуальность остатков без ручных операций после каждой поставки.

```
CREATE OR ALTER TRIGGER trg_UpdateStockOnSupplyInsert ON kdz_group13.dbo.SupplyItem
AFTER INSERT
AS
BEGIN
    SET NOCOUNT ON;

    MERGE dbo.Stock AS target
    USING (
        SELECT
            i.ProductID,
            s.StorageLocationID AS StorageID,
            i.Quantity
        FROM inserted AS i
        JOIN dbo.Supply AS s
            ON s.SupplyID = i.SupplyID
    ) AS source(ProductID, StorageID, Quantity)
    ON target.ProductID = source.ProductID
    AND target.StorageID = source.StorageID

    WHEN MATCHED THEN
        UPDATE SET target.QuantityOnHand = target.QuantityOnHand + source.Quantity

    WHEN NOT MATCHED BY TARGET THEN
        INSERT (StorageID, ProductID, QuantityOnHand)
        VALUES (source.StorageID, source.ProductID, source.Quantity);
END;
```

Рис.8 Триггер

Запросы

Запрос 1

Собирает для каждого товара суммарные объемы продаж и итоговую прибыль, но только по тем позициям, где было продано не менее 5 штук за раз.

```
SELECT
    pr.Name AS ProductName,
    SUM(pi.Quantity) AS TotalQuantity,
    pr.UnitPrice,
    pr.UnitCost,
    SUM((pr.UnitPrice - pr.UnitCost) * pi.Quantity) AS TotalProfit
FROM dbo.PurchaseItem AS pi
JOIN dbo.Product AS pr
    ON pr.ProductID = pi.ProductID
WHERE pi.Quantity >= 5
GROUP BY
    pr.Name,
    pr.UnitPrice,
    pr.UnitCost
```

Рис.9 Запрос 1

Запрос 2

Показывает коэффициент наценки (markup ratio) у товаров с высокой рентабельностью.

```
SELECT
    pr.Name AS ProductName,
    pr.UnitCost,
    pr.UnitPrice,
    ROUND(pr.UnitPrice / pr.UnitCost, 2) AS MarkupRatio
FROM dbo.Product AS pr
WHERE pr.UnitPrice / pr.UnitCost > 1.5
```

Рис.10 Запрос 2

Запрос 3

Показывает топ-10 продуктов по суммарному количеству продаж.

```
SELECT TOP(10)
    stats.ProductID,
    stats.ProductName,
    stats.TotalSold
FROM (
    SELECT
        pr.ProductID,
        pr.Name AS ProductName,
        SUM(pi.Quantity) AS TotalSold
    FROM dbo.PurchaseItem AS pi
    JOIN dbo.Product AS pr ON pr.ProductID = pi.ProductID
    GROUP BY pr.ProductID, pr.Name
) AS stats
ORDER BY stats.TotalSold DESC
```

Рис.11 Запрос 3

Запрос 4

Показывает выручку по категориям.

```
SELECT
    c.Name AS CategoryName,
    ct.CatRevenue
FROM (
    SELECT
        pr.CategoryID,
        SUM(pi.Quantity * pr.UnitPrice) AS CatRevenue
    FROM dbo.PurchaseItem AS pi
    JOIN dbo.Product AS pr ON pr.ProductID = pi.ProductID
    GROUP BY pr.CategoryID
) AS ct
JOIN dbo.Category AS c
ON c.CategoryID = ct.CategoryID
ORDER BY ct.CatRevenue DESC
```

Рис.12 Запрос 4

Запрос 5

Показывает топ-10 продуктов по выручке.

```

SELECT TOP(10)
    sub.ProductID,
    sub.ProductName,
    sub.TotalRevenue
FROM (
    SELECT
        pr.ProductID,
        pr.Name AS ProductName,
        SUM(pi.Quantity * pr.UnitPrice) AS TotalRevenue
    FROM dbo.PurchaseItem AS pi
    JOIN dbo.Product AS pr ON pr.ProductID = pi.ProductID
    GROUP BY pr.ProductID, pr.Name
) AS sub
ORDER BY sub.TotalRevenue DESC

```

Рис.13 Запрос 5

Запрос 6

Показывает дневную выручку за весь период по убыванию

```

SELECT
    sub.OrderDate,
    sub.DailyRevenue
FROM (
    SELECT
        CAST(p.PurchaseDateTime AS DATE) AS OrderDate,
        SUM(pi.Quantity * pr.UnitPrice) AS DailyRevenue
    FROM dbo.Purchase AS p
    JOIN dbo.PurchaseItem AS pi ON pi.PurchaseID = p.PurchaseID
    JOIN dbo.Product AS pr ON pr.ProductID = pi.ProductID
    GROUP BY CAST(p.PurchaseDateTime AS DATE)
) AS sub
ORDER BY sub.DailyRevenue DESC

```

Рис.14 Запрос 6

Запрос 7

Выводит товары дороже средней цены своей категории

```

SELECT
    pr.ProductID,
    pr.Name AS ProductName,
    pr.UnitPrice,
    pr.UnitCost
FROM dbo.Product AS pr
WHERE pr.UnitPrice > (
    SELECT AVG(pr2.UnitPrice)
    FROM dbo.Product AS pr2
    WHERE pr2.CategoryID = pr.CategoryID
)

```

Рис.15 Запрос 7

Запрос 8

Показывает список товаров в каждой категории, чья цена выше, чем у большинства аналогов.

```
SELECT
    p.ProductID,
    p.Name,
    p.UnitPrice,
    p.CategoryID
FROM kdz_group13.dbo.Product AS p
WHERE p.UnitPrice > (
    SELECT AVG(p2.UnitPrice)
    FROM kdz_group13.dbo.Product AS p2
    WHERE p2.CategoryID = p.CategoryID
);
```

Рис.16 Запрос 8

Запрос 9

Показывает выручку и число заказов по каждому сотруднику

```
SELECT
    e.EmployeeID,
    e.Name AS EmployeeName,
    COUNT(DISTINCT p.PurchaseID) AS OrdersCount,
    SUM(pi.Quantity * pr.UnitPrice) AS TotalRevenue
FROM dbo.Purchase AS p
JOIN dbo.PurchaseItem AS pi ON pi.PurchaseID = p.PurchaseID
JOIN dbo.Product AS pr ON pr.ProductID = pi.ProductID
JOIN dbo.Employee AS e ON e.EmployeeID = p.EmployeeID
GROUP BY
    e.EmployeeID,
    e.Name;
```

Рис.17 Запрос 9

Запрос 10

Показывает общее количество и суммарную стоимость закупок по каждому поставщику

```
SELECT
    s.SupplierID,
    s.Name AS SupplierName,
    COUNT(DISTINCT sup.SupplyID) AS SuppliesCount,
    SUM(si.Quantity * pr.UnitCost) AS TotalCost
FROM dbo.Supply AS sup
JOIN dbo.SupplyItem AS si ON si.SupplyID = sup.SupplyID
JOIN dbo.Product AS pr ON pr.ProductID = si.ProductID
JOIN dbo.Supplier AS s ON s.SupplierID = sup.SupplierID
GROUP BY
    s.SupplierID,
    s.Name;
```

Рис.18 Запрос 10

Запрос 11

Показывает объём поставок и число поставок по каждому поставщику

```
SELECT
    s.SupplierID,
    s.Name AS SupplierName,
    COUNT(DISTINCT sup.SupplyID) AS SupplyCount,
    SUM(si.Quantity) AS TotalSuppliedQty
FROM kdz_group13.dbo.Supply AS sup
JOIN kdz_group13.dbo.SupplyItem AS si ON sup.SupplyID = si.SupplyID
JOIN kdz_group13.dbo.Supplier AS s ON sup.SupplierID = s.SupplierID
GROUP BY
    s.SupplierID,
    s.Name
ORDER BY TotalSuppliedQty DESC;
```

Рис.19 Запрос 11

Запрос 12

Показывает какие сотрудники фактически обрабатывали заказы

```
SELECT
    e.EmployeeID,
    e.Name AS EmployeeName
FROM dbo.Employee AS e
WHERE EXISTS (
    SELECT 1
    FROM dbo.Purchase AS p
    WHERE p.EmployeeID = e.EmployeeID
);
```

Рис.20 Запрос 12

Запрос 13

Показывает клиентов, которые сделали покупки, но ни разу не возвращали товары

```
SELECT DISTINCT p.CustomerID
FROM dbo.Purchase AS p
EXCEPT
SELECT DISTINCT r.CustomerID
FROM dbo.[Return] AS r;
```

Рис.21 Запрос 13

Запрос 14

Показывает клиентов, которые еще не делали ни одного заказа

```
SELECT
    c.CustomerID,
    c.Name AS CustomerName
FROM dbo.Customer AS c
LEFT JOIN dbo.Purchase AS p
    ON p.CustomerID = c.CustomerID
WHERE p.PurchaseID IS NULL;
```

Рис.22 Запрос 14

Запрос 15

Показывает суммарную выручку, себестоимость и прибыль по категориям товаров и месяцам на основе проданных товаров.

```
SELECT
    c.Name AS CategoryName,
    YEAR(p.PurchaseDateTime) AS SaleYear,
    MONTH(p.PurchaseDateTime) AS SaleMonth,
    SUM(pi.Quantity * pr.UnitPrice) AS TotalRevenue,
    SUM(pi.Quantity * pr.UnitCost) AS TotalCost,
    SUM(pi.Quantity * (pr.UnitPrice - pr.UnitCost)) AS TotalProfit
FROM kdz_group13.dbo.PurchaseItem AS pi
JOIN kdz_group13.dbo.Purchase AS p ON pi.PurchaseID = p.PurchaseID
JOIN kdz_group13.dbo.Product AS pr ON pi.ProductID = pr.ProductID
JOIN kdz_group13.dbo.Category AS c ON pr.CategoryID = c.CategoryID
GROUP BY
    c.Name,
    YEAR(p.PurchaseDateTime),
    MONTH(p.PurchaseDateTime)
ORDER BY
    c.Name,
    SaleYear,
    SaleMonth;
```

Рис.23 Запрос 15

Запрос 16

Показывает сколько покупок было оплачено наличными и картой по каждому сотруднику.

```
SELECT
    e.EmployeeID,
    e.Name AS EmployeeName,
    COUNT(CASE WHEN p.Method = 'Cash' THEN 1 END) AS CashPayments,
    COUNT(CASE WHEN p.Method = 'Card' THEN 1 END) AS CardPayments
FROM dbo.Employee e
JOIN dbo.Purchase pu ON pu.EmployeeID = e.EmployeeID
JOIN dbo.Payment p ON p.PurchaseID = pu.PurchaseID
GROUP BY e.EmployeeID, e.Name;
```

Рис.24 Запрос 16

Запрос 17

Показывает сотрудников, у которых было больше 10 покупок.

```
SELECT
    e.EmployeeID,
    e.Name AS EmployeeName,
    COUNT(pu.PurchaseID) AS TotalPurchases
FROM dbo.Employee e
JOIN dbo.Purchase pu ON pu.EmployeeID = e.EmployeeID
GROUP BY e.EmployeeID, e.Name
HAVING COUNT(pu.PurchaseID) > 10;
```

Рис.25 Запрос 17

Запрос 18

Показывает ежедневную выручку и накопленную выручку с начала продаж до каждой даты.

```
SELECT
    sub.OrderDate,
    sub.DailyRevenue,
    SUM(sub.DailyRevenue) OVER (
        ORDER BY sub.OrderDate
        ROWS BETWEEN UNBOUNDED PRECEDING AND CURRENT ROW
    ) AS CumulativeRevenue
FROM (
    SELECT
        CAST(p.PurchaseDateTime AS DATE) AS OrderDate,
        SUM(pi.Quantity * pr.UnitPrice) AS DailyRevenue
    FROM dbo.Purchase AS p
    JOIN dbo.PurchaseItem AS pi ON pi.PurchaseID = p.PurchaseID
    JOIN dbo.Product AS pr ON pr.ProductID = pi.ProductID
    GROUP BY CAST(p.PurchaseDateTime AS DATE)
) AS sub
ORDER BY sub.OrderDate;
```

Рис.26 Запрос 18

Запрос 19

Показывает ежедневную выручку, выручку за предыдущий день и разницу между ними

```
SELECT
    sub.OrderDate,
    sub.DailyRevenue,
    LAG(sub.DailyRevenue) OVER (
        ORDER BY sub.OrderDate
    ) AS PrevDayRevenue,
    sub.DailyRevenue
    - LAG(sub.DailyRevenue) OVER (ORDER BY sub.OrderDate)
    AS ChangeFromPrevDay
FROM (
    SELECT
        CAST(p.PurchaseDateTime AS DATE) AS OrderDate,
        SUM(pi.Quantity * pr.UnitPrice) AS DailyRevenue
    FROM dbo.Purchase AS p
    JOIN dbo.PurchaseItem AS pi ON pi.PurchaseID = p.PurchaseID
    JOIN dbo.Product AS pr ON pr.ProductID = pi.ProductID
    GROUP BY CAST(p.PurchaseDateTime AS DATE)
) AS sub
ORDER BY sub.OrderDate;
```

Рис.27 Запрос 19

Визуализация

Мы использовали представление “vw_RecentReturns”, которое содержит данные о возвратах: клиент, причина возврата, сотрудник, товар и количество. На его основе мы визуализировали данные в Excel, построив столбчатую диаграмму по причинам возвратов, агрегируя количество возвращённых товаров. Это даёт возможность выявить основные причины возвратов и принять управленческие меры для их

сокращения. Данные доступные в excel формате [Визуализация1.xlsx](#)

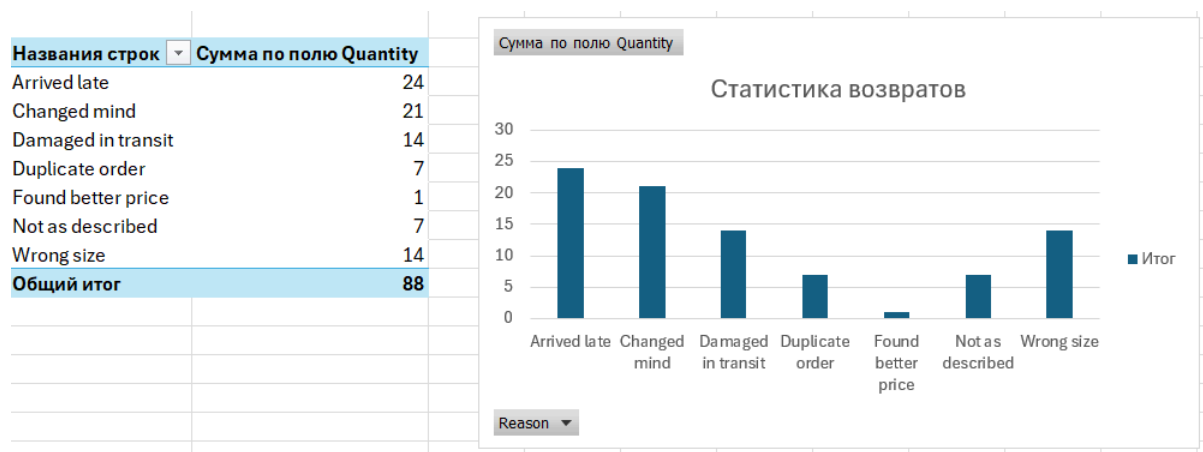


Рис.28 Визуализация причин возврата

Представление “vw_SalesByCategory” агрегирует выручку и число заказов по категориям товаров. Мы визуализировали данные с помощью **столбчатой диаграммы**, отобразив количество заказов (OrdersCount) по категориям (CategoryName). Это позволяет определить наиболее популярные категории и оценить, насколько эффективно каждая из них приносит прибыль. Данные доступны в excel формате [Визуализация2.xlsx](#)

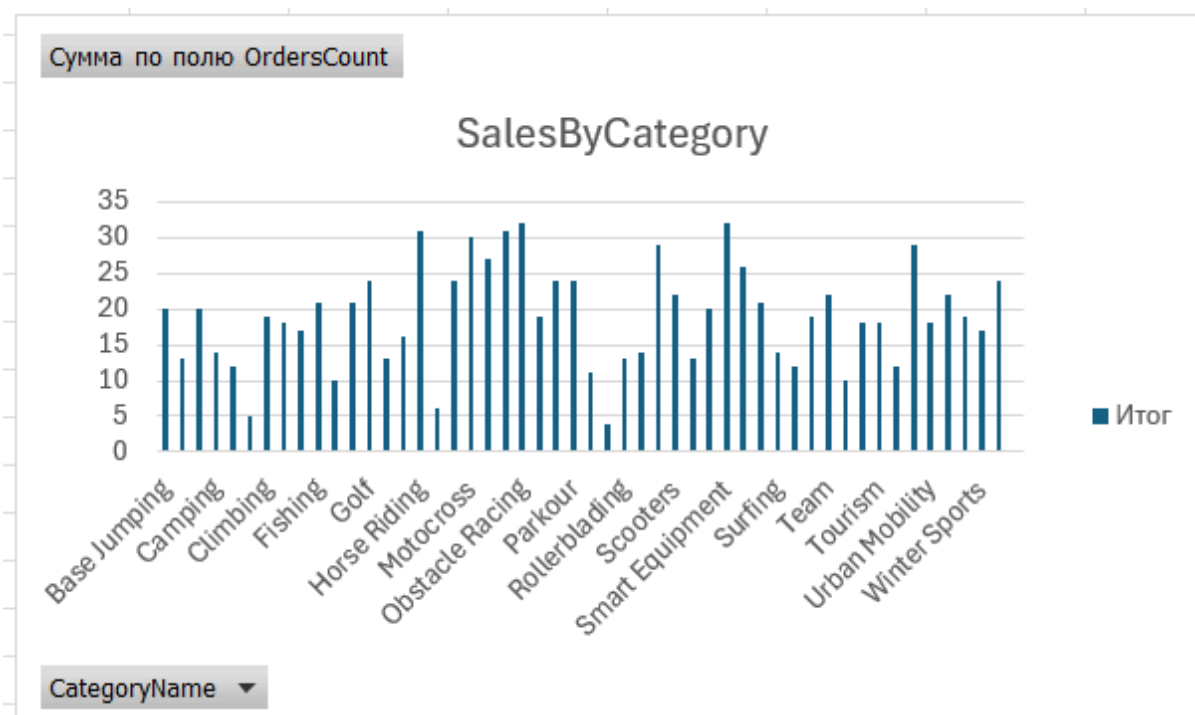


Рис.29 График выручки по категориям

Планируемый перечень отчётных форм

Отчет Revenue By Category

В формате столбчатой диаграммы демонстрирует суммарную выручку по каждой товарной категории. Он позволяет визуально оценить вклад каждой группы товаров в общие продажи компании. Благодаря понятной структуре диаграммы можно быстро выявить лидеров и аутсайдеров по доходности, а также использовать эти данные для стратегического планирования ассортимента. Такой отчет особенно полезен при формировании маркетинговых приоритетов и при принятии решений о расширении или сокращении товарных линеек.

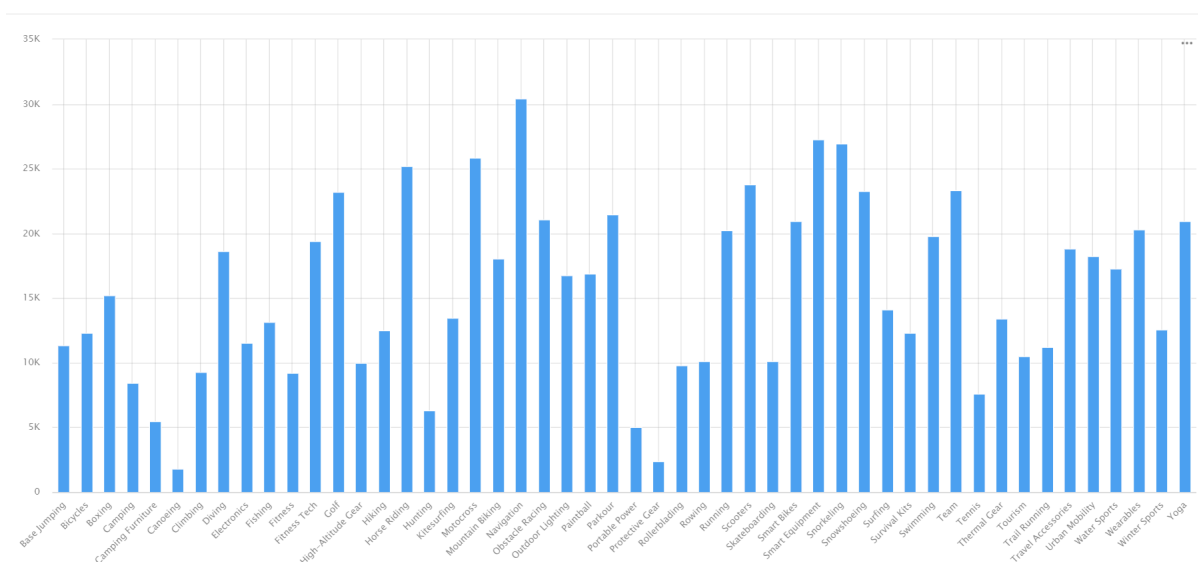


Рис.29 Отчет RevenueByCategory

Дашборды

Отчет “Оплата по клиентам”

Этот дашборд отображает топ клиентов по выручке и включает динамический фильтр по категории товара. Пользователь может выбрать интересующую категорию, после чего таблица автоматически обновится, показывая тех клиентов, которые

принесли наибольшую выручку именно в выбранной категории.

Новый дашборд

оплата по клиентам

CustomerName	SUM([TotalPrice])
Tyler Paul	2 214,40
Natalie Wells	1 854,65
Elizabeth Hardy DVM	1 765,30
Bryan Schneider	1 414,62
Kimberly Long	1 199,52
Rachel Austin	1 059,18
Sarah Gonzalez	749,70

CategoryName*

Bicycles

Поиск

Base Jumping

Bicycles

Boxing

Camping

Camping Furniture

Canoeing

Climbing

Diving

Electronics

Рис.30 Дашборд “Оплата по клиентам”

CustomerName	SUM([TotalPrice])	...
Tyler Paul		2 214,40
Natalie Wells		1 854,65
Elizabeth Hardy DVM		1 765,30
Bryan Schneider		1 414,62
Kimberly Long		1 199,52
Rachel Austin		1 059,18
Sarah Gonzalez		749,70
Jordan Newman		664,32
Cody Ward		552,45
Shawna Parrish		449,82
Yessenia Green		149,94
Heidi Berger		111,40
Melissa Burke		111,40

Рис.31 Отчет “Оплата по клиентам”

<https://datalens.yandex/sdznrovjx9o8e>

Отчет “Клиенты и их покупки”

Этот дашборд представляет собой детализированную таблицу покупок клиента, в которой пользователь может выбрать одного клиента и увидеть все его транзакции и в каком магазине.

клиенты и их покупки

клиент какие купил товары — Таблица

ClientName	PurchaseDateTime	ProductName	UnitPrice	Quantity	TotalPrice	Method
Aaron Fuller	29.08.2023 19:49:46	Factor Spring	288,85	5	1 444,25	Mobile Payment
Aaron Fuller	25.11.2023 06:01:57	Draw Help	268,53	7	1 879,71	Bank Transfer
Aaron Fuller	10.12.2023 10:19:16	International Response	96,80	3	290,40	Cash
Aaron Richardson	17.09.2023 05:57:45	Film Hit	103,95	6	623,70	Card
Aaron Richardson	17.09.2023 05:57:45	Film Hit	103,95	6	623,70	Mobile Payment
Aaron Richardson	17.09.2023 05:57:45	Film Hit	103,95	6	623,70	Online
Aaron Taylor	09.02.2024 16:57:52	Exactly View	156,70	7	1 096,90	Card
Aaron Taylor	09.02.2024 16:57:52	Exactly View	156,70	7	1 096,90	Cash
Abigail Fowler	19.03.2025 03:00:57	His Vote	82,56	4	330,24	Card
Abigail Fowler	19.03.2025 03:00:57	His Vote	82,56	4	330,24	Mobile Payment
Abigail Fowler	19.03.2025 03:00:57	His Vote	82,56	4	330,24	Online
Adam Brandt	25.08.2023 00:25:46	Clear Box	199,71	9	1 797,39	Online
Alexander Allen MD	23.01.2024 06:59:52	Game Finish	219,17	4	876,68	Card
Alexander Allen MD	23.01.2024 06:59:52	Game Finish	219,17	4	876,68	Online
Alexander Allen MD	06.04.2025 01:08:42	Artist Fact	334,75	1	334,75	Cash
Alexander Allen MD	06.04.2025 01:08:42	Artist Fact	334,75	1	334,75	Online
Alexandra Freeman	06.11.2023 02:55:08	Only Instead	218,17	6	1 309,02	Mobile Payment
Alexandra Freeman	06.11.2023 02:55:08	Only Instead	218,17	6	1 309,02	Online
Alexandra Freeman	06.11.2023 02:55:08	Only Instead	218,17	6	1 309,02	Online

ClientName

Нет выбранных значений...

Поиск

Выбрать все

Aaron Coleman

Aaron Fuller

Aaron Richardson

Aaron Sandoval

Aaron Taylor

Abigail Fowler

Abigail Navarro

Adam Brandt

Adam Castro

ClientName	PurchaseDateTime	ProductName	UnitPrice	Quantity	TotalPrice	Method	Store Name
Aaron Fuller	29.08.2023 19:49:46	Factor Spring	288,85	5	1 444,25	Mobile Payment	Branch 2
Aaron Fuller	25.11.2023 06:01:57	Draw Help	268,53	7	1 879,71	Bank Transfer	Outlet 50
Aaron Fuller	10.12.2023 10:19:16	International Response	96,80	3	290,40	Cash	Distribution Center

Рис.32 Отчет “Клиенты и их покупки”

<https://datalens.yandex/yj5w6nba6lkok>

Отчет “Выручка по месяцам”

Отчёт выручка по месяцам отображает суммарную выручку по категориям товаров с разбивкой по месяцам и годам в формате столбчатой диаграммы. Визуализация построена на агрегированных данных, сгруппированных по названию категории, дате покупки и году, что позволяет отслеживать динамику продаж в разных временных периодах. В нижней части панели реализованы фильтры по месяцу, году и категории, что делает отчёт интерактивным и позволяет гибко анализировать изменения выручки по выбранным параметрам.

26

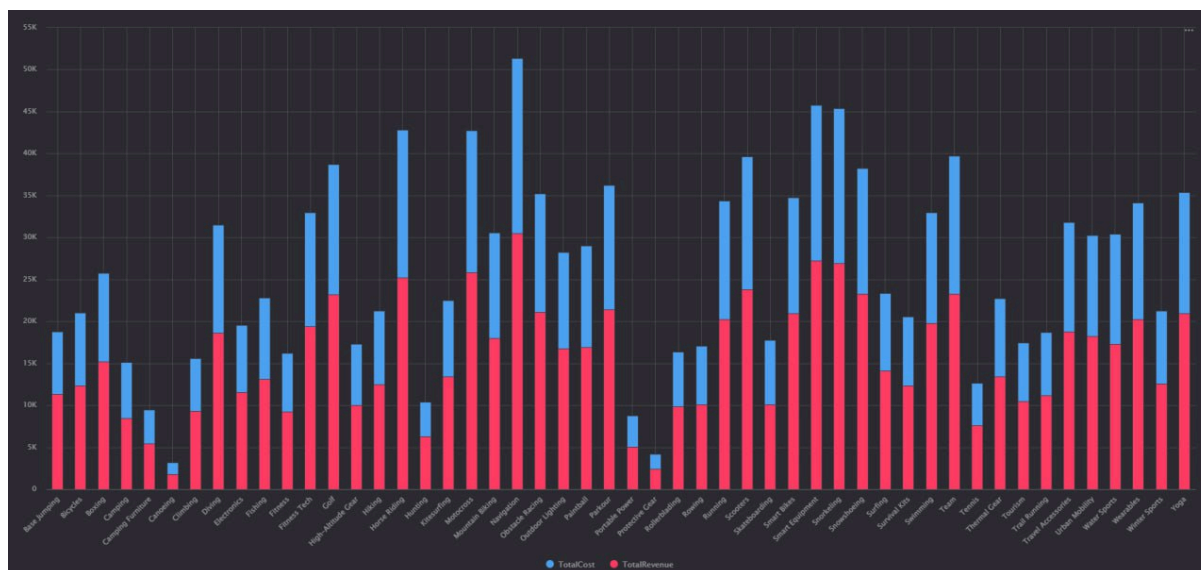


Рис.33 Отчет “Выручка по месяцам”

<https://datalens.yandex/rcyshrjxed8md>

Отчет “Анализ выручки и себестоимости по категориям товаров”

Данный дашборд отображает сравнение выручки и себестоимости по категориям товаров на основе данных представления запроса 15. Это позволяет быстро определить наиболее прибыльные и наименее эффективные категории, визуальнo сравнив разницу между доходом и затратами. Такой анализ помогает бизнесу принимать решения по оптимизации ассортимента и ценовой стратегии.



Вклад участников в проект

В данном проекте по разработке базы данных для магазина “GlobalDrive” принимали участие Бекнеева Гиляна Мергеновна, Тресикн Владислав Сергеевич, Текеев Баир Валерьевич и Рябов Семен Кириллович. Весь проект был выполнен слаженной командой, в которой каждый участник внес значительный вклад. Стоит отметить, что мы старались выполнять большинство этапов КДЗ вместе, чтобы все участники смогли овладеть необходимыми навыками работы с базами данных и формами отчетности.

Ниже прикладываем таблицу, чтобы более наглядно показать как именно распределялась работа между участниками:

Этапы	Тресикн Владислав	Бекнеева Гиляна	Текеев Баир	Рябов Семен
Анализ предметной области	+	+		
Инфологическая модель		+	+	+

Даталогическая модель		+	+	
Разработка диаграммы БД	+			+
Размещение БД в СУБД	+		+	+
Заполнение базы данных		+		+
Настройка индексов			+	+
Полезные запросы	+	+		
Хранимые процедуры		+	+	
Триггеры	+			+
Представления	+		+	+
Функции		+	+	
Отчетные формы	+	+		+
Отчет о проделанной работе	+		+	