

Лабораторная работа №5 Индексы, хранимые процедуры, представления, триггеры	Ф.И.О.	Дьячкова Анастасия Алексеевна
	Группа	ИБТ-263
	Преподаватель	Аль-Мерри Гаис
	Дата сдачи	16.04.2025

В таблицы БД были добавлено более 10000 записей, там где это ВОЗМОЖНО.

## Индексы

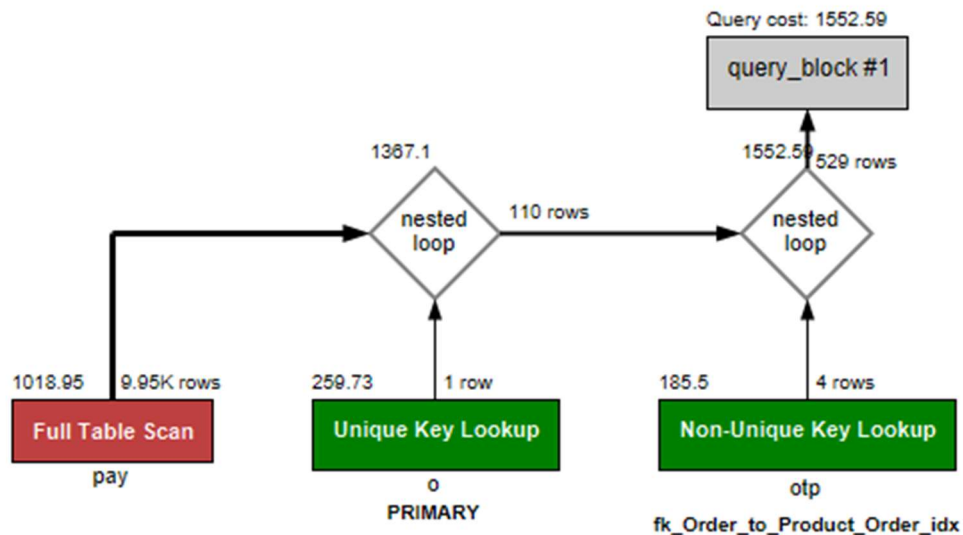
SELECT

```
SUM(pay.Cost) AS TotalRevenue,
COUNT(DISTINCT o.Order_number) AS OrderCount,
SUM(otp.Count) AS ProductCount
FROM `Order` o
JOIN Payment pay ON o.Order_number = pay.Order_Order_number
JOIN Order_to_Product otp ON o.Order_number = otp.Order_Order_number
WHERE o.Order_date BETWEEN '2024-01-01' AND '2025-01-01'
AND pay.Status = 'Оплачено';
```

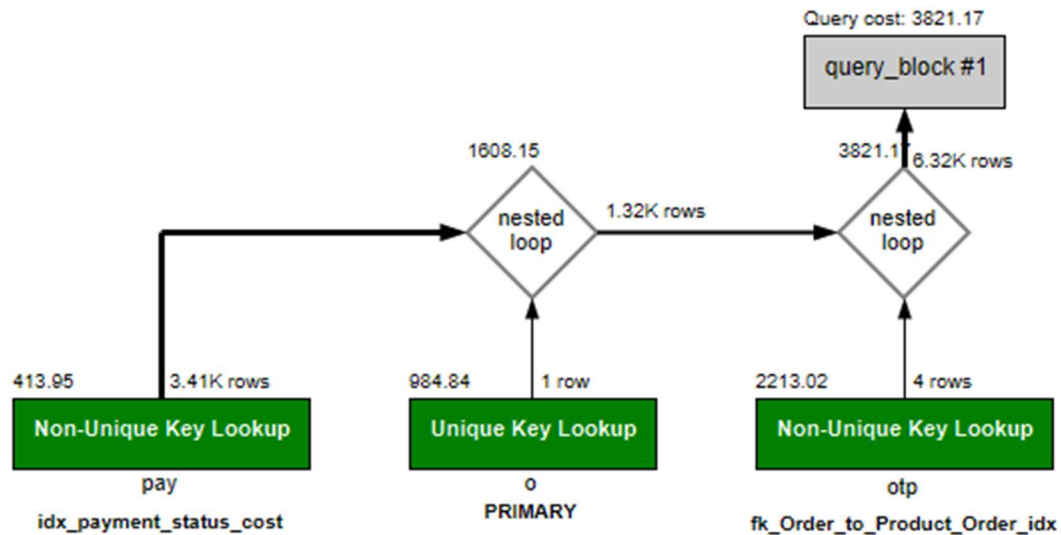
До:

```
CREATE INDEX idx_order_date ON `Order` (Order_date);
CREATE INDEX idx_payment_status_cost ON Payment (Status, Cost);
CREATE INDEX idx_otp_order ON Order_to_Product (Order_Order_number);
```

	id	select_type	table	partitions	type	possible_keys	key	key_len	ref	rows	filtered	Extra
►	1	SIMPLE	pay	INDEX	ALL	fk_Payment_Order_idx	INDEX		INDEX	9947	10.00	Using where
	1	SIMPLE	o	INDEX	eq_ref	PRIMARY	PRIMARY	4	online board game store.pay.Order_Order_num...	1	11.11	Using where
	1	SIMPLE	otp	INDEX	ref	fk_Order_to_Product_Order_idx	fk_Order_to_Product_Order_idx	4	online board game store.pay.Order_Order_num...	4	100.00	INDEX



После:

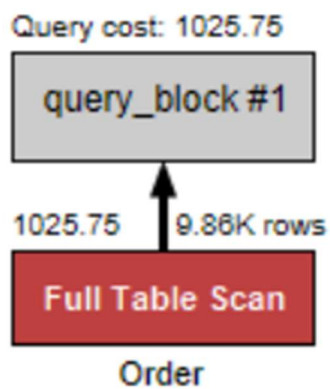


```
SELECT * FROM `Order`
WHERE Status IN ('В обработке', 'В пути');
```

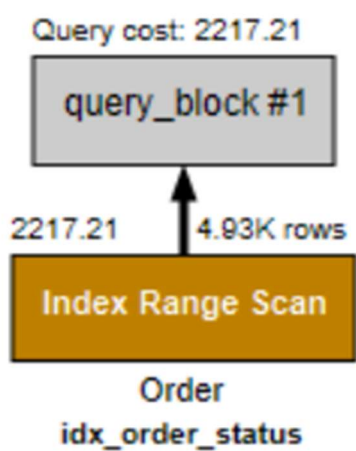
	id	select_type	table	partitions	type	possible_keys	key	key_len	ref	rows	filtered	Extra
►	1	SIMPLE	Order	<u>NULL</u>	ALL	<u>NULL</u>	<u>NULL</u>	<u>NULL</u>	<u>NULL</u>	9855	20.00	Using where

До:

```
CREATE INDEX idx_order_status ON `Order` (Status);
```



После:

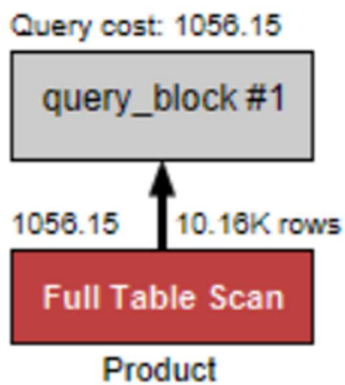


```
SELECT Title, Price FROM Product
WHERE Price BETWEEN 2000 AND 4000;
```

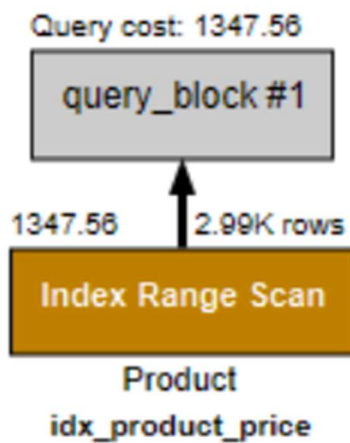
	id	select_type	table	partitions	type	possible_keys	key	key_len	ref	rows	filtered	Extra
►	1	SIMPLE	Product	NULL	ALL	NULL	NULL	NULL	NULL	10159	11.11	Using where

До:

```
CREATE INDEX idx_product_price ON Product(Price);
```



После:



2 часть

### Процедура 1: Обновление цены продукта с проверкой допустимости

```
DELIMITER //
CREATE PROCEDURE UpdateProductPrice(
    IN p_product_title VARCHAR(100),
    IN p_new_price MEDIUMINT,
    OUT p_status VARCHAR(100)
)
BEGIN
    DECLARE current_price MEDIUMINT;
    DECLARE price_change_percent DECIMAL(5,2);

    -- Получаем текущую цену продукта
```

```

SELECT Price INTO current_price FROM Product WHERE Title =
p_product_title;

-- Проверяем существует ли продукт
IF current_price IS NULL THEN
    SET p_status = 'Ошибка: Продукт не найден';
ELSE
    -- Рассчитываем процент изменения цены
    SET price_change_percent = ABS((p_new_price - current_price) /
current_price) * 100;

    -- Проверяем допустимость изменения цены (не более 30%)
    IF price_change_percent > 30 THEN
        SET p_status = CONCAT('Ошибка: Изменение цены на ',
price_change_percent, '% превышает допустимые 30%');
    ELSE
        -- Обновляем цену
        UPDATE Product SET Price = p_new_price WHERE Title =
p_product_title;
        SET p_status = CONCAT('Цена успешно обновлена с ',
current_price, ' до ', p_new_price);
    END IF;
END IF;
END //
DELIMITER ;

-- Вызов процедуры
CALL UpdateProductPrice('Монополия', 3790, @status);
SELECT @status;

```

	@status
►	Цена успешно обновлена с 2990 до 3790

## Процедура 2: Добавление продукта в избранное с проверкой дублирования

```

DELIMITER //
CREATE PROCEDURE AddToFavorites(
    IN p_buyer_login VARCHAR(60),

```

```

    IN p_product_title VARCHAR(100),
    OUT p_result VARCHAR(100)
)
BEGIN
    DECLARE v_favorites_id INT;
    DECLARE v_exists INT DEFAULT 0;

    -- Проверяем существует ли пользователь
    SELECT COUNT(*) INTO v_exists FROM Buyer WHERE Login = p_buyer_login;
    IF v_exists = 0 THEN
        SET p_result = 'Ошибка: Пользователь не найден';
    ELSE
        -- Проверяем существует ли продукт
        SELECT COUNT(*) INTO v_exists FROM Product WHERE Title =
p_product_title;
        IF v_exists = 0 THEN
            SET p_result = 'Ошибка: Продукт не найден';
        ELSE
            -- Получаем ID избранного списка пользователя
            SELECT id_favorites INTO v_favorites_id FROM Favorites WHERE
Buyer_Login = p_buyer_login;

            -- Проверяем, есть ли уже продукт в избранном
            SELECT COUNT(*) INTO v_exists FROM Product_to_Favorites
            JOIN Product ON Product.id_product =
Product_to_Favorites.Product_id_product
            WHERE Product.Title = p_product_title AND
Favorites_id_favorites = v_favorites_id;

            IF v_exists > 0 THEN
                SET p_result = 'Продукт уже в избранном';
            ELSE
                -- Добавляем продукт в избранное
                INSERT INTO Product_to_Favorites (Product.Title,
Favorites_id_favorites, Date_added)
                VALUES (p_product_title, v_favorites_id, CURDATE());

                -- Обновляем счетчик продуктов в избранном

```

```

UPDATE Favorites
SET Count_products = Count_products + 1,
    Update_date = CURDATE()
WHERE id_favorites = v_favorites_id;

SET p_result = 'Продукт успешно добавлен в избранное';
END IF;
END IF;
END IF;
END //
DELIMITER ;

```

-- Вызов процедуры

```

CALL AddToFavorites('sazuki', 'Каркассон', @result);
SELECT @result;

```

@result
Ошибка: Пользователь не найден

-- Вызов процедуры

```

CALL AddToFavorites('sazukil', 5, @result);
SELECT @result;

```

@result
Продукт успешно добавлен в избранное

-- Вызов процедуры

```

CALL AddToFavorites('sazukil', 5, @result);
SELECT @result;

```

@result
Продукт уже в избранном

## Создание купонов для покупателя

```

DELIMITER //
CREATE PROCEDURE GenerateDiscountCoupons(
    IN p_customer_count INT,
    IN p_discount_amount DECIMAL(5,2))
BEGIN
    DECLARE v_counter INT DEFAULT 1;

```

```

DECLARE v_coupon_code VARCHAR(10);
DECLARE v_customer_login VARCHAR(60);

-- Создаем таблицу для результатов
DROP TABLE IF EXISTS temp_coupons;
CREATE TABLE temp_coupons (
    coupon_id INT AUTO_INCREMENT PRIMARY KEY,
    customer_login VARCHAR(60),
    coupon_code VARCHAR(10),
    discount_amount DECIMAL(5,2),
    expiration_date DATE
);

-- Начинаем цикл
customer_loop: LOOP
    -- Проверяем условие выхода из цикла
    IF v_counter > p_customer_count THEN
        LEAVE customer_loop;
    END IF;

    -- Получаем случайного покупателя
    SELECT Login INTO v_customer_login
    FROM Buyer
    ORDER BY RAND()
    LIMIT 1;

    -- Генерируем случайный код купона (пример: DSCT-5A3B)
    SET v_coupon_code = CONCAT(SUBSTRING(MD5(RAND()),4,4), '-',
    FLOOR(RAND() * (9999 - 1000 + 1) + 1000));

    -- Добавляем купон в таблицу
    INSERT INTO temp_coupons (customer_login, coupon_code,
    discount_amount, expiration_date)
    VALUES (v_customer_login, v_coupon_code, p_discount_amount,
    DATE_ADD(CURDATE(), INTERVAL 30 DAY));

    -- Увеличиваем счетчик
    SET v_counter = v_counter + 1;

```



```

END LOOP;

-- Выводим результаты
SELECT
    customer_login AS 'Покупатель',
    coupon_code AS 'Код купона',
    CONCAT(discount_amount, '%') AS 'Размер скидки',
    expiration_date AS 'Годен до'
FROM temp_coupons;

SELECT CONCAT('Сгенерировано ', p_customer_count, ' купонов со
скидкой ', p_discount_amount, '%') AS Result;
END //
DELIMITER ;

-- Пример вызова процедуры
CALL GenerateDiscountCoupons(5, 15);
SELECT * FROM temp_coupons;

```

	coupon_id	customer_login	coupon_code	discount_amount	expiration_date
▶	1	oletar62	dfb0-6683	15.00	2025-05-16
	2	ivasmir88	6c8d-9748	15.00	2025-05-16
	3	vasorl12	6c5f-8912	15.00	2025-05-16
	4	nikitdmit10	29a6-8639	15.00	2025-05-16
	5	denkuzn40	0338-5730	15.00	2025-05-16
*	NULL	NULL	NULL	NULL	NULL

## Функция 1: Расчет скидочной цены продукта

```

DELIMITER //
CREATE FUNCTION GetDiscountedPrice(
    p_product_id INT
) RETURNS DECIMAL(10,2)
DETERMINISTIC
BEGIN
    DECLARE v_price MEDIUMINT;
    DECLARE v_discount_amount DECIMAL(5,2);
    DECLARE v_discounted_price DECIMAL(10,2);

    -- Получаем цену и размер скидки
    SELECT p.Price, IFNULL(d.Discount_amount, 0)

```

```

INTO v_price, v_discount_amount
FROM Product p
LEFT JOIN Discount d ON p.Discount_id_discount = d.id_discount
WHERE p.id_product = p_product_id;

-- Рассчитываем цену со скидкой
IF v_price IS NULL THEN
    RETURN NULL;
ELSE
    SET v_discounted_price = v_price * (1 - v_discount_amount / 100);
    RETURN ROUND(v_discounted_price, 2);
END IF;
END //
DELIMITER ;

-- Вызов функции
SELECT Title, Price, GetDiscountedPrice(id_product) AS DiscountedPrice
FROM Product
WHERE id_product = 208;

```

	Title	Price	DiscountedPrice
►	Академия Мир после ядерной войны политическая	2818	2536.20

## Функция 2: Проверка возрастного ограничения для покупателя

```

DELIMITER //
CREATE FUNCTION CheckAgeLimit(
    p_product_id INT,
    p_buyer_login VARCHAR(60)
) RETURNS VARCHAR(50)
READS SQL DATA
BEGIN
    DECLARE v_buyer_age INT;
    DECLARE v_age_limit INT;
    DECLARE v_result VARCHAR(50);

    -- Получаем возраст покупателя
    SELECT TIMESTAMPDIFF(YEAR, Birthday, CURDATE())

```

```

    INTO v_buyer_age
    FROM Buyer
    WHERE Login = p_buyer_login;

    -- Получаем возрастное ограничение продукта
    SELECT Age_limit
    INTO v_age_limit
    FROM Product
    WHERE id_product = p_product_id;

    -- Проверяем данные
    IF v_buyer_age IS NULL THEN
        SET v_result = 'Покупатель не найден';
    ELSEIF v_age_limit IS NULL THEN
        SET v_result = 'Продукт не найден';
    ELSEIF v_buyer_age >= v_age_limit THEN
        SET v_result = 'Доступ разрешен';
    ELSE
        SET v_result = CONCAT('Доступ запрещен. Возрастное ограничение:
', v_age_limit, '+');
    END IF;

    RETURN v_result;
END //
DELIMITER ;

-- Вызов функции
SELECT CheckAgeLimit(3, 'sazuki') AS AgeCheckResult;

```

	AgeCheckResult
►	Покупатель не найден

	AgeCheckResult
►	Доступ разрешен

### Функция 3: Расчет средней оценки продукта с обработкой NULL

```

DELIMITER //
CREATE FUNCTION GetProductRating(

```

```

        p_product_id INT
    ) RETURNS DECIMAL(3,2)
    READS SQL DATA
    BEGIN
        DECLARE v_avg_rating DECIMAL(3,2);
        DECLARE v_review_count INT;

        -- Получаем среднюю оценку и количество отзывов
        SELECT AVG(Score), COUNT(*)
        INTO v_avg_rating, v_review_count
        FROM Feedback
        WHERE Product_id_product = p_product_id;

        -- Обрабатываем случай, когда отзывов нет
        IF v_review_count = 0 THEN
            RETURN NULL;
        ELSE
            RETURN ROUND(v_avg_rating, 2);
        END IF;
    END //
DELIMITER ;

-- Вызов функции
SELECT
    Title,
    GetProductRating(id_product) AS AvgRating,
    (SELECT COUNT(*) FROM Feedback WHERE Product_id_product =
Product.id_product) AS ReviewCount
FROM Product
WHERE id_product IN (1, 3, 5);

```

	Title	AvgRating	ReviewCount
►	Монополия	NULL	0
	Колонизаторы	1.00	2
	Эволюция	NULL	0

```

SELECT
    Title,
    GetProductRating(id_product) AS AvgRating,

```

```

    (SELECT COUNT(*) FROM Feedback WHERE Product_id_product =
Product.id_product) AS ReviewCount
FROM Product
WHERE id_product IN (23, 3, 501);

```

	Title	AvgRating	ReviewCount
►	Колонизаторы	1.00	2
	Империя Ведьмачий мир боевик	4.00	2
	Лабиринт Средневековые хакерская	5.00	1

## Часть 3

### Представление 1: Активные пользователи с их статистикой

```
CREATE VIEW ActiveUsersStats AS
SELECT
    b.Login,
    CONCAT(b.Firstname, ' ', b.Lastname) AS FullName,
    COUNT(DISTINCT o.Order_numder) AS OrderCount,
    SUM(pay.Cost) AS TotalSpent,
    COUNT(DISTINCT f.id_feedback) AS FeedbackCount,
    COUNT(DISTINCT fav.id_favorites) AS FavoritesCount
FROM Buyer b
LEFT JOIN `Order` o ON b.Login = o.Buyer_Login
LEFT JOIN Payment pay ON o.Order_numder = pay.Order_Order_numder AND
pay.Status = 'Оплачено'
LEFT JOIN Feedback f ON b.Login = f.Buyer_Login
LEFT JOIN Favorites fav ON b.Login = fav.Buyer_Login
GROUP BY b.Login, FullName
HAVING OrderCount > 0 OR FeedbackCount > 0 OR FavoritesCount > 0;

-- Запрос к представлению
SELECT * FROM ActiveUsersStats ORDER BY TotalSpent DESC;
```

	Login	FullName	OrderCount	TotalSpent	FeedbackCount	FavoritesCount
▶	ksevin30	Ксения Виноградов	2	243669	3	1
	markuz48	Маргарита Кузнецов	3	145766	2	1
	marpor95	Маргарита Попов	1	130938	2	1
	anapav46	Анастасия Павлов	2	90973	0	1
	makego36	Максим Егоров	1	79598	2	1
	vlavol76	Владимир Волков	3	64431	1	1
	levbel52	Лев Белов	3	63672	1	1
	ekarom52	Екатерина Романов	1	59301	0	1
	levili57	Лев Ильин	1	58992	0	0
	pavsmi15	Павел Смирнов	4	57628	1	1
	igonov45	Игорь Новиков	1	48090	1	1

### Представление "Товары с категориями и производителями"

```
CREATE VIEW ProductCatalog AS
SELECT
```

```
    p.id_product AS ID,
```

```

    p.Title AS ProductName,
    p.Price,
    p.Age_limit AS AgeLimit,
    m.Name AS Manufacturer,
    GROUP_CONCAT(DISTINCT c.Title SEPARATOR ', ') AS Categories
FROM
    Product p
JOIN
    Maker m ON p.Maker_id_maker = m.id_maker
LEFT JOIN
    Product_to_Category ptc ON p.id_product = ptc.Product_id_product
LEFT JOIN
    Category c ON ptc.Category_id_category = c.id_category
GROUP BY
    p.id_product, p.Title, p.Price, p.Age_limit, m.Name;

SELECT * FROM ProductCatalog ORDER BY Price DESC;

```

	ID	ProductName	Price	AgeLimit	Manufacturer	Categories
►	5074	Ковчег Каменный век магическая	10890	16	Ludicreations	Головоломки, Приключенческие игры
	726	Путешествие Подводный мир рогалик	10888	6	Gigamic	Военные игры, Фэнтези
	4508	Постапокалипсис политическая	10887	18	Ludicreations	Ролевые игры
	8942	Изгнание Древняя Русь динозавры	10871	12	Lautapelit.fi	Музыкальные игры
	6220	Подводная колония гонки	10868	7	Zvezda	Детские игры, Постапокалипсис
	1829	Падение Дикий Запад	10861	12	Queen Games	Исторические игры, Партийные игры
	9038	Древний мир ритм-игра	10831	8	Matagot	Фэнтези
	537	Средневековая Европа научно-фантастичес...	10741	6	Moaideas Game Design	Исторические игры, Музыкальные игры
	909	Космос реалисти	10712	6	Portal Games	Научная фантастика, Семейные игры
	2050	Подводная колония релакс	10669	12	Hobby World	Семейные игры

## 2. Представление "Активные заказы с деталями"

```

CREATE VIEW ActiveOrders AS
SELECT
    o.Order_numder AS OrderID,
    o.Order_date AS OrderDate,
    CONCAT(b.Lastname, ' ', b.Firstname) AS Buyer,
    o.Delivery_method AS DeliveryMethod,
    o.Status AS OrderStatus,
    COUNT(otp.Product_id_product) AS ProductCount,
    SUM(p.Price * otp.Count) AS TotalAmount
FROM
    `Order` o
JOIN

```

```

    Buyer b ON o.Buyer_Login = b.Login
JOIN
    Order_to_Product otp ON o.Order_numder = otp.Order_Order_numder
JOIN
    Product p ON otp.Product_id_product = p.id_product
WHERE
    o.Status NOT IN ('Доставлен', 'Отменен')
GROUP BY
    o.Order_numder, o.Order_date, b.Lastname, b.Firstname,
    o.Delivery_method, o.Status;

SELECT * FROM ActiveOrders WHERE OrderDate > DATE_SUB(CURDATE(), INTERVAL
30 DAY);

```

	OrderID	OrderDate	Buyer	DeliveryMethod	OrderStatus	ProductCount	TotalAmount
►	2503201416	2025-03-20	Егоров Максим	Самовывоз	В обработке	10	39799
	2503237936	2025-03-23	Сергеев Вадим	Почта	В пути	5	26817

## Представление "Популярные товары с рейтингами"

```

CREATE VIEW PopularProducts AS
SELECT
    p.id_product AS ProductID,
    p.Title AS ProductName,
    p.Price,
    AVG(f.Score) AS AvgRating,
    COUNT(DISTINCT f.id_feedback) AS ReviewCount,
    COUNT(DISTINCT otp.Order_Order_numder) AS OrderCount,
    COUNT(DISTINCT ptf.Favorites_id_favorites) AS FavoriteCount,
    (COUNT(DISTINCT otp.Order_Order_numder) * 2 +
    COUNT(DISTINCT ptf.Favorites_id_favorites) +
    COUNT(DISTINCT f.id_feedback)) AS PopularityScore
FROM
    Product p
LEFT JOIN
    Feedback f ON p.id_product = f.Product_id_product
LEFT JOIN
    Order_to_Product otp ON p.id_product = otp.Product_id_product
LEFT JOIN

```



```

Product_to_Favorites ptf ON p.id_product = ptf.Product_id_product
GROUP BY
    p.id_product, p.Title, p.Price
HAVING
    ReviewCount > 0 OR OrderCount > 0 OR FavoriteCount > 0;

SELECT * FROM PopularProducts ORDER BY PopularityScore DESC LIMIT 10;

```

	ProductID	ProductName	Price	AvgRating	ReviewCount	OrderCount	FavoriteCount	PopularityScore
►	2289	Тень Затерянные миры боевик	3820	3.5000	2	18	11	49
	1439	Историческая социальная	4272	3.0000	1	18	10	47
	3790	Война будущего стратегическая	755	3.0000	1	13	17	44
	3962	Индустриальная революция текстовая	4443	4.0000	1	14	15	44
	1671	След Мир гигантов	4351	5.0000	1	14	15	44
	5624	Скиталец Средневековые	3582	4.0000	4	9	22	44
	4973	Средневековая Африка приключенческая	1253	NULL	0	13	18	44
	5765	Врата Средневековая Азия	1196	NULL	0	13	17	43
	9534	Храм Футуристический спорт музыкальная	3416	4.0000	1	10	22	43
	8089	Клятва Атлантида	1683	4.3333	3	13	14	43

```

SELECT
    ProductName,
    Price,
    AvgRating
FROM PopularProducts
WHERE AvgRating >= 4.5
ORDER BY Price;

```

	ProductName	Price	AvgRating
►	Руны Парк Юрского периода строительная	514	5.0000
	Клинок Индустриальная революция	519	5.0000
	Чародей Заброшенная станция боевик	526	5.0000
	Атлантида тактическая	528	4.6667
	Клятва Индустриальная революция абстракт...	541	5.0000
	Древние скандинавы семейная	549	5.0000
	Путешествие Средневековая Африка	554	5.0000
	Мгла Война будущего	554	5.0000
	Котел Мир гигантов	558	5.0000
	Война будущего экономическая	561	4.5000
	Академия Средневековая Европа карточная	563	5.0000
	Мифология подземелья	571	5.0000
	Гильдия Космос	573	5.0000
	Падение Постапокалиптический вайлдленд ...	575	5.0000
	Дальний Восток песочница	581	5.0000
	Викторианская эпоха гиперказуальная	583	5.0000

## Часть 4

### Триггер: Обновление счетчика продуктов в избранном

```
DELIMITER //
CREATE TRIGGER after_favorites_update
AFTER INSERT ON Product_to_Favorites
FOR EACH ROW
BEGIN
    UPDATE Favorites
    SET Count_products = (
        SELECT COUNT(*)
        FROM Product_to_Favorites
        WHERE Favorites_id_favorites = NEW.Favorites_id_favorites
    ),
    Update_date = CURDATE()
    WHERE id_favorites = NEW.Favorites_id_favorites;
END //
DELIMITER ;
```

-- Тестирование триггера

```
INSERT INTO Product_to_Favorites (Date_added, Product_id_product,
Favorites_id_favorites)
VALUES (CURDATE(), 111, 1);
```

-- Проверка обновления

```
SELECT * FROM Favorites WHERE id_favorites = 1;
```

	id_favorites	Count_products	Update_date	Buyer_Login
▶	1	9	2025-04-16	makpet35
*	NULL	NULL	NULL	NULL

-- Тестирование триггера

```
INSERT INTO Product_to_Favorites (Date_added, Product_id_product,
Favorites_id_favorites)
```

```
VALUES (CURDATE(), 113, 1);
```

```
-- Проверка обновления
```

```
SELECT * FROM Favorites WHERE id_favorites = 1;
```

	id_favorites	Count_products	Update_date	Buyer_Login
►	1	10	2025-04-16	makpet35
*	NULL	NULL	NULL	NULL