Министерство науки и высшего образования РФ

Федеральное государственное автономное образовательное

учреждениевысшего образования

«Казанский (Приволжский) федеральный университет»

Институт Вычислительной математики и информационных технологий

Кафедра системного анализа и информационных технологий

**ОТЧЕТ ПО КУРСУ «ТЕХНОЛОГИИ БАЗ ДАННЫХ»**

Разработка базы данных «Shop» и клиентского приложения для доступа к ней

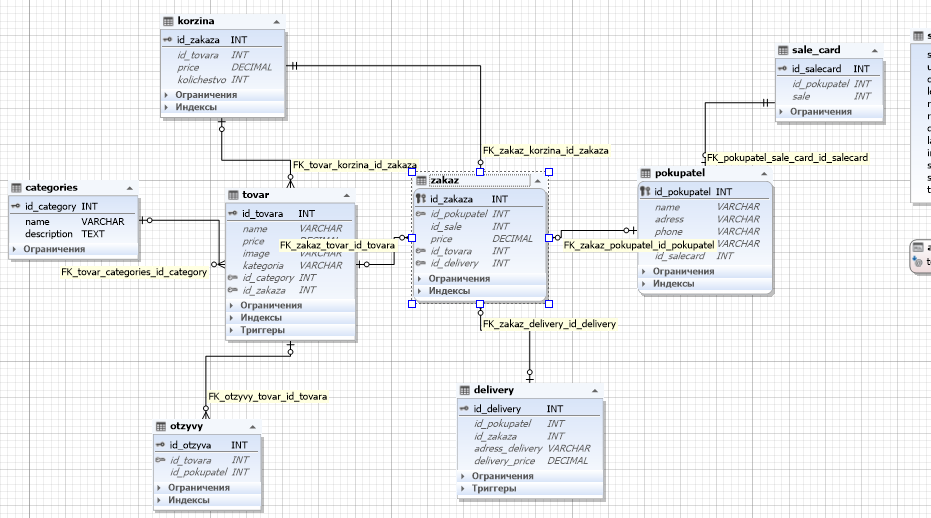
Студентка 3 курса

Группа 09-631

Алимова Сабина

Казань-2018

1. **Описание реляционной модели данных.**



Всего 8 таблиц:

1. **tovar** (товар) содержит id товара, название товара, цену за один товар, рисунок, категорию,id категории,id заказа.
2. **zakaz** (заказ) содержит id покупателя, id заказа, id скидки, цена товара, id товара ,id доставки.
3. **pokupatel** (покупатель) содержит id покупателя, имя (покупателя), адрес(покупателя), номер телефона (покупателя),e-mail (покупателя),id скидочной карты (покупателя).
4. **categories** (категории) содержит id категории, название категории, описание.
5. **otzyvy** (отзывы) содержит id отзыва, id товара, id покупателя
6. **delivery** (доставка) содержит id доставки, id покупателя, id заказа, стоимость доставки, адрес доставки.
7. **sale\_card** ( скидочная карта) содержит id скидочной карты, id клиента, скидка.
8. **korzina** (корзина) содержит id заказа, id товара, цену, количество товара
9. **Запросы к базе данных.**

* *Проекция*

1. SELECT c.id\_category, c.name FROM categories c; — вывести id категории, имя категории
2. SELECT l.name, l.adress, l.phone FROM pokupatel l; — вывести имя покупателя, адрес и телефон
3. SELECT t.name, t.price, t.kategoria FROM tovar t; — вывели имя товара, цену и категорию

* *Селекция*

1. SELECT \* FROM pokupatel p WHERE p.name ='Сабина';

— выводим информацию по покупателю с необходимым именем

1. SELECT \* FROM tovar t WHERE t.price < 1000; — выводим информацию о товаре, цена которого меньше 1000
2. SELECT \* FROM categories c WHERE c.name = 'шары';

— выводим информацию о категории шары

* *Объединение*

1. SELECT t.id\_tovara, t.price FROM tovar t UNION SELECT k.id\_tovara, k.price FROM korzina k; — объеденили таблиц товра и корзина. Включает в себя все кортежи первого и второго. Повторяющиеся строки исключены
2. SELECT t.id\_category, t.name FROM tovar t UNION SELECT c.id\_category, c.name FROM categories c; — объединение таблиц товар и категория. Потворяющиеся строки исключены
3. SELECT o.id\_tovara, o.id\_otzyva FROM otzyvy o UNION SELECT t.id\_tovara, t.name FROM tovar t; — объединение таблиц отзыв и товары

* *Пересечение*

1. SELECT k.id\_zakaza, t.name, k.price FROM korzina k, tovar t WHERE k.id\_tovara = t.id\_tovara; — поиск пересечения между товарами и заказами.
2. SELECT p.name,z.id\_zakaza, z.price, p.adress FROM zakaz z, pokupatel p WHERE z.id\_pokupatel = p.id\_pokupatel AND z.price> 1400; — нашли покупателей, сделавших заказы дороже 1400
3. SELECT \* FROM delivery d, zakaz z WHERE d.id\_zakaza = z.id\_zakaza AND d.delivery\_price < 1400; — нашли доставки, которые обошлись меньше 1400

* *Разность*

1. SELECT \* FROM tovar t WHERE NOT EXISTS(SELECT \* FROM tovar

t1 WHERE t.kategoria ='Малый шар'); — выводим все товары категории кроме малый шар

1. SELECT \* FROM delivery d WHERE NOT EXISTS (SELECT \* FROM delivery d1 WHERE d.delivery\_price < 1500); — получили все доставки, которые не меньше 1500
2. SELECT \* FROM korzina k WHERE NOT EXISTS(SELECT \* FROM korzina k1 WHERE k.price > 600) ORDER BY k.price; — вывели цены на товары из корзины, котоыре меньше 600. Упорядочили по цене

* *Декартово произведение*

1. SELECT DISTINCT t.name, t.price FROM otzyvy o CROSS JOIN tovar t WHERE o.id\_pokupatel = 1 GROUP BY t.name;

захотели посмотреть, на какие товары оставил отзыв пользователь, с id = 1.

1. SELECT t.name, COUNT(t.id\_tovara) AS 'КОЛИЧЕСТВО ОТЗЫВОВ О ТОВАРЕ' FROM tovar t INNER JOIN otzyvy o ON t.id\_tovara = o.id\_tovara WHERE o.id\_tovara = 1;

выводим имя товара и количество отзывов на этот товар. Выбор товара осуществляется по ID

1. SELECT p.name, p.adress, p.phone FROM zakaz z CROSS JOIN pokupatel p WHERE z.id\_pokupatel = p.id\_pokupatel;

вывели покупателей, сделавших заказ

* *inner join. Внутреннее соединение*

а)SELECT d.id\_delivery, z.price, d.adress\_delivery FROM zakaz z INNER JOIN delivery d ON z.id\_zakaza = d.id\_zakaza; — выводим id доставки, цену и адрес.

b)SELECT t.name, t.kategoria, k.price FROM tovar t INNER JOIN korzina k ON t.id\_tovara = k.id\_tovara; — выводим название товара, категорию и общую сумму

c)SELECT t.name,t.price,o.id\_otzyva FROM otzyvy o INNER JOIN tovar t ON t.id\_tovara = o.id\_tovara; — вывод названия, цены и ид отзыва

* *Группировки*

a) SELECT DISTINCT [t.name](https://vk.com/away.php?to=http%3A%2F%2Ft.name&cc_key=), t.price FROM otzyvy o CROSS JOIN tovar t WHERE [o.id](https://vk.com/away.php?to=http%3A%2F%2Fo.id&cc_key=" \t "_blank)\_pokupatel = 1 GROUP BY [t.name](https://vk.com/away.php?to=http%3A%2F%2Ft.name&cc_key=" \t "_blank);

захотели посмотреть, на какие товары оставил отзыв пользователь, с id = 1. DISTINCT - убирает повторы. операция группировки

1. **Серверные процедуры и функции.**

**Триггеры:**

* *триггер на удаление доставки. Перед тем, как удалить доставку, необходимо удалить заказ.*

*CREATE TRIGGER on\_delete\_del*

*BEFORE DELETE*

*ON delivery FOR EACH ROW*

*BEGIN*

*DELETE FROM zakaz WHERE id\_zakaza = OLD.id\_zakaza;*

*END*

*CALL on\_delete\_del ('srg')*

*DELETE FROM delivery WHERE id\_delivery = 1;*

* *триггер на удаление товара. Если удаляем товар, то удаляем заказ и отзыв.*

*SELECT \* FROM tovar t;*

*CREATE TRIGGER on\_delete\_tovar*

*BEFORE DELETE*

*ON tovar FOR EACH ROW*

*BEGIN*

*DELETE FROM zakaz WHERE id\_zakaza = OLD.id\_zakaza;*

*DELETE FROM otzyvy WHERE id\_tovara = OLD.id\_tovara;*

*END*

*SELECT \* FROM delivery d;*

**Процедуры:**

* Вводим id пользователя, и находим стоимость всех его покупок.

*CREATE PROCEDURE total\_sum( ID\_pok int)*

*BEGIN*

*SELECT SUM(price) AS 'Общая стоймость покупок пользователя' FROM zakaz z WHERE z.id\_pokupatel = ID\_pok;*

*END*

*call total\_sum(1);*

* поиск количества отзывов о товаре с именем

*CREATE PROCEDURE ans\_quant(tovar\_name varchar(100))*

*BEGIN*

*DECLARE d1 int; //*переменная для харенения количество отзывов о товаре

*SELECT COUNT(t.id\_tovara) INTO d1 FROM tovar t INNER JOIN otzyvy o ON t.id\_tovara = o.id\_tovara WHERE t.name = tovar\_name;*

*SET @answer = d1*; //присваиваем в ответ

*END;*

//вызов

*CALL ans\_quant('srg')*

*SELECT @answer*