

Санкт-Петербургский Государственный Политехнический
Университет Петра Великого
Институт Компьютерных Наук и Технологий
Кафедра Компьютерных Систем и Программных Технологий

Отчёт по лабораторной работе
Дисциплина: Базы данных
Тема: Ознакомление с основами SQL-DML

Выполнил студент группы 43501/3

(подпись) Круминьш Д.В.

Преподаватель

(подпись) Мяснов А.В.

Программа работы

1. Изучите SQL-DML
2. Выполните все запросы из списка стандартных запросов. Продемонстрируйте результаты преподавателю.
3. Получите у преподавателя и реализуйте SQL-запросы в соответствии с индивидуальным заданием. Продемонстрируйте результаты преподавателю.
4. Выполненные запросы SELECT сохраните в БД в виде представлений, запросы INSERT, UPDATE или DELETE – в виде ХП. Выложите скрипт в Subversion.

Список стандартных запросов

- Сделайте выборку всех данных из каждой таблицы
- Сделайте выборку данных из одной таблицы при нескольких условиях, с использованием логических операций, LIKE, BETWEEN, IN (не менее 3-х разных примеров)
- Создайте в запросе вычисляемое поле
- Сделайте выборку всех данных с сортировкой по нескольким полям
- Создайте запрос, вычисляющий несколько совокупных характеристик таблиц
- Сделайте выборку данных из связанных таблиц (не менее двух примеров)
- Создайте запрос, рассчитывающий совокупную характеристику с использованием группировки, наложите ограничение на результат группировки
- Придумайте и реализуйте пример использования вложенного запроса
- С помощью оператора INSERT добавьте в каждую таблицу по одной записи
- С помощью оператора UPDATE измените значения нескольких полей у всех записей, отвечающих заданному условию
- С помощью оператора DELETE удалите запись, имеющую максимальное (минимальное) значение некоторой совокупной характеристики
- С помощью оператора DELETE удалите записи в главной таблице, на которые не ссылается подчиненная таблица (используя вложенный запрос)

Ход работы

По итогам предыдущей работы имеется следующая SQL - схема:

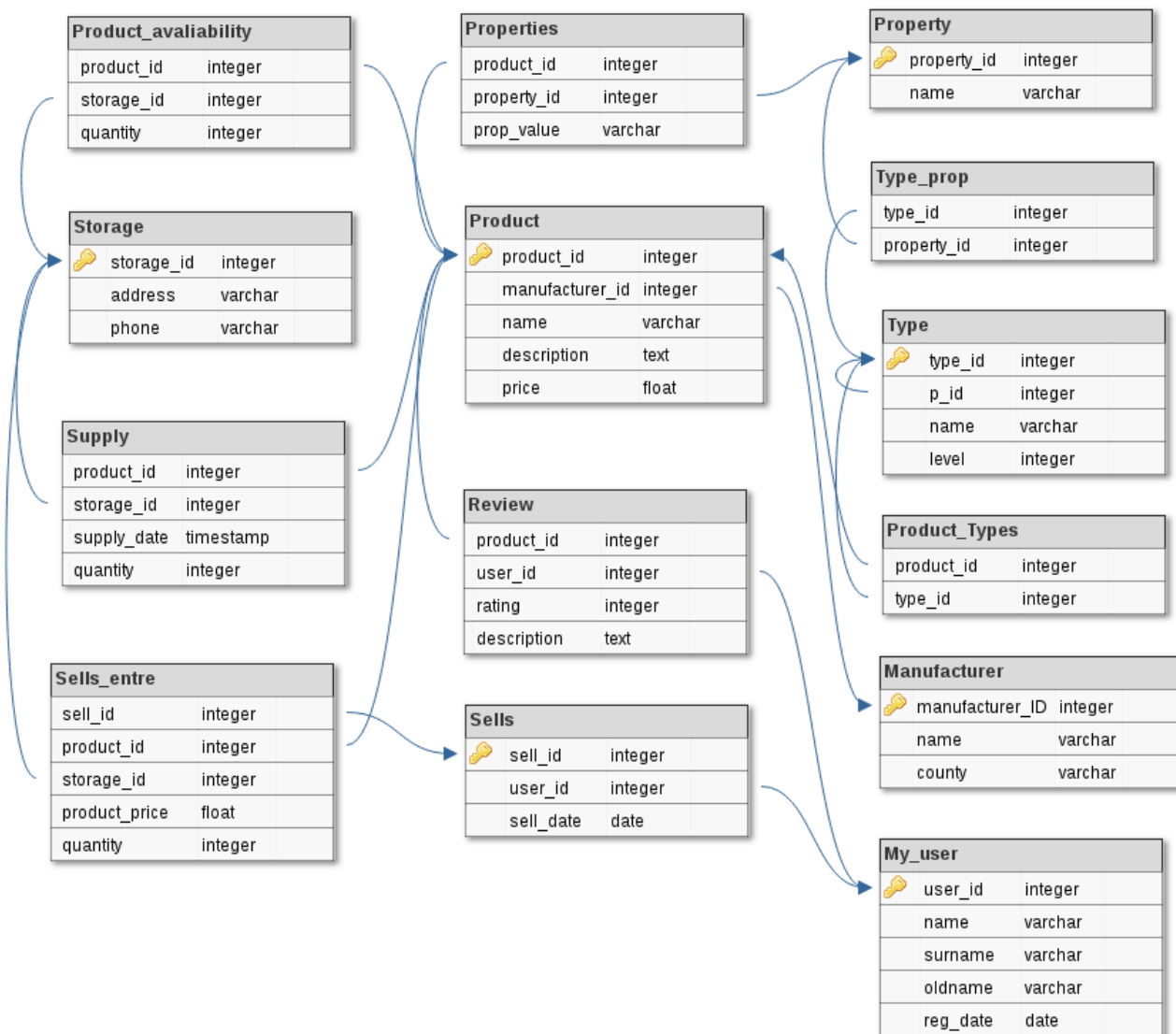


Рис. 1: SQL-схема БД

1 Сделайте выборку всех данных из каждой таблицы

```

1 SELECT * FROM MANUFACTURER;
2 SELECT * FROM MY_USER;
3 SELECT * FROM PRODUCT;
4 SELECT * FROM PRODUCT_AVALIABILITY;
5 SELECT * FROM PRODUCT_TYPES;
6 SELECT * FROM PROPERTIES;
7 SELECT * FROM PROPERTY;
8 SELECT * FROM REVIEW;
9 SELECT * FROM SELLS;
10 SELECT * FROM SELLS_ENTRE;
11 SELECT * FROM STORAGE;
12 SELECT * FROM SUPPLY;
13 SELECT * FROM TYPE;
14 SELECT * FROM TYPE_PROP;

```

Листинг 1: Запрос на выборку всех данных из таблиц

2 Сделайте выборку данных из одной таблицы при нескольких условиях, с использованием логических операций, LIKE, BETWEEN, IN (не менее 3-х разных примеров)

Вывод продукта, название которого начинается с AMD.

```
1 SELECT * FROM PRODUCT WHERE NAME LIKE 'AMD%';
```

Листинг 2: Выборка с использованием LIKE

Вывод продуктов, цены которых в указанном интервале.

```
1 SELECT * FROM PRODUCT WHERE PRICE BETWEEN 31998 AND 32000;
```

Листинг 3: Выборка с использованием BETWEEN

Вывод продуктов, цены которых соответствуют какому-либо значению из скобок.

```
1 SELECT * FROM PRODUCT WHERE PRICE IN (31998, 32000, 75448);
```

Листинг 4: Выборка с использованием IN

3 Создайте в запросе вычисляемое поле

Вычисляется дополнительное поле цены, которое показывает цену с 5% скидкой.

```
1 SELECT PRODUCT_ID, PRODUCT.PRICE, (PRICE*95/100) AS Discount_Price  
   ↪ FROM PRODUCT;
```

Листинг 5: Выборка продуктов

Объединение инициалов в одно поле.

```
1 SELECT USER_ID, (SURNAME || ' ' || NAME || ' ' || OLDNAME) AS FIO FROM  
   ↪ MY_USER;
```

Листинг 6: Выборка пользователей

4 Сделайте выборку всех данных с сортировкой по нескольким полям

```
1 SELECT * FROM MY_USER ORDER BY NAME, REG_DATE;
```

Листинг 7: Сортировка пользователей по имени и дате регистрации

```
1 SELECT * FROM PRODUCT ORDER BY PRICE DESC, NAME;
```

Листинг 8: Обратная сортировка продуктов по цене, затем по имени

5 Создайте запрос, вычисляющий несколько совокупных характеристик таблиц

Вывод среднего, минимального и максимального значения цены среди всех продуктов.

```
1 SELECT AVG(PRICE), MIN(PRICE), MAX(PRICE) FROM PRODUCT;
```

Листинг 9: Совокупные характеристика продуктов

6 Сделайте выборку данных из связанных таблиц (не менее двух примеров)

```
1 SELECT PRODUCT_ID, MY_USER.USER_ID, (MY_USER.SURNAME || ' ' || MY_USER.  
  ↳ NAME || ' ' || MY_USER.OLDNAME ) AS FIO, RATING, DESCRIPTION FROM  
  ↳ MY_USER JOIN REVIEW ON MY_USER.USER_ID=REVIEW.USER_ID ORDER BY  
  ↳ USER_ID;
```

Листинг 10: Выборка пользователей и всех их комментариев

```
1 SELECT PRODUCT_ID, PRODUCT.NAME as PRODUCT_NAME, MANUFACTURER.name AS  
  ↳ MANUFACTURER_NAME FROM PRODUCT JOIN MANUFACTURER ON PRODUCT.  
  ↳ MANUFACTURER_ID=MANUFACTURER.MANUFACTURER_ID ORDER BY PRODUCT_ID  
  ↳ ;
```

Листинг 11: Выборка продуктов и их производителей

7 Создайте запрос, рассчитывающий совокупную характеристику с использованием группировки, наложите ограничение на результат группировки

```
1 SELECT USER_ID, COUNT(USER_ID) from REVIEW GROUP by USER_ID having  
  ↳ COUNT(USER_ID) > 1;
```

Листинг 12: Выборка пользователей у которых более одного комментария

8 Придумайте и реализуйте пример использования вложенного запроса

```
1 SELECT * FROM PRODUCT WHERE MANUFACTURER_ID=(select MANUFACTURER_ID  
  ↳ FROM MANUFACTURER WHERE NAME="СКАЙНЕТ");
```

Листинг 13: Выборка продуктов производитель которых СКАЙНЕТ

9 С помощью оператора INSERT добавьте в каждую таблицу по одной записи

```
1 insert INTO MANUFACTURER(MANUFACTURER_ID,NAME,COUNTRY) VALUES(4, 'JP',  
  ↳ 'JAPAN');  
2 insert INTO MY_USER(USER_ID,NAME,SURNAME,OLDNAME,REG_DATE) VALUES  
  ↳ (100002,'Иван','Петров','Валентинович','11.02.2014');  
3 insert INTO PRODUCT(PRODUCT_ID,MANUFACTURER_ID,NAME,DESCRIPTION, PRICE  
  ↳ ) VALUES(100002,4,'Процессор x552','Ультра быстрый процессор',  
  ↳ 19999.9);  
4 insert INTO PRODUCT_AVALIABILITY(PRODUCT_ID,STORAGE_ID,QUANTITY)  
  ↳ VALUES(100002,1,20);  
5 insert INTO PRODUCT_TYPES(PRODUCT_ID,TYPE_ID) VALUES(100002,5);  
6 insert INTO PROPERTIES(PRODUCT_ID,PROPERTY_ID,PROP_VALUE) VALUES  
  ↳ (100002,2,'XMA 931');  
7 insert INTO PROPERTY(PROPERTY_ID,NAME) VALUES(4,'Тип памяти');  
8 insert INTO REVIEW(PRODUCT_ID, USER_ID, RATING, DESCRIPTION) VALUES  
  ↳ (100002,100002,5,'Лучший товар');  
9 insert INTO SELLS(SELL_ID,USER_ID,SELL_DATE) VALUES(2,100002,'  
  ↳ 22.6.2013');
```

```

10 insert INTO SELLS_ENTRE(SELL_ID,PRODUCT_ID,STORAGE_ID,PRODUCT_PRICE,
    ↪ QUANTITY) VALUES(2,100002,1,19999.9,1);
11 insert INTO STORAGE(STORAGE_ID,ADDRESS,PHONE) VALUES(2,'ул. Пушкина, д
    ↪ ом 44','8(909)331-89-49');
12 insert INTO SUPPLY(PRODUCT_ID,STORAGE_ID,SUPPLY_DATE, QUANTITY) VALUES
    ↪ (100002,2,'10.10.2010',21);
13 insert INTO TYPE(TYPE_ID,P_ID,NAME,LEVEL) VALUES(9,7,'Легковесный',3);
14 insert INTO TYPE_PROP(TYPE_ID,PROPERTY_ID) VALUES(7,1);

```

Листинг 14: Добавление в каждую из таблиц по одной записи

10 С помощью оператора UPDATE измените значения нескольких полей у всех записей, отвечающих заданному условию

```

1 UPDATE PRODUCT SET PRICE=PRICE+1;

```

Листинг 15: Едичный инкремент цены каждого продукта

11 С помощью оператора DELETE удалите запись, имеющую максимальное (минимальное) значение некоторой совокупной характеристики

```

1 insert INTO PRODUCT(PRODUCT_ID,MANUFACTURER_ID,NAME,DESCRIPTION, PRICE
    ↪ ) VALUES(100003,4,'Тест','Тест', 999999.9);
2 DELETE FROM product WHERE PRODUCT_ID=(SELECT PRODUCT_ID FROM product
    ↪ WHERE PRICE=(SELECT MAX(PRICE) FROM PRODUCT));

```

Листинг 16: Добавление и удаление продукта с максимальной ценой

12 С помощью оператора DELETE удалите записи в главной таблице, на которые не ссылается подчиненная таблица (используя вложенный запрос)

```

1 DELETE FROM PRODUCT where PRODUCT_ID IN(
2 SELECT PRODUCT_ID FROM PRODUCT WHERE PRODUCT_ID NOT IN(SELECT
    ↪ PRODUCT_ID FROM review)
3 AND PRODUCT_ID NOT IN(select PRODUCT_ID from PRODUCT_AVALIABILITY)
4 AND PRODUCT_ID NOT IN(SELECT PRODUCT_ID from SUPPLY)
5 AND PRODUCT_ID NOT IN (SELECT PRODUCT_ID FROM SELLS_ENTRE)
6 AND PRODUCT_ID NOT IN(SELECT PRODUCT_ID FROM PRODUCT_TYPES));

```

Листинг 17: Удаление записи на которую не ссылаются подчиненные таблицы

Индивидуальное задание

1 Выбрать уникальные значения определенной характеристики для заданной категории товаров.

```

1 SELECT PROPERTIES.PROP_VALUE FROM PROPERTIES
2 WHERE PROPERTY_ID IN(SELECT PROPERTY_ID FROM TYPE_PROP WHERE TYPE_ID
   ↪ =10)
3 AND PROPERTY_ID IN(2)
4 GROUP BY PROPERTIES.PROP_VALUE;

```

Листинг 18: Выборка уникальных характеристик при заданных значениях

2 Выбрать все товары заданной категории с учетом подкатегорий.

Решение в лоб:

```

1 SELECT PRODUCT.* from PRODUCT
2 JOIN product_types ON PRODUCT.PRODUCT_ID=PRODUCT_TYPES.product_id
3 JOIN type ON PRODUCT_TYPES.TYPE_ID=TYPE.TYPE_ID
4 WHERE PRODUCT_TYPES.type_id IN (
5 SELECT TYPE_ID FROM TYPE WHERE P_ID IN
6 (SELECT TYPE_ID from TYPE WHERE P_ID in
7 (1))) ;

```

Листинг 19: Выборка товаров заданной категории и подкатегорий

Решение используя рекурсию:

```

1 with recursive RR as(
2     select * from TYPE Where     TYPE_ID=1
3     union all
4     select type.* FROM TYPE, RR where TYPE.P_ID=RR.TYPE_ID
5 )
6 select PRODUCT.* FROM product
7 JOIN product_types ON PRODUCT.PRODUCT_ID=PRODUCT_TYPES.product_id
8 where type_id in
9 (SELECT type_id FROM RR);

```

Листинг 20: Выборка товаров заданной категории и подкатегорий

3 Вычислить отношение среднего рейтинга товара к объему его продаж для заданной категории в течение заданного периода.

Выборка айди товаров, среднего рейтинга, объема продаж, отношение рейтинга к объему продаж, при учетывании категории и её подкатегорий и заданного промежутка времени.

```

1 with recursive RR as(
2     select * from TYPE Where     TYPE_ID=1
3     union all
4     select type.* FROM TYPE, RR where TYPE.P_ID=RR.TYPE_ID
5 )
6 select PRODUCT.product_id,
7 avg(CAST(RATING as Float)) AS AVG_RATING,
8 sum(product_price) as total_sum,
9 (10000*(avg(CAST(RATING as Float))/sum(product_price) )) AS RATIO
10
11 FROM PRODUCT

```

```

12 JOIN REVIEW ON PRODUCT.PRODUCT_ID=REVIEW.product_id
13 JOIN product_types ON PRODUCT.PRODUCT_ID=PRODUCT_TYPES.product_id
14 JOIN SELLS_ENTRE ON PRODUCT.PRODUCT_ID=SELLS_ENTRE.PRODUCT_ID
15 JOIN sells ON SELLS_ENTRE.SELL_ID=SELLS.SELL_ID
16 where PRODUCT_TYPES.type_id in
17 (SELECT type_id FROM RR)
18 and SELL_DATE between date '10.09.2012' and date '23.06.2013'
19 group by product.product_id;

```

Листинг 21: Выборка согласно требованиям

Примечание

Пришлось сменить SQL dialect с 1 на 3, так как в первом случае команда between для двух промежутков времени не работала.

Вывод

В ходе данной работы я познакомился с языком SQL-DML который позволяет производить операции над данными, хранящимися в базе. Основные операции языка SQL-DML:

- INSERT – добавление записи в таблицу
- SELECT – выборка данных из таблицы
- UPDATE – обновление данных, находящихся в таблице
- DELETE – удаление данных из таблицы.
- JOIN – объединение таблиц для выборки
- WHERE – наложение условия на выборку
- ORDER BY - группировка выбранных данных

Использование данных операторов открывает большие возможности по созданию запросов различного характера, а именно объединение таблиц, группировку данных, вложенные запросы, операторы условий и т.д.

Также стоит отметить полезность операции recursive, которая позволяет рекурсивно извлекать(с пределом глубины в 1024) данные из требуемой таблицы.