Московский авиационный институт

(национальный исследовательский университет)

**Факультет №8**

**«Информационные технологии и прикладная математика»**

**Лабораторная работа**

**по курсу**

**«Базы данных»**

**«Учет товаров в магазине запчастей для компьютеров»**

Студент: Бахарев Алексей Тимофеевич

Группа: М80 - 306Б - 16

Преподаватель: Кузнецова Елена Владимировна

Оценка:

Дата:

Москва, 2018

1. ТАБЛИЦЫ И СХЕМА ДАННЫХ.

Описание таблиц:

**cashier**– таблица, содержащая в себе данные о всех кассирах предприятия.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Наименование полей | cashier\_id | name | last\_name | phone\_number |
| Значение данных | Идентификационный номер кассира | Имя кассира | Фамилия кассира | Телефонный номер кассира |
| Тип данных | numeric | character | character | character |
| Размер поля | 3 | 10 | 20 | 10 |
| Тип ключа | Primary | Regular | Regular | Regular |

**cashbox**– таблица, содержащая данные о кассовых аппаратах, используемых на предприятии.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Наименование полей | cashbox\_id | producer | start\_of\_use | service\_date |
| Значение данных | Идентификационный номер кассового аппарата | Название компонента | Дата ввода в эксплуатацию | Дата последнего тех обслуживания |
| Тип данных | numeric | character | date | date |
| Размер поля | 4 | 100 |  |  |
| Тип ключа | Primary | Regular | Regular | Regular |

**сheck**– таблица, содержащая информацию о выданных чеках.

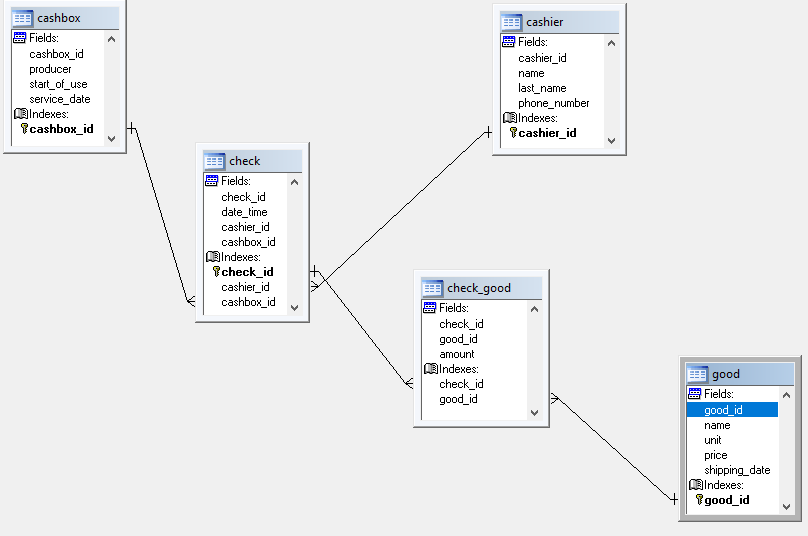
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Наименование полей | check\_id | date\_time | cashier\_id | cashbox\_id |
| Значение данных | Идентификационный номер чека, выбитого кассиром на кассовом аппарате | Дата и время создания чека | Идентификационный номер кассира | Идентификационный номер кассового аппарата |
| Тип данных | numeric | date | numeric | numeric |
| Размер поля | 4 |  | 3 | 4 |
| Тип ключа | Primary | Regular | Regular | Regular |

**Check\_good** – связующая таблица для таблиц чеков и товаров.

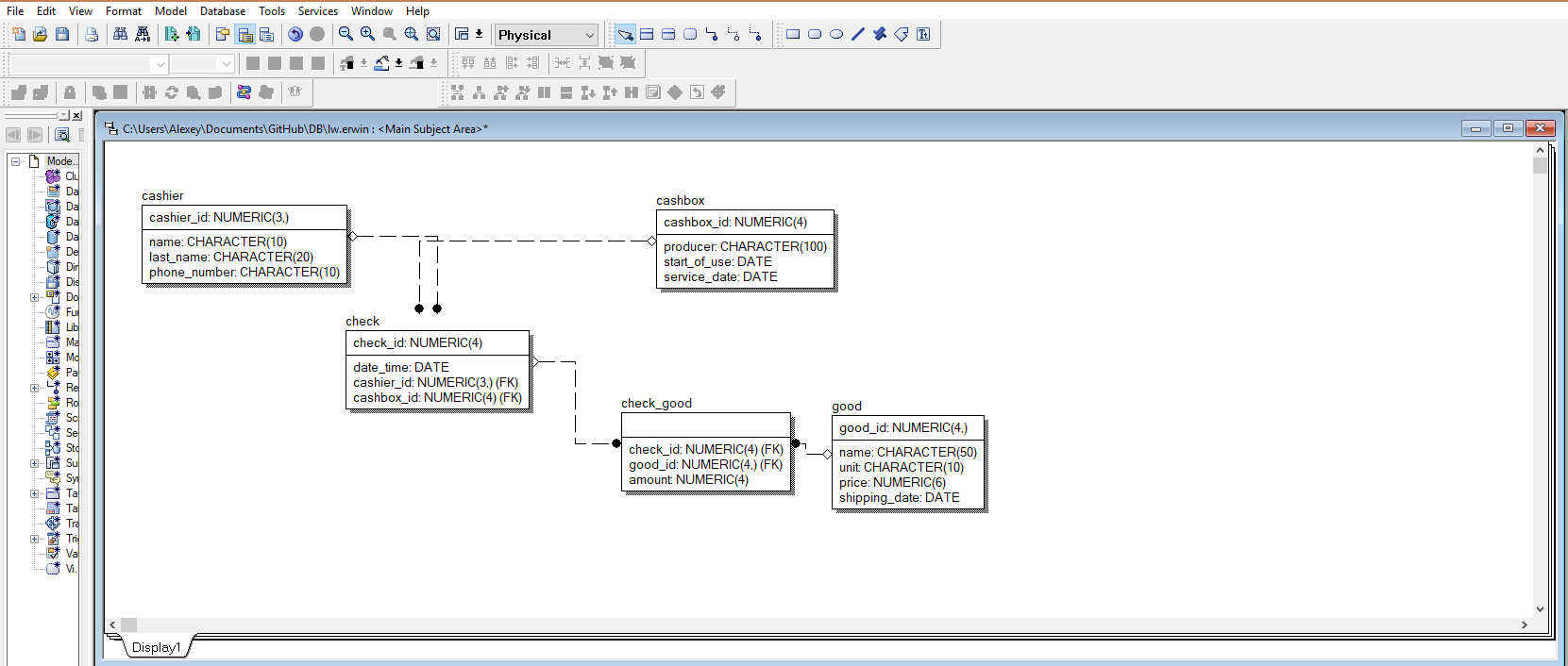
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Наименование полей | check\_good\_id | check\_id | good\_id | amount |
| Значение данных | Идентификационный номер записи | Идентификационный номер лекарства(например: M1) | Название лекарства | Идентификационный номер компании(например: I1) |
| Тип данных | integer | numeric | numeric | numeric |
| Размер поля | 4 | 4 | 4 | 4 |
| Тип ключа | Primary | Regular | Regular | Regular |

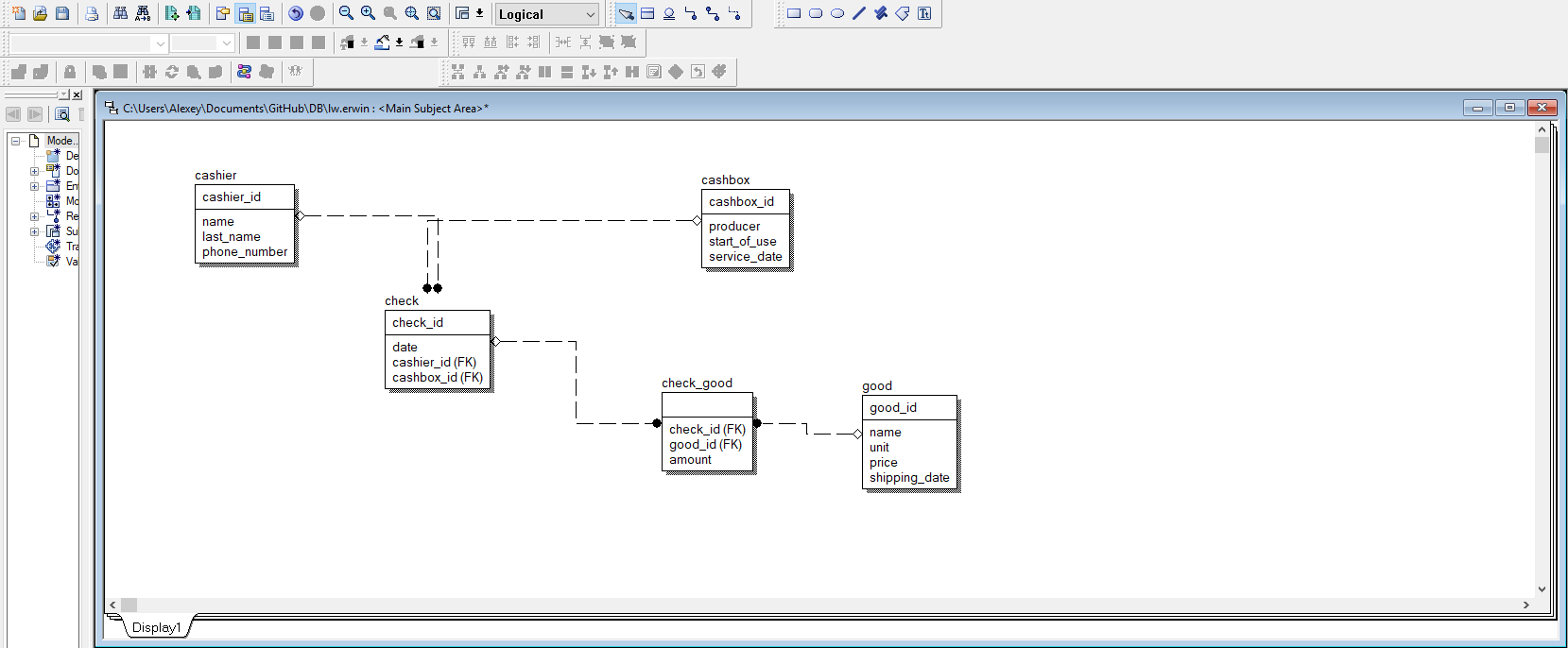
**good** – таблица, в которой указана информация о товарах, реализуемых на предприятии.

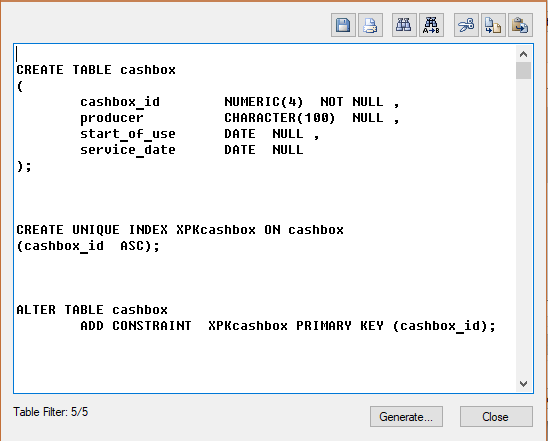
|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Наименование полей | good\_id | name | unit | price | shipping\_date |
| Значение данных | Идентификационный номер товара | Название товара | Единица измерения | цена | Дата выгрузки на склад |
| Тип данных | numeric | character | character | numeric | date |
| Размер поля | 10 | 50 | 10 | 6 |  |
| Тип ключа | Primary | Regular | Regular | Regular | Regular |



1. Проектирование базы данных







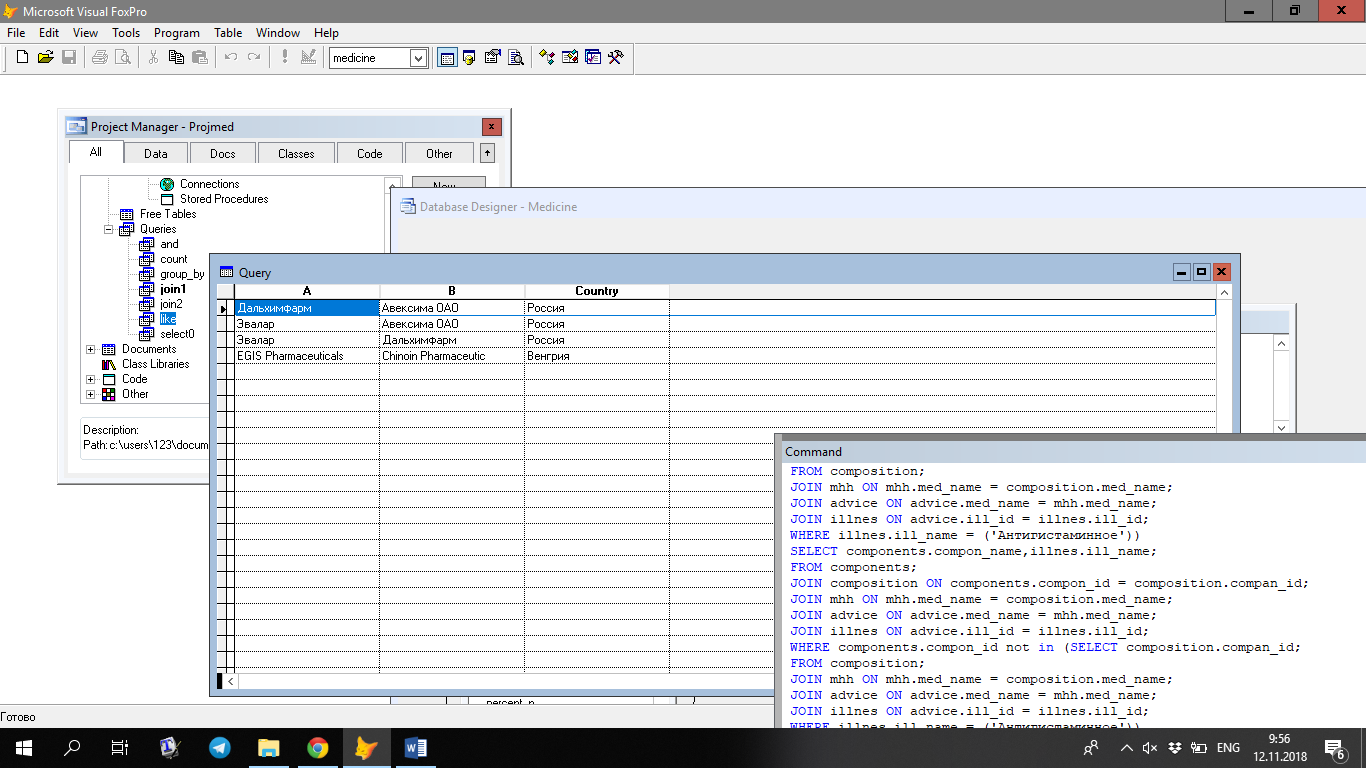
1. Запросы:

**1.Пары компаний из одной страны.**

SELECT a.company\_name as A, b.company\_name as B, a.country;

FROM company as a, company as b;

WHERE (a.company\_name > b.company\_name) AND (a.country = b.country)



**2.Какие компоненты не используются в антигистаминных препаратах?**

SELECT components.compon\_name,illnes.ill\_name;

FROM components;

JOIN composition ON components.compon\_id = composition.compan\_id;

JOIN mhh ON mhh.med\_name = composition.med\_name;

JOIN advice ON advice.med\_name = mhh.med\_name;

JOIN illnes ON advice.ill\_id = illnes.ill\_id;

WHERE components.compon\_id not in (SELECT composition.compan\_id;

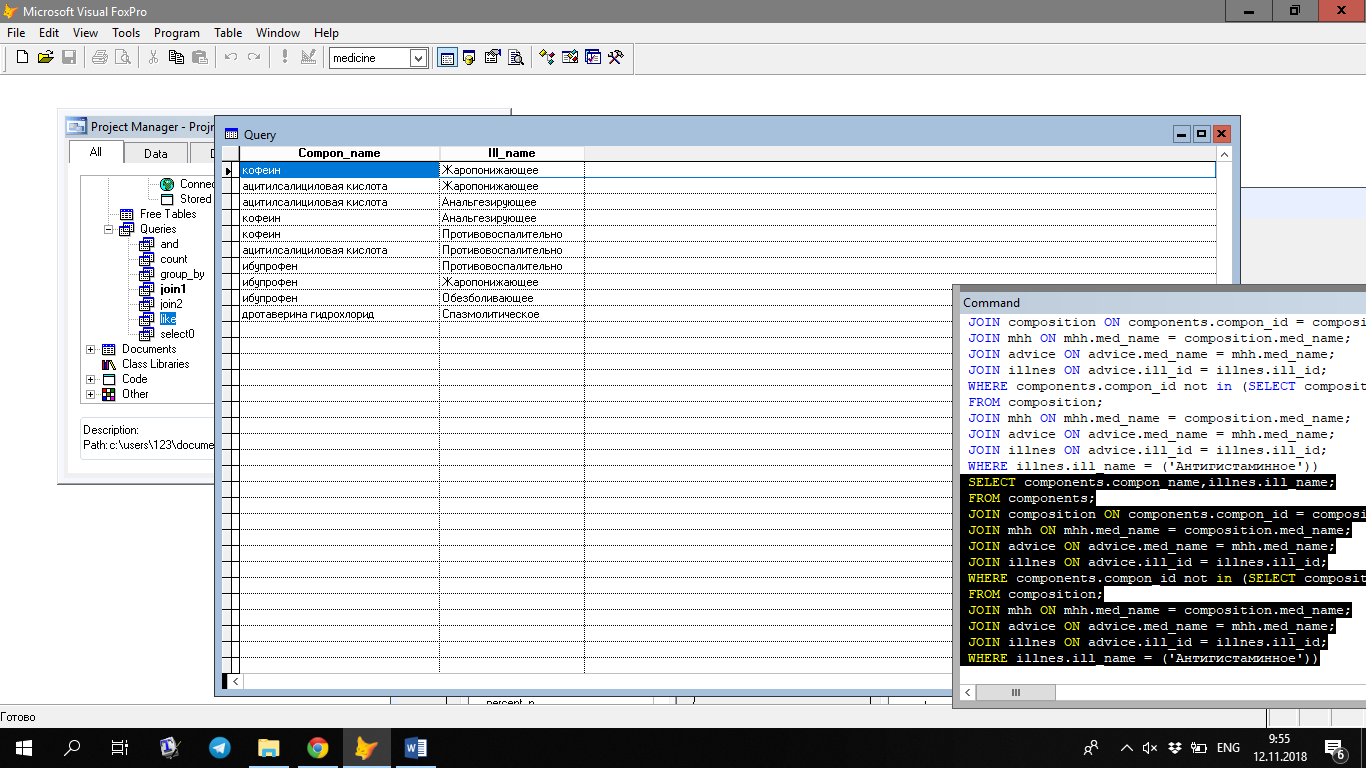
FROM composition;

JOIN mhh ON mhh.med\_name = composition.med\_name;

JOIN advice ON advice.med\_name = mhh.med\_name;

JOIN illnes ON advice.ill\_id = illnes.ill\_id;

WHERE illnes.ill\_name = ('Антигистаминное'))



**3.Компания, которая выпускает максимальное количество лекарств**.

SELECT A.company, A.count;

FROM (;

SELECT company.company\_name as company,COUNT(directory.med\_name) as count;

FROM directory;

JOIN company ON company.company\_id = directory.company\_id;

GROUP BY company.company\_name) as A;

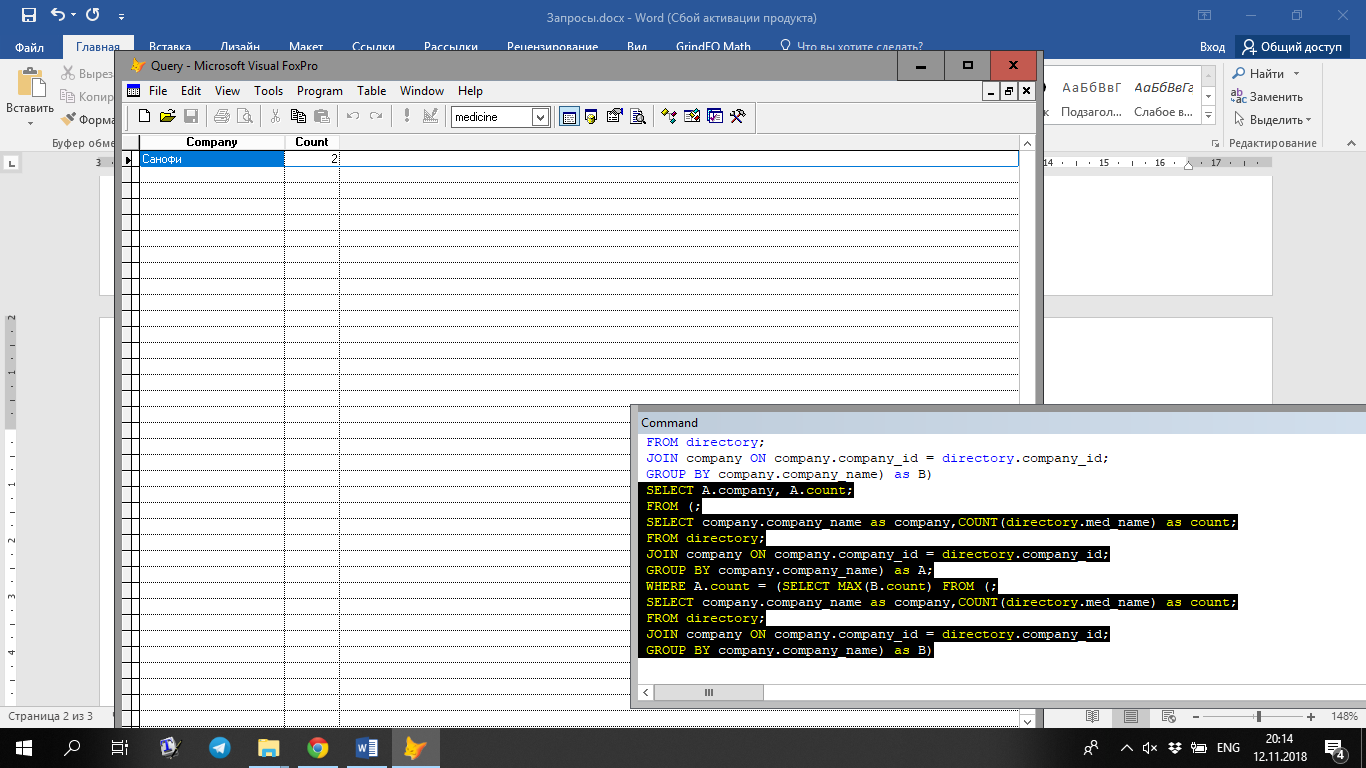
WHERE A.count = (SELECT MAX(B.count) FROM (;

SELECT company.company\_name as company,COUNT(directory.med\_name) as count;

FROM directory;

JOIN company ON company.company\_id = directory.company\_id;

GROUP BY company.company\_name) as B)



**4. Лекарства, у которых процент содержания активного вещества выше среднего.**

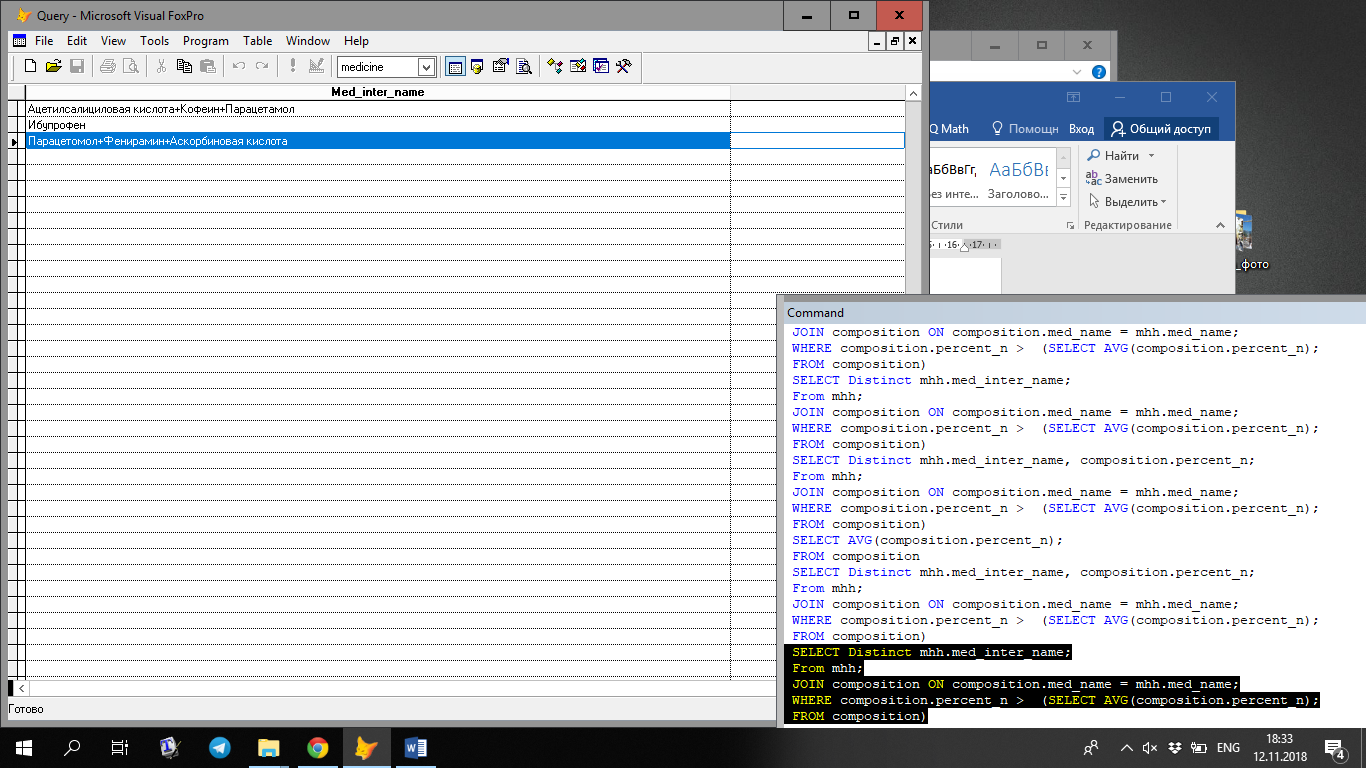
SELECT Distinct mhh.med\_inter\_name;

From mhh;

JOIN composition ON composition.med\_name = mhh.med\_name;

WHERE composition.percent\_n > (SELECT AVG(composition.percent\_n);

FROM composition)



**5. Лекарства, в которые входят такие же компоненты, как и в Цитрамон П.**

SELECT DISTINCT components.compon\_name,directory.med\_title;

FROM components;

JOIN composition ON components.compon\_id = composition.compan\_id;

JOIN mhh ON mhh.med\_name = composition.med\_name;

JOIN directory ON directory.med\_name = mhh.med\_name;

WHERE composition.compan\_id in (;

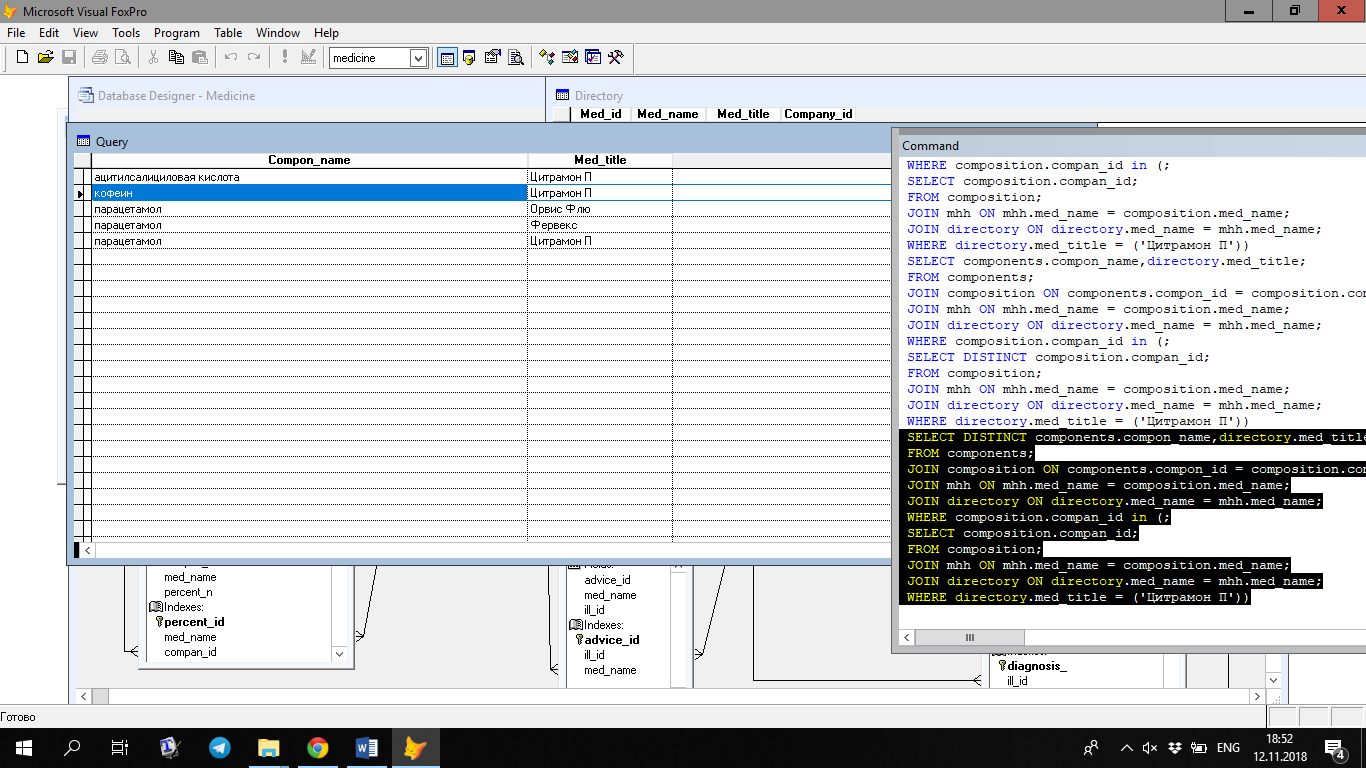
SELECT composition.compan\_id;

FROM composition;

JOIN mhh ON mhh.med\_name = composition.med\_name;

JOIN directory ON directory.med\_name = mhh.med\_name;

WHERE directory.med\_title = ('Цитрамон П'))



**6. Лекарство, в котором меньше всего парацетамола.**

SELECT DISTINCT directory.med\_title, components.compon\_name, composition.percent\_n;

FROM components;

JOIN composition ON components.compon\_id = composition.compan\_id;

JOIN mhh ON mhh.med\_name = composition.med\_name;

JOIN directory ON directory.med\_name = mhh.med\_name;

WHERE composition.percent\_n in;

(SELECT Min(composition.percent\_n);

FROM composition;

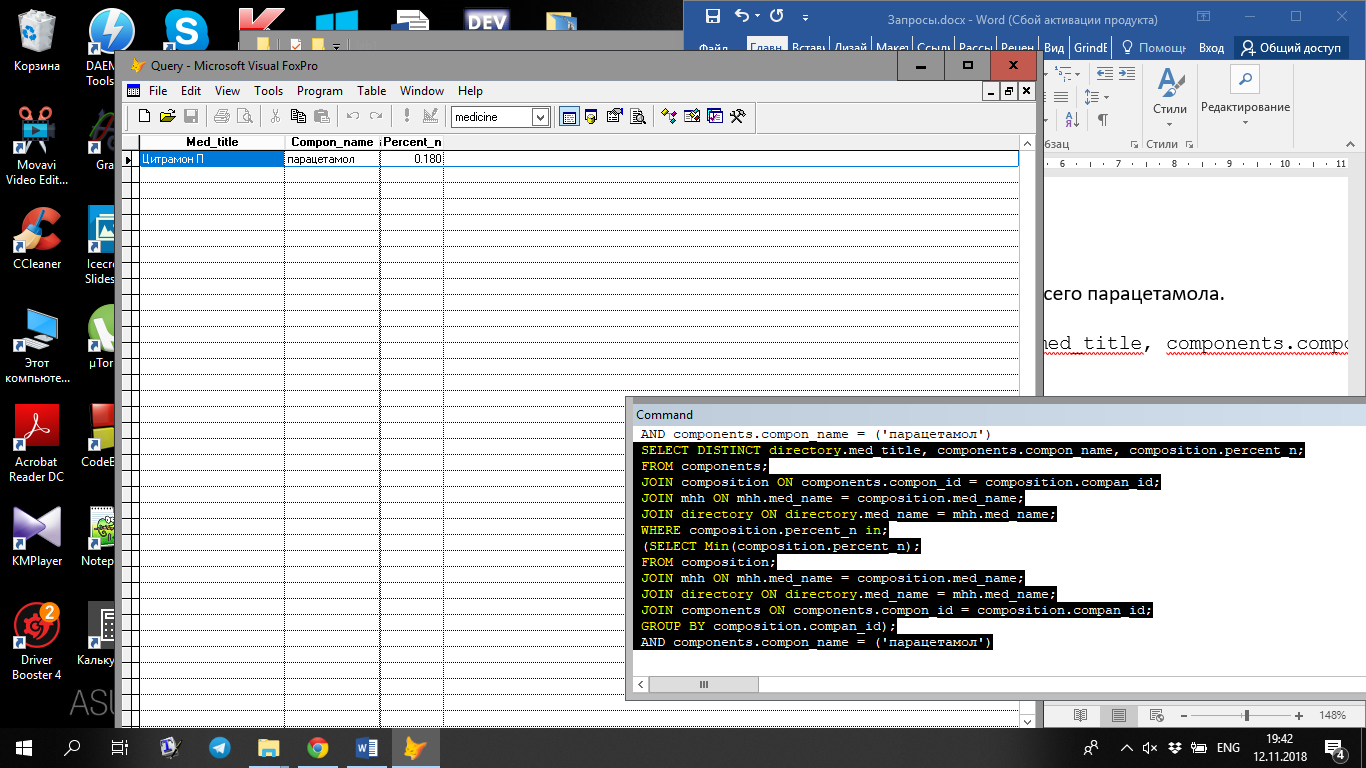
JOIN mhh ON mhh.med\_name = composition.med\_name;

JOIN directory ON directory.med\_name = mhh.med\_name;

JOIN components ON components.compon\_id = composition.compan\_id;

GROUP BY composition.compan\_id);

AND components.compon\_name = ('парацетамол')



2. SELECT cashier.name FROM cashbox, check, cashier;

WHERE cashbox.cashbox\_id = check.cashbox\_id AND check.cashier\_id = cashier.cashier\_id AND DTOS(check.date\_time) > "20110501" AND DTOS(check.date\_time) < "20111231" AND cashbox.cashbox\_id = 100;

group BY cashier.name

4. SELECT check.check\_id FROM check, check\_good, good;

WHERE check.check\_id = check\_good.check\_id AND check\_good.good\_id = good.good\_id AND good.good\_id = 8100 AND DTOS(check.date\_time) >= "20110601" AND DTOS(check.date\_time) <= "20110901"

1. SELECT good.name as A, good.name as B FROM check, check\_good, good;

WHERE check.check\_id = check\_good.check\_id AND check\_good.good\_id = good.good\_id AND check.check\_id = 15 AND A > B

3, SELECT check.check\_id, SUM(check\_good.amount \* good.price) as sum FROM check, check\_good, good INTO CURSOR tmp;

WHERE check.check\_id = check\_good.check\_id AND ;

check\_good.good\_id = good.good\_id ;

GROUP BY check.check\_id

SELECT tmp.check\_id, tmp.sum FROM tmp ;

WHERE tmp.sum in (SELECT MAX(tmp.sum) from tmp)

5, SELECT cashier.name, SUM(good.price \* check\_good.amount) as sum into cursor tmp;

FROM cashier, check, check\_good, good ;

WHERE cashier.cashier\_id = check.cashier\_id AND ;

check.check\_id = check\_good.check\_id AND ;

check\_good.good\_id = good.good\_id AND ;

DTOS(check.date\_time) > "20100101" AND ;

DTOS(check.date\_time) < "20101201" ;

GROUP BY cashier.name

SELECT \* FROM tmp

SELECT tmp.name, tmp.sum FROM tmp ;

WHERE tmp.sum in (SELECT MAX(tmp.sum) FROM tmp)