**Московский Авиационный Институт**

**(Национальный Исследовательский Университет)**

Институт информационных технологий и прикладной математики

**Лабораторная работа**

По курсу «Базы Данных»

Тема: «Аптека(готовые формы)»

Студент: Шорохов А. П.

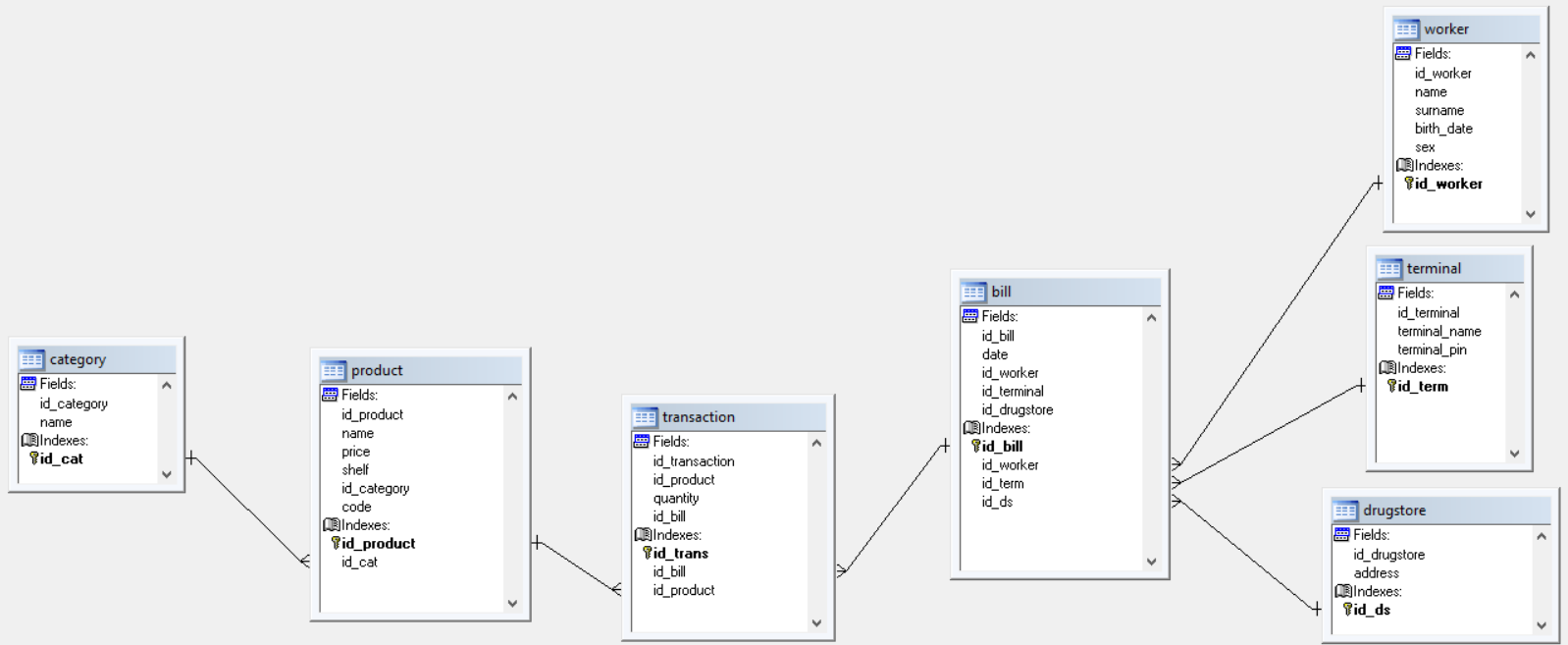
Группа: М8О-306Б-18

Преподаватель: Кузнецова Е. В.

Оценка:

Москва, 2020

**Таблицы и схема данных в Visual FoxPro**



Наша база данных состоит из 7 таблиц. Они называются category, product, transaction, bill, worker, terminal, drugstore.

Category – таблица категорий лекарств.

|  |  |  |
| --- | --- | --- |
| Наименование поля | id\_category | name |
| Значение данных | Идентификационный номер категории | Название категории |
| Тип данных | Integer | Character |
| Размер поля | 4 | 30 |
| Тип ключа | Primary |  |

Product – таблица лекарств, продающихся в аптеке.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Наименование поля | id\_product | name | price | shelf | id\_category | code |
| Значение данных | Идентификационный номер лекарства | Название лекарства | Цена 1 упаковки лекарства | Номера полок, на которых хранится лекарство на складе | Идентификационный номер категории | Артикул лекарства |
| Тип данных | Integer | Character | Currency | Integer | Integer | Integer |
| Размер поля | 4 | 100 | 8 | 4 | 4 | 4 |
| Тип ключа | Primary |  |  |  | Regular |  |

Transaction – таблица транзакций.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Наименование поля | id\_transaction | id\_product | quantity | id\_bill |
| Значение данных | Идентификационный номер транзакции | Идентификационный номер лекарства | Количество лекарства | Идентификационный номер чека |
| Тип данных | Integer | Integer | Integer | Integer |
| Размер поля | 4 | 4 | 4 | 4 |
| Тип ключа | Primary | Regular |  | Regular |

Bill – таблица чеков

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Наименование поля | id\_bill | date | id\_terminal | id\_worker | id\_drugstore |
| Значение данных | Идентификационный номер чека | Дата и время | Идентификационный номер терминала | Идентификационный номер кассира | Идентификационный номер аптеки |
| Тип данных | Integer | DateTime | Integer | Integer | Integer |
| Размер поля | 4 | 8 | 4 | 4 | 4 |
| Тип ключа | Primary |  | Regular | Regular | Regular |

Worker – таблица работников аптеки

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Наименование поля | id\_worker | name | surname | birth\_date | sex |
| Значение данных | Идентификационный номер работника | Имя работника | Фамилия работника | Дата рождения работника | Пол работника |
| Тип данных | Integer | Character | Character | Date | Character |
| Размер поля | 4 | 15 | 15 | 8 | 1 |
| Тип ключа | Primary |  |  |  |  |

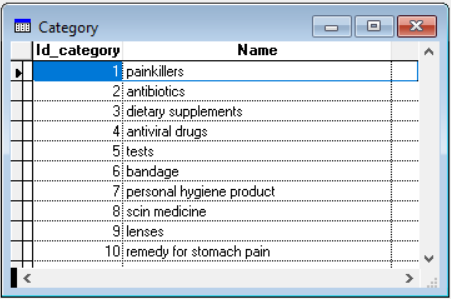
Terminal – таблица кассовых аппаратов(терминалов), которые выписывают чек

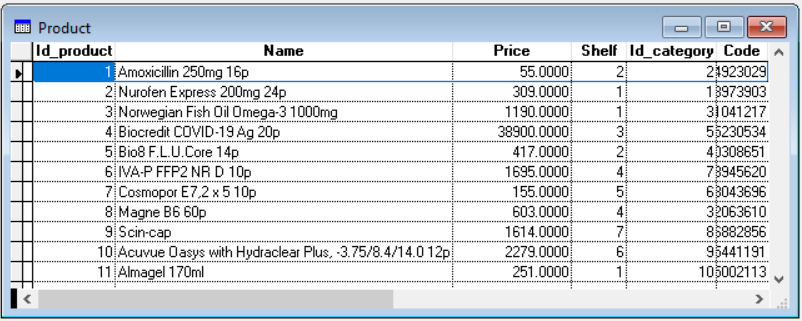
|  |  |  |  |
| --- | --- | --- | --- |
| Наименование поля | id\_terminal | terminal\_name | terminal\_pin |
| Значение данных | Идентификационный номер терминала | Имя системы терминала | Пароль от терминала |
| Тип данных | Integer | Character | Character |
| Размер поля | 4 | 20 | 15 |
| Тип ключа | Primary |  |  |

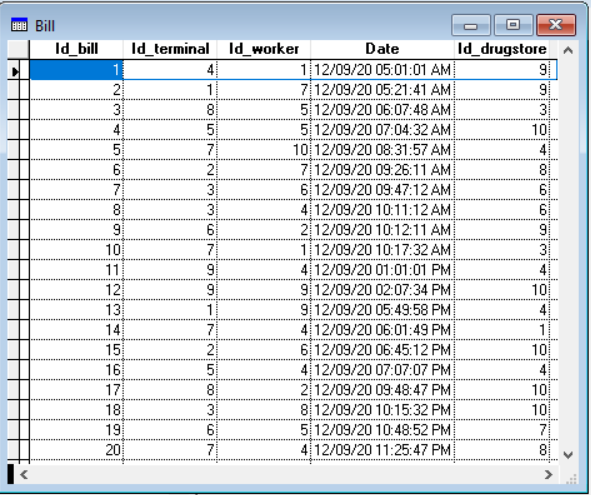
Drugstore – таблица адресов аптек. Адреса сгенерированы на Python.

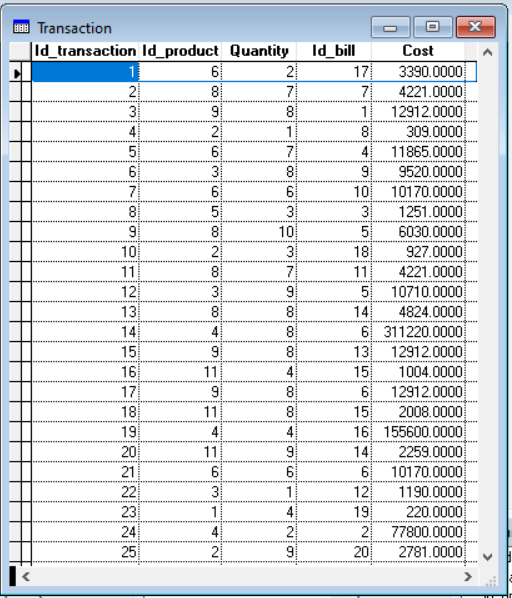
|  |  |  |
| --- | --- | --- |
| Наименование поля | id\_drugstore | address |
| Значение данных | Идентификационный номер аптеки | Адрес |
| Тип данных | Integer | Character |
| Размер поля | 4 | 50 |
| Тип ключа | Primary |  |

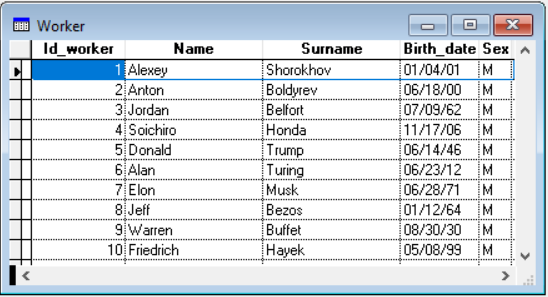
**Содержимое таблиц**

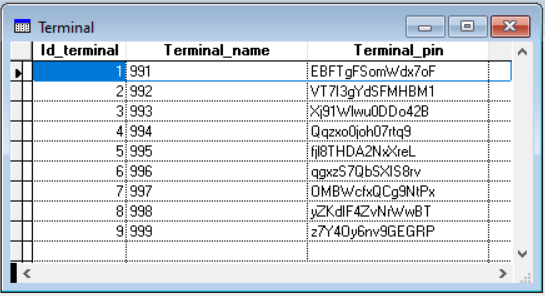


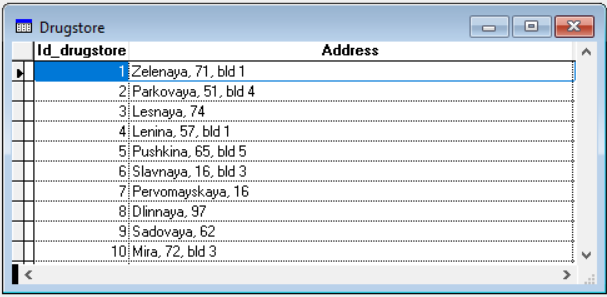












Для наполнения поля Terminal\_pin таблицы Terminal был написан генератор паролей на Python.

Его код:

**pin.py**

import string

import random

\_string = string.ascii\_letters + "0123456789"

for i in range(15):

print(random.choice(\_string),end = "")

print()

**address.py**

import random

streets = ["Zelenaya", "Lenina", "Sadovaya", "Pervomayskaya", "Parkovaya", "Lesnaya", "Mira", "Pushkina", "Dlinnaya", "Slavnaya"]

used = []

for i in range(10):

while(True):

rn = random.randint(0, 10)

if (rn not in used):

break

print(streets[rn] + ", " + str(random.randint(1, 100)), end = '')

if (random.randint(0, 1) == 1):

print(', bld ' + str(random.randint(1, 5)))

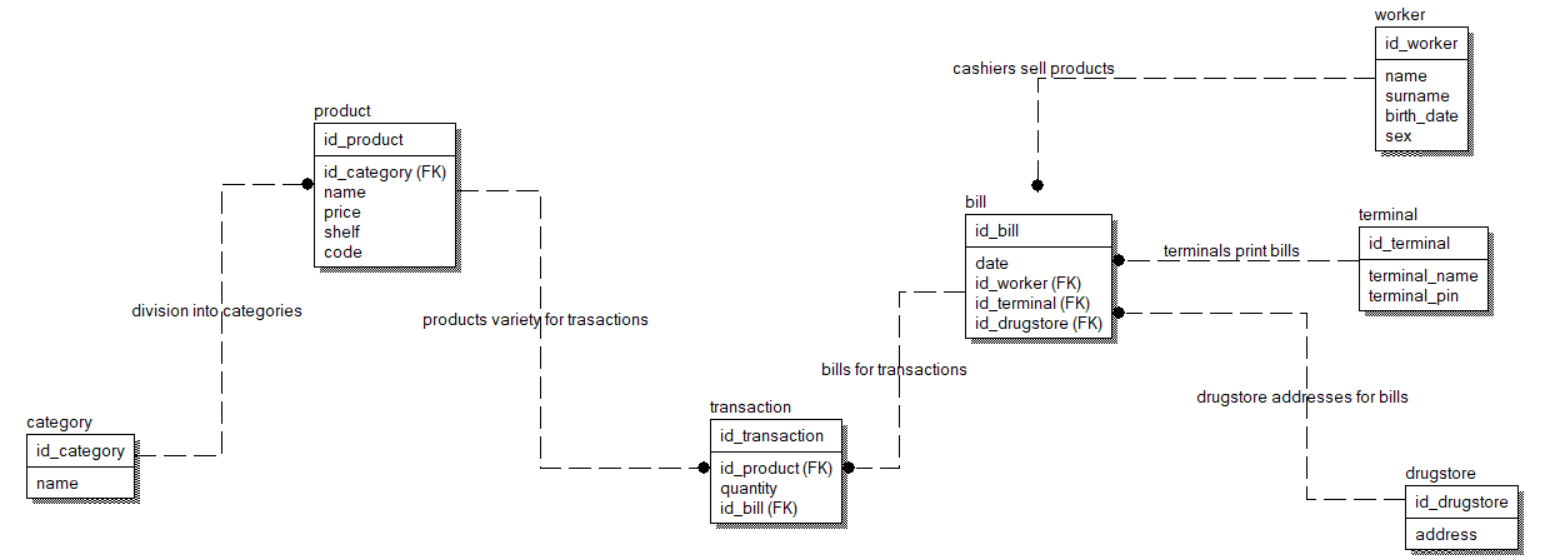
else:

print()

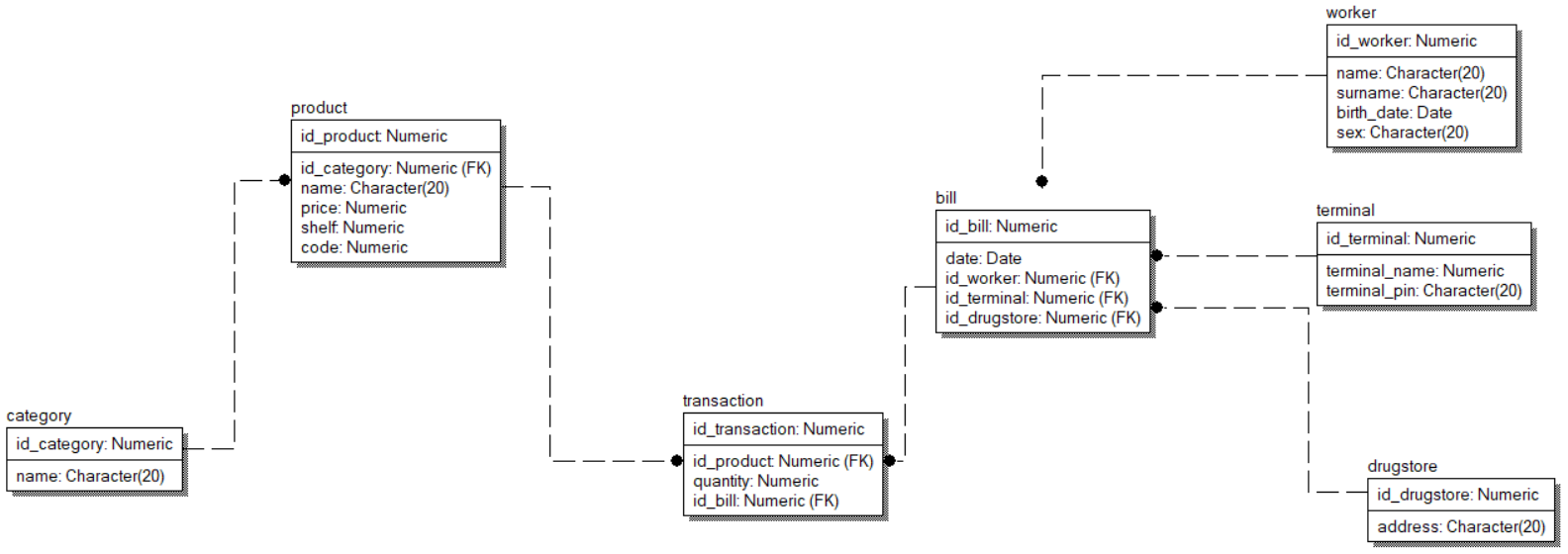
used.append(rn

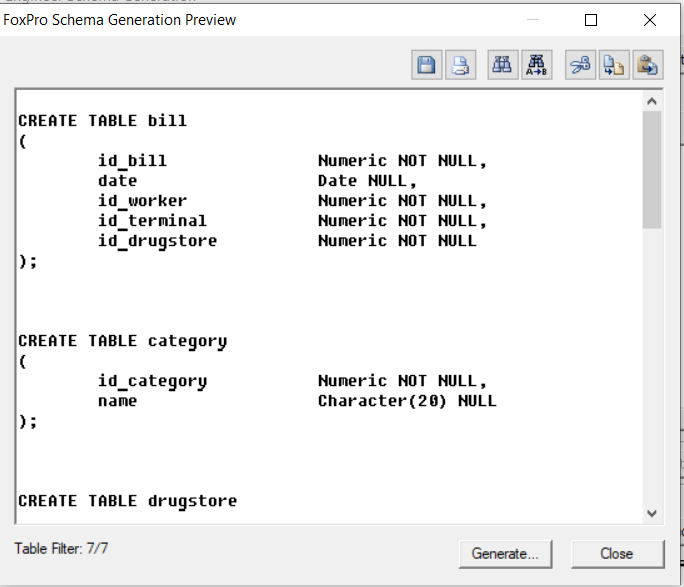
**Проектирование базы данных**

**Logical**



**Physical**





**Запросы**

1. Пары лекарств одной категории из заданного ценового диапазона (соединение таблицы с собой).

SELECT DISTINCT P1.name AS product1, P2.name AS product2;

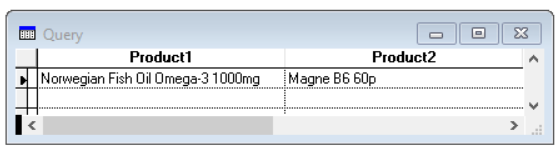
FROM product P1, product P2;

WHERE (P1.price BETWEEN 300.0000 AND 1200.0000);

AND (P2.price BETWEEN 300.0000 AND 1200.0000);

AND (P1.id\_category = P2.id\_category);

AND (P1.id\_product < P2.id\_product)



1. Кассир заданной аптеки, не выбивавший заданное лекарство заданного числа.

SELECT W.name, W.surname;

FROM worker W JOIN bill ON W.id\_worker = bill.id\_worker;

JOIN drugstore ON bill.id\_drugstore = drugstore.id\_drugstore;

WHERE W.id\_worker NOT IN;

(SELECT DISTINCT bill.id\_worker FROM bill;

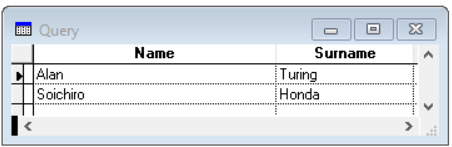
JOIN transaction ON bill.id\_bill = transaction.id\_bill;

JOIN product ON transaction.id\_product = product.id\_product;

WHERE product.name = 'Scin-cap' AND bill.date;

BETWEEN CTOT('12/10/2020 00:00:00am') AND CTOT('12/10/2020 11:59:59pm'));

AND drugstore.address = "Slavnaya, 16, bld 3"



1. Наиболее популярное лекарство (выбивали выше среднего).

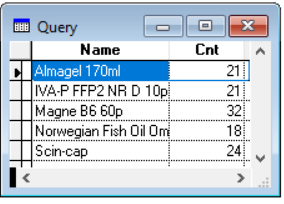
SELECT DISTINCT product.name, SUM(quantity) AS cnt;

FROM transaction JOIN product ON product.id\_product = transaction.id\_product;

GROUP BY product.name INTO CURSOR tmp

SELECT tmp.name, tmp.cnt FROM tmp;

WHERE tmp.cnt > (SELECT AVG(tmp.cnt) FROM tmp)



1. Самая успешная аптека, продавшая лекарств на максимальную сумму.

SELECT T.id\_bill, B.id\_drugstore, D.address,;

P.price \* T.quantity AS cnt;

FROM product P, transaction T, bill B, drugstore D;

WHERE P.id\_product = T.id\_product;

AND B.id\_bill = T.id\_bill;

AND D.id\_drugstore = B.id\_drugstore;

INTO CURSOR temp

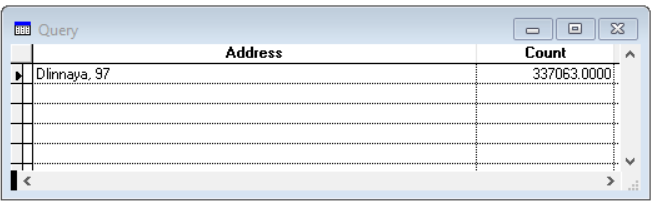
SELECT temp.address, SUM(temp.cnt) AS count;

FROM temp GROUP BY temp.address INTO CURSOR temp1

SELECT temp1.address, temp1.count;

FROM temp1;

WHERE temp1.count = (SELECT MAX(temp1.count) FROM temp1)



1. Придумать и реализовать запрос с предикатом на подзапросе (All, Any, Exists)

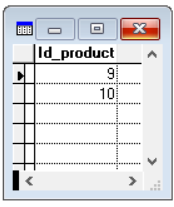
Поиск идентификационных номеров тех лекарств, цена которых выше заданной стоимости, а полка расположения(shelf) выше всех лекарств дешевле заданной стоимости.

SELECT product.id\_product FROM product;

WHERE product.price > 300.0000;

AND product.shelf > ALL(SELECT product.shelf;

FROM product WHERE product.price < 300.0000)



1. Кассир, выбивший чеков на максимальную сумму на заданном интервале дат.

SELECT T.id\_bill, B.id\_worker, P.price \* T.quantity AS cnt;

FROM product P, transaction T, bill B;

WHERE P.id\_product = T.id\_product;

AND B.id\_bill = T.id\_bill;

AND B.date;

BETWEEN CTOT('12/10/2020 00:00:00am');

AND CTOT('12/10/2020 11:59:59pm') INTO CURSOR t

SELECT t.id\_worker, SUM(t.cnt) AS count;

FROM t GROUP BY t.id\_worker INTO CURSOR t1

SELECT W.name, W.surname, T.count;

FROM worker W, t1 T;

WHERE W.id\_worker = t1.id\_worker;

AND T.count = (SELECT MAX(t1.count) FROM t1)



1. Лекарства, которые заданного числа продала исключительно заданная аптека (Заданная и никакая другая).

SELECT product.name, drugstore.address,;

product.id\_product, drugstore.id\_drugstore;

FROM product;

JOIN transaction ON product.id\_product = transaction.id\_product;

JOIN bill ON bill.id\_bill = transaction.id\_bill;

JOIN drugstore ON bill.id\_drugstore = drugstore.id\_drugstore;

WHERE bill.date BETWEEN CTOT('12/10/2020 00:00:00am') AND CTOT('12/10/2020 11:59:59pm');

INTO CURSOR t0

SELECT name, address FROM t0;

AS f WHERE 1 = (SELECT COUNT(t0.id\_drugstore);

FROM t0 WHERE t0.name = f.name);

AND f.address = 'Dlinnaya, 97'

