SQL Homework

Магазин еды

Вам представляется возможность поработать с базой данных вымышленного продуктового онлайн-магазина. Внутри неё будут несколько таблиц: Users, Goods, Orders, GoodsInOrders.

Таблица Users содержит в себе информацию о пользователях данного магазина:

```
create table Users (
    id integer primary key autoincrement,  -- user id
    gender text not null,   -- gender, can be "M" or "F"
    name text not null,   -- name
    country text not null,  -- country code, "RU", "US", "UK"...
    birth_date date not null  -- birth date
);
```

Таблица Goods содержит в себе информацию о товарах:

```
create table Goods (
    id integer primary key autoincrement,  -- good id
    name text not null,  -- name
    quantity integer not null,  -- quantity in units
    units text,  -- units "KG", "L", "ML"
    price real not null -- price
);
```

```
sqlite> select * from Goods limit 5;
1|Beans|5|KG|587.5
2|Maize|5|KG|145.0
3|Rice|5|KG|90.0
4|Maize meal|5|KG|252.0
5|Bread|5|KG|50.0
```

Что означает, что мы можем купить 5 килограмм бобов, за 587.5 условных единиц. Все стоимости здесь указаны в условных единицах.

Таблица Orders содержит информацию о заказах:

```
create table Orders (
    id integer primary key autoincrement,  -- order id
    user_id integer not null,  -- user id who ordered
    created datetime default CURRENT_TIMESTAMP,  -- when order was
                                                 -- created
    paid bool default FALSE  --  is this order paid
);
```

```
sqlite> select * from Orders limit 2;
1|86|2002-06-12 02:28:15|0
2|42|2002-06-14 05:38:20|1
```

Здесь следует обратить внимание, что sqlite3 хранит логические переменные True/False в качестве числовых значений 1/0. Колонка paid в данной таблице показывает был ли заказ оплачен. Человек мог создать заказ и не оплатить его, он получит товары лишь тогда, когда оплатит. Последняя таблица GoodsInOrders описывает, какие товары входят в конкретный заказ:

```
create table GoodsInOrders (
    order_id integer not null,
    good_id integer not null
);
```

```
sqlite> select * from GoodsInOrders limit 2;
1|6
1|219
```

Что означает, что товары с *id* 6 и 219 входят в заказ с *id* 1. В один заказ может входить несколько товаров (в том числе и одинаковых).

Вам будет предоставлен файл с этой базой данных. Таблицы создавать не нужно. Изменять данные и вводить свои данные тоже не стоит. Задание состоит из двух частей:

В первой части от вас потребуется лишь составить правильные sql-запросы. Рекомендую это делать внутри треминала sqlite3.

1. Всю информацию о всех пользователях **(все колонки в любом порядке)**
2. Количество всех пользователей **(число)**
3. Количество пользователей 40 лет или старше **(число)**. Чтобы сравнить столбец таблицы с конкретной датой можно использовать функцию date – where birth_date <= date("2000-12-20") с датой в формате "YYYY-MM-DD".
4. Страна + количество пользователей из данной страны **(страна|количество)**
5. Придумайте, как проверить, есть ли люди с одинаковым именем **(в любом удобном формате)**
6. Количество заказов в 2016 году **(число)**
7. День, когда совершили наибольшее число заказов **(день|число заказов)**
8. Процент неоплаченных заказов **(число)**
9. Всю информацию о хлебе среди товаров. Используйте конструкцию where name like "%bread%". **Ссылка**. **(все колонки в любом порядке)**
10. Самые 10 популярных товаров (встречаемость в GoodsInOrders) **(id|name|количество)**
11. Чистая выручка в 2016 году. Нужно учитывать только оплаченные заказы **(число)**
12. Самые 10 популярных товаров среди женщин **(id|название)**
13. Пользователь, который купил больше всего килограмм **(id|имя)**
14. (Дополнительно) Для каждой страны, самый популярный покупаемый товар в этой стране. **(Код страны|id товара|имя товара|количество)**

Разработчик сайта для нашего магазина уехал на майские в горы, а личный кабинет для пользователей запилить забыл. Вся надежда на вас!

Во второй части этого задания от вас потребуется написать функцию на языке python, которая собрала бы интересующую нас информацию. Пользователь может зайти в личный кабинет и увидеть неоплаченные заказы. Реализуйте функцию unpaid, которая принимает id пользователя:

```
def unpaid(user_id):
    # body
    return data

print(unpaid(23))
# [('Maria Doomhammer', 58, 45013.15),
#  ('Maria Doomhammer', 374, 5110.61),
#  ('Maria Doomhammer', 1039, 4.84)]
```

Результат данной функции – список содержащий кортежи (имя, id заказа, сумма в заказе).

Как решать:

1. Сначала составить правильный sql-запрос для одного конкретного пользователя, который бы вернул эту информацию об этом пользователе в правильном формате. Запишите этот запрос в переменную (например, query)

2. Внутри функции создаем соединение с базой и курсор (всё как на практике было)

3. В нашем запросе, где мы указываем id пользователя поставить "?"

4. Затем вызвать метод execute курсора в следующим способом:

   ```
   data = cur.execute(query, [user_id]).fetchall()
   ```

   Такая конструкция подставит user_id вместо id в строке нашего запроса (query), и сразу вернет нам список в нужном формате

   Про этот ситаксис можно почитать на странице официальной документации sqlite3.