**Тестовое задание**

Для написания запросов рекомендуется использовать стандарт языка SQL.   
К заданию 2 и 3 так же приложить скрипт для заполнения тестовыми данными.

# Задание 1.

Какие поля вы бы выбрали для задания первичного ключа в нижеприведённой таблице? Объясните выбор.

**Ответ:**

Мой выбор первичного ключа это **CityID**.

Давайте разберем примерное назначение таблицы. Из названия видим, что ее основная цель хранить в себе информацию о городах, первичный ключ — это уникальное значение для идентификации мы можем взять только id, почему? Да, мы могли бы взять название города, но тогда при работе с БД нам может выйти город с таким же названием, к примеру у нас есть два Кирова. Еще можно попробовать по координатам, но это лишняя нагрузка, да и полушарий у нас два, нецелесообразно. Край, тоже не имеет смысла, население тоже.

**CREATE TABLE** Cities (

**CityID INT NOT NULL**,

**CityName NVARCHAR**(50) **NOT NULL**,

**CountryID INT NOT NULL**,

**StateProvinceID INT NOT NULL**,

**[Location]** GEOGRAPHY **NULL**,

**LatestRecordedPopulation BIGINT NULL**,

)

# Задание 2.

Таблица Orders содержит список заказов, в котором SalespersonPersonID – продавец, CustomerID – покупатель, а OrderDate – дата заказа.

Напишите запрос, который вернёт список продавцов, оформивших заказы в 2013 году и количество заказов, оформленных каждым продавцом. Формат результата запроса: SalespersonPersonID, OrderCount.

В качестве ответа приведите текст запроса.

**CREATE TABLE** Sales.Orders (

**OrderID INT PRIMARY KEY NOT NULL**,

**CustomerID INT NOT NULL**,

**SalespersonPersonID INT NOT NULL**,

**OrderDate DATE NOT NULL**

)



**Ответ:**

Запрос два столбца **SalespersonPersonID и OrderCount.**   
**SELECT `SalespersonPersonID`, COUNT(`OrderID`) AS OrderCount FROM orders WHERE YEAR(`OrderDate`) = 2013 GROUP BY `SalespersonPersonID`;**Дополнительно запрос на заполнение таблицы:   
  
**SELECT \* FROM `orders`;**

**INSERT INTO orders (`OrderID`,`CustomerID`,`SalespersonPersonID`,`OrderDate`) VALUES (19441,537,7,"2013-12-31"), (19442,473,2,"2013-12-31"), (19443,474,20,"2014-01-01"),(19444,171,20,"2014-01-01"), (19445,553,15,"2014-01-01"), (19446,406,2,"2014-01-01")**

# Задание 3.

Таблица OrderLines содержит список товаров каждого заказа, где UnitPrice – цена единицы товара, а Quantity – их количество. OrderLines связана с Orders по OrderID.

**CREATE TABLE** Sales.OrderLines (

**OrderLineID INT PRIMARY KEY NOT NULL**,

**OrderID INT NOT NULL**,

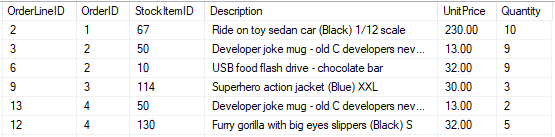
**StockItemID INT NOT NULL**,

[**Description] NVARCHAR**(100) **NOT NULL**,

**Quantity INT NOT NULL**,

**UnitPrice DECIMAL**(18, 2) **NULL**

)



Напишите запрос, который вернёт список заказов, покупателя, а также полную стоимость заказа. Результат должен содержать только заказы с полной стоимостью более 25000 и быть упорядоченным по убыванию этой величины. Формат результата: OrderID, CustomerID, TotalCost.

**Ответ:**

**SELECT orderlines.OrderID, CustomerID, SUM(`Quantity` \* `UnitPrice`) AS TotalCost FROM orderlines**

**JOIN `orders` ON CustomerID = CustomerID**

**GROUP BY OrderID HAVING SUM(`Quantity` \* `UnitPrice`) > 25000**

**ORDER BY TotalCost DESC;**

В ходе выполнения 3 задачи, была обнаружена логическая ошибка в самом задании, можете уточнить, нет ли опечаток в задании, или проверить его полноту.