

EPAM Systems, RD Dep.
Homework Task
10 – Queries to Database

REVISION HISTORY					
Ver.	Description of Change	Author	Date	Approved	
				Name	Effective Date
1.0	Initial version	Святослав Куликов	27.03.2012		
2.0		Nikolay Piskarev	29.05.2017		

Цель выполнения

Получить практические навыки работы с Query Analyzer для выполнения операторов языка Transact SQL. Освоить применение различных команд DML для получения, изменения и удаления данных в БД. Научиться разрабатывать запросы с применением следующих операторов и конструкций языка Transact SQL:

- DISTINCT, ORDER BY, IN, BETWEEN, LIKE, EXISTS, CASE;
- агрегатных функций SUM, COUNT и предложений GROUP BY и HAVING;
- соединений таблиц и самосоединений, конструкций Inner и Outer JOIN;
- простых и коррелированных подзапросов;

Научиться разрабатывать хранимые процедуры и функции и вызывать их. Научиться применять и использовать курсоры

Ограничения

При выполнении задания использовать требования, изложенные в руководстве «Database Modeling Guidelines» EPAM Systems

Список заданий

- 1 Работа с типами данных Date, NULL значениями, трехзначная логика. Возвращение определенных значений в результатах запроса в зависимости от полученных первоначальных значений результата запроса. Высветка в результатах запроса только определенных колонок.**
 - 1.1** Выбрать в таблице Orders заказы, которые были доставлены после 6 мая 1998 года (колонок ShippedDate) включительно и которые доставлены с ShipVia >= 2. Формат указания даты должен быть верным при любых региональных настройках, согласно требованиям статьи “Writing International Transact-SQL Statements” в Books Online раздел “Accessing and Changing Relational Data Overview”. Этот метод использовать далее для всех заданий. Запрос должен высвечивать только колонки OrderID, ShippedDate

и ShipVia.
Пояснить почему сюда не попали заказы с NULL-ом в колонке ShippedDate.

- 1.2 Написать запрос, который выводит только недоставленные заказы из таблицы Orders. В результатах запроса высвечивать для колонки ShippedDate вместо значений NULL строку 'Not Shipped' – использовать системную функцию CASE. Запрос должен высвечивать только колонки OrderID и ShippedDate.
- 1.3 Выбрать в таблице Orders заказы, которые были доставлены после 6 мая 1998 года (ShippedDate) не включая эту дату или которые еще не доставлены. В запросе должны высвечиваться только колонки OrderID (переименовать в Order Number) и ShippedDate (переименовать в Shipped Date). В результатах запроса высвечивать для колонки ShippedDate вместо значений NULL строку 'Not Shipped', для остальных значений высвечивать дату в формате по умолчанию.

2 Использование операторов IN, DISTINCT, ORDER BY, NOT

- 2.1 Выбрать из таблицы Customers всех заказчиков, проживающих в USA и Canada. Запрос сделать с только помощью оператора IN. Высвечивать колонки с именем пользователя и названием страны в результатах запроса. Упорядочить результаты запроса по имени заказчиков и по месту проживания.
- 2.2 Выбрать из таблицы Customers всех заказчиков, не проживающих в USA и Canada. Запрос сделать с помощью оператора IN. Высвечивать колонки с именем пользователя и названием страны в результатах запроса. Упорядочить результаты запроса по имени заказчиков.
- 2.3 Выбрать из таблицы Customers все страны, в которых проживают заказчики. Страна должна быть упомянута только один раз и список отсортирован по убыванию. Не использовать предложение GROUP BY. Высвечивать только одну колонку в результатах запроса.

3 Использование оператора BETWEEN, DISTINCT

- 3.1 Выбрать все заказы (OrderID) из таблицы Order Details (заказы не должны повторяться), где встречаются продукты с количеством от 3 до 10 включительно – это колонка Quantity в таблице Order Details. Использовать оператор BETWEEN. Запрос должен высвечивать только колонку OrderID.
- 3.2 Выбрать всех заказчиков из таблицы Customers, у которых название страны начинается на буквы из диапазона b и g. Использовать оператор BETWEEN. Проверить, что в результаты запроса попадает Germany. Запрос должен высвечивать только колонки CustomerID и Country и отсортирован по Country.
- 3.3 Выбрать всех заказчиков из таблицы Customers, у которых название страны начинается на буквы из диапазона b и g, не используя оператор BETWEEN. С помощью опции “Execution Plan” определить какой запрос предпочтительнее 3.2 или 3.3 – для этого надо ввести в скрипт выполнение текстового Execution Plan-а для двух этих запросов, результаты выполнения Execution Plan надо ввести в скрипт в виде комментария и по их результатам дать ответ на вопрос – по какому параметру было проведено сравнение. Запрос должен высвечивать только колонки CustomerID и Country и отсортирован по Country.

4 Использование оператора LIKE

- 4.1 В таблице Products найти все продукты (колонка ProductName), где встречается подстрока 'chocolate'. Известно, что в подстроке 'chocolate' может быть изменена одна буква 'c' в середине - найти все продукты, которые удовлетворяют этому условию. Подсказка: результаты запроса должны высвечивать 2 строки.

5 Использование агрегатных функций (SUM, COUNT)

- 5.1 Найти общую сумму всех заказов из таблицы Order Details с учетом количества закупленных товаров и скидок по ним. Результат округлить до сотых и высветить в стиле 1 для типа данных money. Скидка (колонка Discount) составляет процент из стоимости для данного товара. Для определения действительной цены на

проданный продукт надо вычесть скидку из указанной в колонке UnitPrice цены. Результатом запроса должна быть одна запись с одной колонкой с названием колонки 'Totals'.

- 5.2 По таблице Orders найти количество заказов, которые еще не были доставлены (т.е. в колонке ShippedDate нет значения даты доставки). Использовать при этом запросе только оператор COUNT. Не использовать предложения WHERE и GROUP.
- 5.3 По таблице Orders найти количество различных покупателей (CustomerID), сделавших заказы. Использовать функцию COUNT и не использовать предложения WHERE и GROUP.

6 Явное соединение таблиц, самосоединения, использование агрегатных функций и предложений GROUP BY и HAVING

- 6.1 По таблице Orders найти количество заказов с группировкой по годам. В результатах запроса надо высвечивать две колонки с названиями Year и Total. Написать проверочный запрос, который вычисляет количество всех заказов.
- 6.2 По таблице Orders найти количество заказов, сделанных каждым продавцом. Заказ для указанного продавца – это любая запись в таблице Orders, где в колонке EmployeeID задано значение для данного продавца. В результатах запроса надо высвечивать колонку с именем продавца (Должно высвечиваться имя полученное конкатенацией LastName & FirstName. Эта строка LastName & FirstName должна быть получена отдельным запросом в колонке основного запроса. Также основной запрос должен использовать группировку по EmployeeID.) с названием колонки 'Seller' и колонку с количеством заказов высвечивать с названием 'Amount'. Результаты запроса должны быть упорядочены по убыванию количества заказов.
- 6.3 По таблице Orders найти количество заказов, сделанных каждым продавцом и для каждого покупателя. Необходимо определить это только для заказов сделанных в 1998 году. В результатах запроса надо высвечивать колонку с именем продавца (название колонки 'Seller'), колонку с именем покупателя (название колонки

'Customer') и колонку с количеством заказов высвечивать с названием 'Amount'. В запросе необходимо использовать специальный оператор языка T-SQL для работы с выражением GROUP (Этот же оператор поможет выводить строку "ALL" в результатах запроса). **Группировки должны быть сделаны по ID продавца и покупателя.** Результаты запроса должны быть упорядочены по продавцу, покупателю и по убыванию количества продаж. В результатах должна быть сводная информация по продажам. Т.е. в результирующем наборе должны присутствовать дополнительно к информации о продажах продавца для каждого покупателя следующие строчки:

Seller	Customer	Amount				
ALL	ALL	<общее	число	продаж	для данного	продавца>
<имя>	ALL	<число	продаж	для данного	продавца	>
ALL	<имя>	<число	продаж	для данного	покупателя	>
<имя>	<имя>	<число продаж данного продавца для данного покупателя>				

- 6.4 Найти покупателей и продавцов, которые живут в одном городе. Если в городе живут только один или несколько продавцов или только один или несколько покупателей, то информация о таких покупателя и продавцах не должна попадать в результирующий набор. Не использовать конструкцию JOIN. В результатах запроса необходимо вывести следующие заголовки для результатов запроса: 'Person', 'Type' (здесь надо выводить строку 'Customer' или 'Seller' в зависимости от типа записи), 'City'. Отсортировать результаты запроса по колонке 'City' и по 'Person'.
- 6.5 Найти всех покупателей, которые живут в одном городе. В запросе использовать соединение таблицы Customers с собой - самосоединение. Высветить колонки CustomerID и City. Запрос не должен высвечивать дублируемые записи. Для проверки написать запрос, который высвечивает города, которые встречаются более одного раза в таблице Customers. Это позволит проверить правильность запроса.
- 6.6 По таблице Employees найти для каждого продавца его руководителя, т.е. кому он делает репорты. Высветить колонки с именами 'User Name' (LastName) и 'Boss'. В колонках должны быть

высвечены имена из колонки LastName. Высвечены ли все продавцы в этом запросе?

7 Использование Inner JOIN

- 7.1 Определить продавцов, которые обслуживают регион 'Western' (таблица Region). Результаты запроса должны высвечивать два поля: 'LastName' продавца и название обслуживаемой территории ('TerritoryDescription' из таблицы Territories). Запрос должен использовать JOIN в предложении FROM. Для определения связей между таблицами Employees и Territories надо использовать графические диаграммы для базы Northwind.

8 Использование Outer JOIN

- 8.1 Высветить в результатах запроса имена всех заказчиков из таблицы Customers и суммарное количество их заказов из таблицы Orders. Принять во внимание, что у некоторых заказчиков нет заказов, но они также должны быть выведены в результатах запроса. Упорядочить результаты запроса по возрастанию количества заказов.

9 Использование подзапросов

- 9.1 Высветить всех поставщиков колонка CompanyName в таблице Suppliers, у которых нет хотя бы одного продукта на складе (UnitsInStock в таблице Products равно 0). Использовать вложенный SELECT для этого запроса с использованием оператора IN. Можно ли использовать вместо оператора IN оператор '=' ?

10 Коррелированный запрос

- 10.1 Высветить всех продавцов, которые имеют более 150 заказов. Использовать вложенный коррелированный SELECT.

11 Использование EXISTS

- 11.1 Высветить всех заказчиков (таблица Customers), которые не имеют ни одного заказа (подзапрос по таблице Orders). Использовать коррелированный SELECT и оператор EXISTS.

12 Использование строковых функций

- 12.1 Для формирования алфавитного указателя Employees высветить из таблицы Employees список только тех букв алфавита, с которых начинаются фамилии Employees (колонок LastName) из этой таблицы. Алфавитный список должен быть отсортирован по возрастанию.

13 Разработка функций и процедур

- 13.1 Написать процедуру, которая возвращает самый крупный заказ для каждого из продавцов за определенный год. В результатах не может быть несколько заказов одного продавца, должен быть только один и самый крупный. В результатах запроса должны быть выведены следующие колонки: колонка с именем и фамилией продавца (FirstName и LastName – пример: Nancy Davolio), номер заказа и его стоимость. В запросе надо учитывать Discount при продаже товаров. Процедуре передается год, за который надо сделать отчет, и количество возвращаемых записей. Результаты запроса должны быть упорядочены по убыванию суммы заказа. Процедура должна быть реализована с использованием оператора SELECT и БЕЗ ИСПОЛЬЗОВАНИЯ КУРСОРОВ. Название функции соответственно GreatestOrders. Необходимо продемонстрировать использование этих процедур. Также помимо демонстрации вызовов процедур в скрипте Query.sql надо написать отдельный ДОПОЛНИТЕЛЬНЫЙ проверочный запрос для тестирования правильности работы процедуры GreatestOrders. Проверочный запрос должен выводить в удобном для сравнения с результатами работы процедур виде для определенного продавца для всех его заказов за определенный указанный год в результатах следующие колонки: имя продавца, номер заказа, сумму заказа. Проверочный запрос не должен повторять запрос, написанный в процедуре, - он

должен выполнять только то, что описано в требованиях по нему. ВСЕ ЗАПРОСЫ ПО ВЫЗОВУ ПРОЦЕДУР ДОЛЖНЫ БЫТЬ НАПИСАНЫ В ФАЙЛЕ Query.sql – см. пояснение ниже в разделе «Требования к оформлению».

- 13.2 Написать процедуру, которая возвращает заказы в таблице Orders, согласно указанному сроку доставки в днях (разница между OrderDate и ShippedDate). В результатах должны быть возвращены заказы, срок которых превышает переданное значение или еще недоставленные заказы. Значению по умолчанию для передаваемого срока 35 дней. Название процедуры ShippedOrdersDiff. Процедура должна высвечивать следующие колонки: OrderID, OrderDate, ShippedDate, ShippedDelay (разность в днях между ShippedDate и OrderDate), SpecifiedDelay (переданное в процедуру значение). Необходимо продемонстрировать использование этой процедуры.
- 13.3 Написать процедуру, которая высвечивает всех подчиненных заданного продавца, как непосредственных, так и подчиненных его подчиненных. В качестве входного параметра функции используется EmployeeID. Необходимо распечатать имена подчиненных и выровнять их в тексте (использовать оператор PRINT) согласно иерархии подчинения. Продавец, для которого надо найти подчиненных также должен быть высвечен. Название процедуры SubordinationInfo. В качестве алгоритма для решения этой задачи надо использовать пример, приведенный в Books Online и рекомендованный Microsoft для решения подобного типа задач. Пр продемонстрировать использование процедуры.
- 13.4 Написать функцию, которая определяет, есть ли у продавца подчиненные. Возвращает тип данных BIT. В качестве входного параметра функции используется EmployeeID. Название функции IsBoss. Пр продемонстрировать использование функции для всех продавцов из таблицы Employees.

Требования к оформлению

1. Для каждой домашней работы необходимо создать собственную папку в корне репозитория. ,
2. В качестве результата выполненного задания необходимо представить файлы, указанные ниже в списке. Перед сдачей задания на проверку обязательно проверьте, что скрипты можно запускать несколько раз и это не вызывает ошибки.
3. Задание представляет собой:
 - Query.sql - в нем должны быть приведены все требуемые запросы и примеры использования разработанных процедур и функций. В файле должны быть выполнены все необходимые действия для использования базы Northwind при запуске на выполнение этого файла. **Перед каждым запросом должен быть комментарий, содержащий номер запроса и текст описания задания. Если в задании есть вопрос, то ответ должен быть дан также в комментарии к запросу.**
 - Procedure.sql – в нем должны быть исходные текста процедур и функций задания. В файле должны быть выполнены все необходимые действия по инсталляции процедур и функций в базу Northwind при выполнении этого файла. Инсталляция должна учитывать тот факт, что процедуры с данным именем могут присутствовать в базе. **Перед каждой процедурой или функцией должен быть комментарий, соответствующий ее номеру в задании. ПРИМЕРЫ использования процедур должны быть приведены в файле Query.sql**
4. Написать письмо преподавателю, информирующее о том, что домашняя работа выполнена. Тема письма должна выглядеть так: [STYYYY_N]Фамилия_Имя_TaskNN. Здесь ST – сокращённое наименование программы (Students Training), YYYY – номер года, N – номер группы, NN – порядковый номер задания в виде двух цифр, напр. 01, 05, 12 и т.д.

Например, ST2017_1 Иванов Иван Task01.

Рекомендации к оформлению кода

1. При добавлении нового файла желательно вверху файла добавить блок с комментарием, для чего предназначен этот файл.