

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ
ФЕДЕРАЦИИ

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
«Национальный исследовательский технологический университет «МИСиС»

КАФЕДРА
НАПРАВЛЕНИЕ
ПРОФИЛЬ

Автоматизированных систем управления
09.03.01 «Информатика и вычислительная техника»
Интеллектуальные системы обработки информации и управления

ЛАБОРАТОРНАЯ РАБОТА
по дисциплине

Базы данных

на тему: «Множества группировок и поворот данных»

Студент(ка) ак. группы БИВТ-18-1 _____
аббревиатура _____ подпись _____Гузев В. Н.
И.О. Фамилия

Оценка с учетом защиты _____
оценка _____
дата _____

Преподаватель _____
подпись _____И.О. Фамилия

Москва 2021

Обзор

На этом семинаре Вы будете использовать множества группировок и оператор PIVOT для обобщения данных в БД AdventureWorksLT.

Задача 1: Получение итогов продаж по регионам

Компания Adventure Works продает товары клиентам, расположенным в различных странах / регионах по всему миру.

1. Получите итоги для страны/региона и штата/провинции.

В существующем отчете используется следующий запрос, чтобы вернуть общий доход от продаж, сгруппированный по стране/региону и штату/провинции.

```
SELECT a.CountryRegion, a.StateProvince, SUM(soh.TotalDue) AS Revenue
FROM SalesLT.Address AS a
INNER JOIN SalesLT.CustomerAddress AS ca ON a.AddressID = ca.AddressID
INNER JOIN SalesLT.Customer AS c ON ca.CustomerID = c.CustomerID
INNER JOIN SalesLT.SalesOrderHeader AS soh ON c.CustomerID = soh.CustomerID
GROUP BY a.CountryRegion, a.StateProvince
ORDER BY a.CountryRegion, a.StateProvince;
```

Вам необходимо изменить этот запрос, чтобы результаты включали общую сумму всех доходов от продаж и промежуточную сумму для каждой страны/региона в дополнение к промежуточным суммам по штатам/провинциям, которые уже возвращены.

Команда:

```
SELECT a.CountryRegion, a.StateProvince, SUM(soh.TotalDue) AS Revenue
FROM SalesLT.Address AS a
INNER JOIN SalesLT.CustomerAddress AS ca ON a.AddressID = ca.AddressID
INNER JOIN SalesLT.Customer AS c ON ca.CustomerID = c.CustomerID
INNER JOIN SalesLT.SalesOrderHeader AS soh ON c.CustomerID = soh.CustomerID
GROUP BY
ROLLUP (a.CountryRegion, a.StateProvince)
ORDER BY a.CountryRegion, a.StateProvince;
```

Результат:

	CountryRegion	StateProvince	Revenue
1	NULL	NULL	956303,5949
2	United Kingdom	NULL	572496,5594
3	United Kingdom	England	572496,5594
4	United States	NULL	383807,0355
5	United States	California	346517,6072
6	United States	Colorado	14017,9083
7	United States	Nevada	7330,8972
8	United States	New Mexico	15275,1977
9	United States	Utah	665,4251

2. Укажите уровни группировки в результатах.

Измените свой предыдущий запрос, чтобы включить столбец с именем «Level», который указывает, на каком уровне в иерархии отображается показатель дохода в строке: итоговом, страна/регион и штат/провинция. Например, общая итоговая строка должна содержать значение «Total», строка, показывающая промежуточный итог для Соединенных Штатов, должна содержать значение «United States Subtotal», а строка, показывающая промежуточный итог для Калифорнии, должна содержать значение «California Subtotal».

Команда:

```
SELECT a.CountryRegion
, a.StateProvince
, GROUPING_ID (a.CountryRegion, a.StateProvince) AS GroupingLevel
, SUM(soh.TotalDue) AS Revenue
FROM SalesLT.Address AS a
INNER JOIN SalesLT.CustomerAddress AS ca ON a.AddressID = ca.AddressID
INNER JOIN SalesLT.Customer AS c ON ca.CustomerID = c.CustomerID
INNER JOIN SalesLT.SalesOrderHeader AS soh ON c.CustomerID = soh.CustomerID
GROUP BY
GROUPING SETS (a.CountryRegion, a.StateProvince, ())
ORDER BY a.CountryRegion, a.StateProvince;
```

Результат:

	CountryRegion	StateProvince	GroupingLevel	Revenue
1	NULL	NULL	3	956303,5949
2	NULL	California	2	346517,6072
3	NULL	Colorado	2	14017,9083
4	NULL	England	2	572496,5594
5	NULL	Nevada	2	7330,8972
6	NULL	New Mexico	2	15275,1977
7	NULL	Utah	2	665,4251
8	United Kingdom	NULL	1	572496,5594
9	United States	NULL	1	383807,0355

3. Добавьте уровень группировки для городов.

Расширьте свой предыдущий запрос, чтобы включить столбец «City» (после столбца *StateProvince*) и группировку для отдельных городов. Измените правила для столбца «Level» так, чтобы строка, показывающая промежуточный итог для города Лондон, содержала значение «London Subtotal». Добавьте столбец *City* в выражение сортировки результатов.

Команда:

```
SELECT a.CountryRegion
, a.StateProvince
, City
, CHOOSE (GROUPING_ID(a.CountryRegion, a.StateProvince)
, city + ' subtotal'
, a.StateProvince + ' subtotal'
, a.CountryRegion + ' subtotal'
, 'Total') as GroupingLevel
, SUM(soh.TotalDue) AS Revenue
FROM SalesLT.Address AS a
INNER JOIN SalesLT.CustomerAddress AS ca ON a.AddressID = ca.AddressID
INNER JOIN SalesLT.Customer AS c ON ca.CustomerID = c.CustomerID
INNER JOIN SalesLT.SalesOrderHeader AS soh ON c.CustomerID = soh.CustomerID
GROUP BY
GROUPING SETS (a.CountryRegion, a.StateProvince, City, ())
ORDER BY a.CountryRegion, a.StateProvince, City;
```

Результат:

	CountryRegion	StateProvince	City	GroupingLevel	Revenue
23	NULL	NULL	Santa Ana	NULL	3673,3249
24	NULL	NULL	Santa Fe	NULL	15275,1977
25	NULL	NULL	Sherman Oaks	NULL	63686,2708
26	NULL	NULL	Union City	NULL	92663,5609
27	NULL	NULL	Van Nuys	NULL	43962,7901
28	NULL	NULL	West Sussex	NULL	42452,6519
29	NULL	NULL	Wokingham	NULL	87,0851
30	NULL	NULL	Woolston	NULL	119960,824
31	NULL	California	NULL	California subtotal	346517,6072
32	NULL	Colorado	NULL	Colorado subtotal	14017,9083
33	NULL	England	NULL	England subtotal	572496,5594
34	NULL	Nevada	NULL	Nevada subtotal	7330,8972
35	NULL	New Mexico	NULL	New Mexico subtotal	15275,1977
36	NULL	Utah	NULL	Utah subtotal	665,4251
37	United Kingd...	NULL	NULL	NULL	572496,5594
38	United States	NULL	NULL	NULL	383807,0355

Задача 2: Получение доходов от продаж по категориям

Продукты Adventure Works группируются по категориям, которые, в свою очередь, имеют родительские категории (определенные в представлении SalesLT.vGetAllCategories). Клиенты Adventure Works являются компаниями розничной торговли, и они могут размещать заказы на продукцию любой категории. Доход каждого товара в заказе записывается как значение в столбце LineTotal таблицы SalesLT.SalesOrderDetail.

1. Получите доход от продаж для каждой родительской категории.

Получите список названий компаний-клиентов (столбец CompanyName), их общий доход (сумма значений LineTotal в таблице SalesLT.SalesOrderDetail) для каждой родительской категории в разделах Accessories, Bikes, Clothing и Components (в соответствующих столбцах Accessories, Bikes, Clothing и Components). Отсортируйте результаты запроса по названию компании.

Команда:

```
SELECT CompanyName
, Bikes
, Accessories
, Clothing
FROM (SELECT c.CompanyName AS CompanyName
, sum(sod.LineTotal) AS Sum
, v.ParentProductName AS Category
FROM SalesLT.Customer AS c
JOIN SalesLT.SalesOrderHeader AS soh
ON c.CustomerID = soh.CustomerID
JOIN SalesLT.SalesOrderDetail AS sod
ON soh.SalesOrderID = sod.SalesOrderID
JOIN SalesLT.Product AS p
ON sod.ProductID = p.ProductID
JOIN SalesLT.vGetAllCategories AS v
ON p.ProductCategoryID = v.ProductCategoryID
GROUP BY v.ParentProductName, c.CompanyName) AS sales
PIVOT (SUM(Sum) FOR Category IN([Bikes], [Accessories], [Clothing])) AS pvt
```

Результат:

	CompanyName	Bikes	Accessories	Clothing
1	Action Bicycle Specialists	76613.651796	1299.885268	2461.657250
2	Aerobic Exercise Company	NULL	NULL	NULL
3	Bulk Discount Store	70597.284000	730.464000	851.562000
4	Central Bicycle Specialists	NULL	NULL	NULL
5	Channel Outlet	NULL	216.000000	308.664000
6	Closest Bicycle Store	20389.668000	NULL	559.164108
7	Coalition Bike Company	529.492800	NULL	124.776000
8	Discount Tours	2041.188000	72.000000	341.058000
9	Eastside Department Store	51096.054808	1220.235601	2772.229577
10	Engineered Bike Systems	2604.762000	NULL	178.746000
11	Essential Bike Works	NULL	NULL	37.758000
12	Extreme Riding Supplies	41337.936000	NULL	NULL
13	Futuristic Bikes	NULL	NULL	174.288000
14	Good Toys	NULL	NULL	NULL
15	Instruments and Parts Company	50771.952000	NULL	NULL
16	Many Bikes Store	49800.595200	770.700000	1659.378000
17	Metropolitan Bicycle Supply	50339.739600	20.994000	1149.044424

Выходы

В ходе лабораторной работы мы использовали множества группировок и оператор PIVOT для обобщения данных.