МИНОБРНАУКИ РОССИИ
САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ
ЭЛЕКТРОТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ
«ЛЭТИ» ИМ. В.И. УЛЬЯНОВА (ЛЕНИНА)
Кафедра САПР

ОТЧЕТ

по лабораторной работе №2

по дисциплине «Базы данных»

Тема: Группировка и агрегирование данных

Студент гр. 1302	_____	Новиков Г.В.
Студентка гр. 1302	_____	Романова О.В.
Преподаватель	_____	Новакова Н.Е.

Санкт-Петербург
2023

Цель работы.

Знакомство с опциями GROUP BY и HAVING, а также агрегированием данных. В лабораторной работе используется база данных AdventureWorks.

Порядок выполнения.

Упражнение 1 – использование ключевого слова TOP в команде SELECT

Запрос 1: из таблицы Sales.SalesPerson выводим значения полей SalesPersonID и Bonus. Запрос был отсортирован по полю Bonus по убыванию.

```
SELECT SalesPersonID, Bonus
FROM Sales.SalesPerson
ORDER BY Bonus DESC;
```

	SalesPersonID	Bonus
1	279	6700,00
2	290	5650,00
3	285	5150,00
4	280	5000,00
5	282	5000,00
6	275	4100,00
7	287	3900,00
8	281	3550,00
9	283	3500,00
10	277	2500,00
11	276	2000,00
12	286	985,00
13	278	500,00
14	289	75,00
15	268	0,00
16	288	0,00
17	284	0,00

Запрос успешно... | TULENCHI (15.0 RTM) | TULENCHI\olya- (62) | adventure | 00:00:00 | 17 строки

Рис. 1 – Результат выполнения запроса 1 из упражнения №1

Результаты | Сообщения

```
(затронуто строк: 17)

Время выполнения: 2023-10-12T01:12:11.6746491+03:00
```

100 %

Запрос успешно... | TULENCHI (15.0 RTM) | TULENCHI\olya- (62) | adventure | 00:00:00 | 17 строки

Рис. 2 – Сообщение о выполнении запроса 1 из упражнения №1

Запрос 2: модифицируем код запроса таким образом, чтобы возвращались только 4 записи о значениях самых больших премий (бонусов) для продавцов.

```sql
SELECT TOP(4) SalesPersonID, Bonus
FROM Sales.SalesPerson
ORDER BY Bonus DESC;
```

Результаты | Сообщения

	SalesPersonID	Bonus
1	279	6700,00
2	290	5650,00
3	285	5150,00
4	280	5000,00

Запрос успешно в... | TULENCHI (15.0 RTM) | TULENCHI\olya- (62) | adventure | 00:00:00 | 4 строки

Рис. 3 – Результат выполнения запроса 2 из упражнения №1

Результаты | Сообщения

```
(затронуто строк: 4)

Время выполнения: 2023-10-12T01:31:54.6252630+03:00
```

100 %

Запрос успешно в... | TULENCHI (15.0 RTM) | TULENCHI\olya- (62) | adventure | 00:00:00 | 4 строки

Рис. 4 – Сообщение о выполнении запроса 2 из упражнения №1

Запрос 3: модифицируем запрос из предыдущего задания так, чтобы он возвращал строки не только со значениями первых четырех самых больших премий для продавцов, но и данные по тем продавцам, чьи премии имеют то же значение, что и последнее значение, полученное в предыдущем запросе.

```sql
SELECT TOP(4) WITH TIES SalesPersonID, Bonus
FROM Sales.SalesPerson
ORDER BY Bonus DESC;
```

Результаты | Сообщения

	SalesPersonID	Bonus
1	279	6700,00
2	290	5650,00
3	285	5150,00
4	280	5000,00
5	282	5000,00

Запрос успешно в... | TULENCHI (15.0 RTM) | TULENCHI\olya- (62) | adventure | 00:00:00 | 5 строки

Рис. 5 – Результат выполнения запроса 3 из упражнения №1

Результаты | Сообщения

```
(затронуто строк: 5)

Время выполнения: 2023-10-12T01:49:13.2830609+03:00
```

100 %

Запрос успешно в... | TULENCHI (15.0 RTM) | TULENCHI\olya- (62) | adventure | 00:00:00 | 5 строки

Рис. 6 – Сообщение о выполнении запроса 3 из упражнения №1

Упражнение 2 – Использование агрегатных функций и конструкций GROUP BY и HAVING

Запрос 1: использование агрегатных функций

1.1. Подсчитаем общее количество строк в таблице Employee схемы HumanResources

```
SELECT COUNT(*)
FROM HumanResources.Employee;
```

Результаты | Сообщения

	(Отсутствует имя столбца)
1	290

Запрос успешно в... | TULENCHI (15.0 RTM) | TULENCHI\olya- (62) | adventure | 00:00:00 | 1 строки

Рис. 7 – Результат выполнения запроса 1.1 из упражнения №2

Результаты | Сообщения

```
(затронута одна строка)

Время выполнения: 2023-10-12T02:01:25.7242190+03:00
```

100 %

Запрос успешно в... | TULENCHI (15.0 RTM) | TULENCHI\olya- (62) | adventure | 00:00:00 | 1 строки

Рис. 8 – Сообщение о выполнении запроса 1.1 из упражнения №2

1.2. Подсчитаем общее количество сотрудников, имеющих менеджеров (поле ManagerID)

```
SELECT COUNT(ManagerID)
FROM HumanResources.Employee;
```

	(Отсутствует имя столбца)
1	289

Запрос успешно в... | TULENCHI (15.0 RTM) | TULENCHI\olya- (62) | adventure | 00:00:00 | 1 строки

Рис. 9 – Результат выполнения запроса 1.2 из упражнения №2

Результаты | Сообщения

```
Внимание! Значение NULL исключено в агрегатных или других операциях SET.

(затронута одна строка)

Время выполнения: 2023-10-12T02:12:54.0104419+03:00
```

100 %

Запрос успешно в... | TULENCHI (15.0 RTM) | TULENCHI\olya- (62) | adventure | 00:00:00 | 1 строки

Рис. 10 – Сообщение о выполнении запроса 1.2 из упражнения №2

Запрос 2: использование операции GROUP BY

2.1. Запрос к таблице Sales.SalesOrderDetail, подсчитывающий суммарное количество заказанного товара (поле OrderQty) для каждого продукта (поле ProductID)

```
SELECT ProductID, SUM(OrderQty)
AS SUM FROM Sales.SalesOrderDetail
GROUP BY ProductID;
```

Результаты | Сообщения

	ProductID	SUM
1	925	625
2	902	36
3	710	90
4	879	249
5	733	90
6	856	1616
7	756	346
8	779	2394
9	802	190
10	971	322

Запрос успешн... | TULENCHI (15.0 RTM) | TULENCHI\olya- (62) | adventure | 00:00:00 | 266 строки

Рис. 11 – Результат выполнения запроса 2.1 из упражнения №2

Результаты | Сообщения

```
(затронуто строк: 266)

Время выполнения: 2023-10-12T02:36:47.9913133+03:00
```

100 %

Запрос успешн... | TULENCHI (15.0 RTM) | TULENCHI\olya- (62) | adventure | 00:00:00 | 266 строки

Рис. 12 – Сообщение о выполнении запроса 2.1 из упражнения №2

2.2. Отсортировать результат запроса по суммарному количеству заказанного товара

```
SELECT ProductID, SUM(OrderQty)
AS SUM FROM Sales.SalesOrderDetail
GROUP BY ProductID
ORDER BY SUM DESC;
```

Результаты | Сообщения

	ProductID	SUM
1	712	8311
2	870	6815
3	711	6743
4	715	6592
5	708	6532
6	707	6266
7	864	4247
8	873	3865
9	884	3864
10	714	3636

Запрос успешн... | TULENCHI (15.0 RTM) | TULENCHI\olya- (62) | adventure | 00:00:00 | 266 строки

Рис. 13 – Результат выполнения запроса 2.2 из упражнения №2

Результаты | Сообщения

```
(затронуто строк: 266)

Время выполнения: 2023-10-12T02:47:13.3931048+03:00
```

100 %

Запрос успешн... | TULENCHI (15.0 RTM) | TULENCHI\olya- (62) | adventure | 00:00:00 | 266 строки

Рис. 14 – Сообщение о выполнении запроса 2.2 из упражнения №2

2.3. Модифицировать запрос так, чтобы в результирующий набор попадали только те товары, суммарное заказов, по которым не менее 2000

```
SELECT ProductID, SUM(OrderQty)
AS SUM FROM Sales.SalesOrderDetail
GROUP BY ProductID
HAVING SUM(OrderQty) >= 2000
ORDER BY SUM DESC;
```

Результаты | Сообщения

	ProductID	SUM
1	712	8311
2	870	6815
3	711	6743
4	715	6592
5	708	6532

Запрос успешно... | TULENCHI (15.0 RTM) | TULENCHI\olya- (62) | adventure | 00:00:00 | 38 строки

Рис. 15 – Результат выполнения запроса 2.3 из упражнения №2

Результаты | Сообщения

```
(затронуто строк: 38)

Время выполнения: 2023-10-12T02:52:31.0000221+03:00
```

100 %

Запрос успешно... | TULENCHI (15.0 RTM) | TULENCHI\olya- (62) | adventure | 00:00:00 | 38 строки

Рис. 16 – Сообщение о выполнении запроса 2.3 из упражнения №2

Запрос 3: Использование предложения GROUP BY для формирования нескольких групп

3.1. Запрос к таблице Sales.SalesOrderDetail, в списке SELECT которого должны быть представлены поля ProductID, SpecialOfferID, среднее значение цены за единицу товара (поле UnitPrice) и суммарное значение по полю LineTotal

```
SELECT ProductID, SpecialOfferID, AVG(UnitPrice) AS Price, SUM(LineTotal) AS Total
FROM Sales.SalesOrderDetail
GROUP BY ProductID, SpecialOfferID;
```

Результаты | Сообщения

	ProductID	SpecialOfferID	Price	Total
1	815	1	36,447	22013.988000
2	758	1	874,794	621103.740000
3	955	1	1923,6978	869708.736000
4	925	2	144,8782	1561.786996
5	954	14	1030,9491	197210.270400

Запрос успешн... | TULENCHI (15.0 RTM) | TULENCHI\olya- (62) | adventure | 00:00:00 | 484 строки

Рис. 17 – Результат выполнения запроса 3.1 из упражнения №2

Результаты | Сообщения

```
(затронуто строк: 484)

Время выполнения: 2023-10-12T03:05:37.9350834+03:00
```

100 %

Запрос успешн... | TULENCHI (15.0 RTM) | TULENCHI\olya- (62) | adventure | 00:00:00 | 484 строки

Рис. 18 – Сообщение о выполнении запроса 3.1 из упражнения №2

3.2. Отсортировать полученный результат по полю ProductID по возрастанию. Дать псевдонимы тем элементам списка SELECT, которые соответствуют агрегированным значениям. Выполнить группировку

```
SELECT ProductID, SpecialOfferID, AVG(UnitPrice) AS Price, SUM(LineTotal) AS Total
FROM Sales.SalesOrderDetail
GROUP BY ProductID, SpecialOfferID
ORDER BY ProductID ASC;
```

Результаты | Сообщения

	ProductID	SpecialOfferID	Price	Total
1	707	11	15,7455	2971.175850
2	707	8	16,8221	2452.662180
3	707	3	18,9272	2191.058910
4	707	1	31,3436	141271.252000
5	707	2	20,0556	8886.245452
6	708	8	16,8221	2316.403170

Запрос успешн... | TULENCHI (15.0 RTM) | TULENCHI\olya- (62) | adventure | 00:00:00 | 484 строки

Рис. 19 – Результат выполнения запроса 3.2 из упражнения №2

Результаты | Сообщения

```
(затронуто строк: 484)

Время выполнения: 2023-10-12T03:11:07.2273033+03:00
```

100 %

Запрос успешн... | TULENCHI (15.0 RTM) | TULENCHI\olya- (62) | adventure | 00:00:00 | 484 строки

Рис. 20 – Сообщение о выполнении запроса 3.2 из упражнения №2

Упражнение 3 – Использование операторов ROLLUP и CUBE

Запрос 1: использование оператора ROLLUP для создания сводных результатов

1.1. Написать запрос к таблице Sales.SalesPerson. В списке SELECT укажите поле SalesQuota и суммарное значение по полю SalesYTD. Выполнить группировку. Дать псевдоним TotalSalesYTD для суммы

```
SELECT SalesQuota, SUM(SalesYTD) AS TotalSalesYTD
FROM Sales.SalesPerson
GROUP BY SalesQuota;
```

Результаты | Сообщения

	SalesQuota	TotalSalesYTD
1	NULL	1533087,5999
2	250000,00	33461260,59
3	300000,00	9299677,9445

Запрос успешно в... | TULENCHI (15.0 RTM) | TULENCHI\olya- (62) | adventure | 00:00:00 | 3 строки

Рис. 21 – Результат выполнения запроса 1.1 из упражнения №3

Результаты | Сообщения

```
(затронуто строк: 3)

Время выполнения: 2023-10-12T03:20:39.5300916+03:00
```

100 %

Запрос успешно в... | TULENCHI (15.0 RTM) | TULENCHI\olya- (62) | adventure | 00:00:00 | 3 строки

Рис. 22 – Сообщение о выполнении запроса 1.1 из упражнения №3

1.2. Изменить запрос, чтобы получать свободный результат по полученной выборке. Дополнительно применить функцию GROUPING

```
SELECT SalesQuota, SUM(SalesYTD) AS TotalSalesYTD, GROUPING(SalesQuota) AS GROUPING
FROM Sales.SalesPerson
GROUP BY ROLLUP(SalesQuota);
```

Результаты | Сообщения

	SalesQuota	TotalSalesYTD	GROUPING
1	NULL	1533087,5999	0
2	250000,00	33461260,59	0
3	300000,00	9299677,9445	0
4	NULL	44294026,1344	1

Запрос успешно в... | TULENCHI (15.0 RTM) | TULENCHI\olya- (62) | adventure | 00:00:00 | 4 строки

Рис. 23 – Результат выполнения запроса 1.2 из упражнения №3

Результаты | Сообщения

```
(затронуто строк: 4)

Время выполнения: 2023-10-12T03:27:40.5444608+03:00
```

100 %

Запрос успешно в... | TULENCHI (15.0 RTM) | TULENCHI\olya- (62) | adventure | 00:00:00 | 4 строки

Рис. 24 – Сообщение о выполнении запроса 1.2 из упражнения №3

Строки со значениями NULL – промежуточные значения, последняя строка – общий итог.

Запрос 2: Использование оператора CUBE для создания сводных результатов

2.1. Запрос к таблице Sales.SalesOrderDetail. В списке SELECT указаны поле ProductID и сумма по полю LineTotal. Выводить необходимо только те значения, для которых UnitPrice < $5.00. Выполнить сортировку и группировку по полю ProductID

```
SELECT ProductID, SUM(LineTotal) AS TOTAL
FROM Sales.SalesOrderDetail
WHERE UnitPrice < 5
GROUP BY ProductID
ORDER BY ProductID;
```

Результаты | Сообщения

	ProductID	TOTAL
1	709	247.950000
2	712	3448.312275
3	870	28654.163327
4	873	8232.597632
5	875	2458.405400
6	877	11188.372080
7	921	15444.050000

Запрос успешно в... | TULENCHI (15.0 RTM) | TULENCHI\olya- (62) | adventure | 00:00:00 | 9 строки

Рис. 25 – Результат выполнения запроса 2.1 из упражнения №3

Результаты | Сообщения

```
(затронуто строк: 9)

Время выполнения: 2023-10-12T03:46:04.1708313+03:00
```

100 %

Запрос успешно в... | TULENCHI (15.0 RTM) | TULENCHI\olya- (62) | adventure | 00:00:00 | 9 строки

Рис. 26 – Сообщение о выполнении запроса 2.1 из упражнения №3

2.2. Модифицировать запрос, добавить оператор CUBE, а в группировку добавить поле OrderQty

```
SELECT ProductID, SUM(LineTotal) AS TOTAL
FROM Sales.SalesOrderDetail
WHERE UnitPrice < 5
GROUP BY CUBE(ProductID, OrderQty)
ORDER BY ProductID;
```

Результаты | Сообщения

	ProductID	TOTAL
13	NULL	503.943440
14	NULL	669.313736
15	NULL	1070.853300
16	NULL	807.616824
17	NULL	1068.246595
18	NULL	822.528000
19	NULL	1925.935760
20	NULL	1427.798064
21	NULL	1400.760000
22	NULL	1460.832000
23	NULL	1860.264000
24	NULL	1941.300000
25	NULL	1873.860000
26	NULL	833.124000
27	NULL	1466.154000
28	NULL	2059.752000
29	NULL	61159.530...
30	NULL	86579.210...
31	709	111.150000
32	709	136.800000
33	709	247.950000
34	712	97.247250
35	712	98.642775
36	712	103.881455
37	712	109.228500
38	712	112.734600
39	712	187.891000

Запрос успешн... | TULENCHI (15.0 RTM) | TULENCHI\olya- (62) | adventure | 00:00:00 | 119 строки

Рис. 27 – Результат выполнения запроса 2.2 из упражнения №3

Результаты | Сообщения

```
(затронуто строк: 119)

Время выполнения: 2023-10-12T03:50:15.8041115+03:00
```

100 %

Запрос успешн... | TULENCHI (15.0 RTM) | TULENCHI\olya- (62) | adventure | 00:00:00 | 119 строки

Рис. 28 – Сообщение о выполнении запроса 2.2 из упражнения №3

Выводы.

В ходе выполнения лабораторной работы мы изучили конструкции GROUP BY и HAVING (для формирования групп и др.), использовали агрегатные функции, операторы (ROLLUP, CUBE), функцию GROUPING, ключевое слово TOP и предложение WITH TIES в команде SELECT (для возвращения части отсортированных значений из результата).

Список используемых материалов.

1. Распределенные базы данных: Методические указания к лабораторным работам / Сост.: А. В. Горячев, Н. Е. Новакова. СПб.: Изд-во СПбГЭТУ «ЛЭТИ», 2008. 32 с