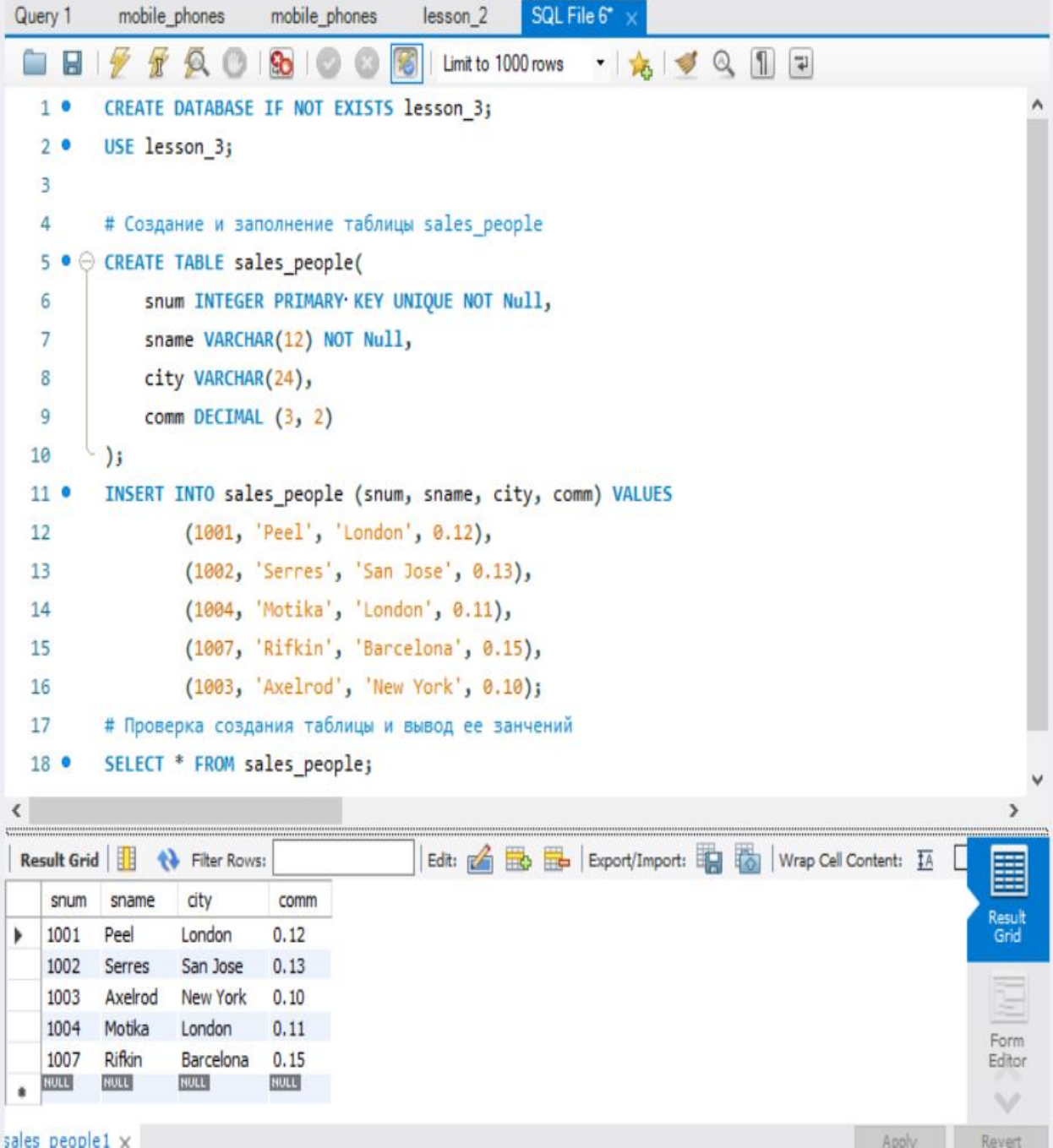


Домашняя работа 3

1. Создание таблиц для ДЗ

а. Sales_people



The screenshot shows a SQL IDE window with a query editor and a result grid. The query editor contains the following SQL code:

```
1 • CREATE DATABASE IF NOT EXISTS lesson_3;
2 • USE lesson_3;
3
4 # Создание и заполнение таблицы sales_people
5 • CREATE TABLE sales_people(
6     snum INTEGER PRIMARY KEY UNIQUE NOT Null,
7     sname VARCHAR(12) NOT Null,
8     city VARCHAR(24),
9     comm DECIMAL (3, 2)
10 );
11 • INSERT INTO sales_people (snum, sname, city, comm) VALUES
12     (1001, 'Peel', 'London', 0.12),
13     (1002, 'Serres', 'San Jose', 0.13),
14     (1004, 'Motika', 'London', 0.11),
15     (1007, 'Rifkin', 'Barcelona', 0.15),
16     (1003, 'Axelrod', 'New York', 0.10);
17 # Проверка создания таблицы и вывод ее значений
18 • SELECT * FROM sales_people;
```

The result grid displays the data inserted into the 'sales_people' table:

snum	sname	city	comm
1001	Peel	London	0.12
1002	Serres	San Jose	0.13
1003	Axelrod	New York	0.10
1004	Motika	London	0.11
1007	Rifkin	Barcelona	0.15
NULL	NULL	NULL	NULL







The IDE interface includes a toolbar with icons for file operations, a 'Limit to 1000 rows' dropdown, and a 'Result Grid' button on the right. The bottom status bar shows 'sales people1 x' and 'Apply' and 'Revert' buttons.

б. Customers

```

20      # Создание и заполнение таблицы customers
21 • CREATE TABLE IF NOT EXISTS customers(
22      cnum INTEGER PRIMARY KEY UNIQUE NOT Null,
23      cname VARCHAR(12) NOT Null,
24      city VARCHAR(24),
25      rating INTEGER,
26      snum INTEGER NOT Null,
27      FOREIGN KEY (snum) REFERENCES sales_people (snum)
28  );
29
30 • INSERT INTO customers (cnum, cname, city, rating, snum) VALUES
31      (2001, 'Hoffman', 'London', 100, 1001),
32      (2002, 'Giovanni', 'Rome', 200, 1003),
33      (2003, 'Liu', 'San Jose', 200, 1002),
34      (2004, 'Grass', 'Berlin', 300, 1002),
35      (2006, 'Clemens', 'London', 100, 1001),
36      (2008, 'Cisneros', 'San Jose', 300, 1007),
37      (2007, 'Pereira', 'Rome', 100, 1004);
38      # Проверка создания таблицы и вывод ее значений
39 • SELECT * FROM customers;
40

```

Result Grid					
Filter Rows: <input type="text"/>					
Edit:   					
Export/Import:  					
Wrap Cell Content: 					
	cnum	cname	city	rating	snum
▶	2001	Hoffman	London	100	1001
	2002	Giovanni	Rome	200	1003
	2003	Liu	San Jose	200	1002
	2004	Grass	Berlin	300	1002
	2006	Clemens	London	100	1001
	2007	Pereira	Rome	100	1004
	2008	Cisneros	San Jose	300	1007
*	NULL	NULL	NULL	NULL	NULL

c. orders

```

41 # Создание и заполнение таблицы orders
42 • CREATE TABLE orders(
43     onum INTEGER PRIMARY KEY UNIQUE NOT Null,
44     amt DECIMAL(7, 2),
45     odate DATE,
46     cnum INTEGER NOT Null,
47     snum INTEGER NOT Null,
48     FOREIGN KEY (snum) REFERENCES sales_people (snum),
49     FOREIGN KEY (cnum) REFERENCES customers (cnum)
50 );
51 • INSERT INTO orders (onum, amt, odate, cnum, snum) VALUES
52     (3001, 18.69, '1990-03-10', 2008, 1007),
53     (3003, 767.19, '1990-03-10', 2001, 1001),
54     (3002, 1900.10, '1990-03-10', 2007, 1004),
55     (3005, 5160.45, '1990-03-10', 2003, 1002),
56     (3006, 1098.16, '1990-03-10', 2008, 1007),
57     (3009, 1713.23, '1990-04-10', 2002, 1003),
58     (3007, 75.75, '1990-04-10', 2004, 1002),
59     (3008, 4723.95, '1990-05-10', 2006, 1001),
60     (3010, 1309.95, '1990-06-10', 2004, 1002),
61     (3011, 9891.88, '1990-06-10', 2006, 1001);
62 • SELECT * FROM orders;

```

Задания:

1. Напишите запрос, который вывел бы таблицу со столбцами в следующем порядке: city, sname, snum, comm. (к первой или второй таблице, используя SELECT)

```

64 # Напишите запрос, который вывел бы таблицу со столбцами в следующем порядке: city, sname, snum, comm. (к первой или второй таблице, используя SELECT)
65 • SELECT city, sname, snum, comm FROM sales_people;
66

```

city	sname	snum	comm
London	Peel	1001	0.12
San Jose	Serres	1002	0.13
New York	Axelrod	1003	0.10
London	Motika	1004	0.11
Barcelona	Rifkin	1007	0.15

2. Напишите команду SELECT, которая вывела бы оценку(rating), сопровождаемую именем каждого заказчика в городе San Jose. (“заказчики”)

```
66 # Напишите команду SELECT, которая вывела бы оценку(rating), сопровождаемую именем каждого заказчика в городе San Jose. (“заказчики”)
67 • SELECT rating, cname FROM customers WHERE city = 'San Jose';
68
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content: [↗](#)

	rating	cname
▶	200	Liu
	300	Cisneros

3. Напишите запрос, который вывел бы значения snum всех продавцов из таблицы заказов без каких бы то ни было повторений. (уникальные значения в “snum” “Продавцы”)

```
68 # Напишите запрос, который вывел бы значения snum всех продавцов из таблицы заказов без каких бы то ни было повторений. (уникальные значения в “snum” “Продавцы”)
69 • SELECT DISTINCT snum FROM orders;
70
71
72
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content: [↗](#)

	snum
▶	1001
	1002
	1003
	1004
	1007

4. Напишите запрос, который бы выбирал заказчиков, чьи имена начинаются с буквы G. Используется оператор "LIKE": (“заказчики”)

```
70 # Напишите запрос, который бы выбирал заказчиков, чьи имена начинаются с буквы G. Используется оператор "LIKE":
71 • SELECT cname FROM customers WHERE cname LIKE 'G%';
72
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content: [↗](#)

	cname
▶	Giovanni
	Grass

5. Напишите запрос, который может дать вам все заказы со значениями суммы выше чем \$1,000. (“Заказы”, “amt” - сумма)

```
72 # Напишите запрос, который может дать вам все заказы со значениями суммы выше чем $1,000. (“Заказы”, “amt” - сумма)
73 • SELECT onum, amt, odate AS odate FROM orders WHERE amt > 1000;
74
```

Result Grid | Filter Rows: | Edit: | Export/Import: | Wrap Cell Content: [↗](#)

	onum	amt	odate
▶	3002	1900.10	1990-03-10
	3005	5160.45	1990-03-10
	3006	1098.16	1990-03-10
	3008	4723.95	1990-05-10
	3009	1713.23	1990-04-10
	3010	1309.95	1990-06-10
	3011	9891.88	1990-06-10
*	NULL	NULL	NULL

6. Напишите запрос который выбрал бы наименьшую сумму заказа.

```
73 • SELECT onum, amt, odate AS odate FROM orders WHERE amt > 1000;
74 # Напишите запрос который выбрал бы наименьшую сумму заказа.
75 • SELECT MIN(amt) FROM orders;
76 # Напишите запрос к таблице “Заказчики”, который может показать всех заказчиков, у которых рейтинг больше 100 и они находятся не в Риме.
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content: [↗](#)

	MIN(amt)
▶	18.69

7. Напишите запрос к таблице “Заказчики”, который может показать всех заказчиков, у которых рейтинг больше 100 и они находятся не в Риме.

```
76 # Напишите запрос к таблице “Заказчики”, который может показать всех заказчиков, у которых рейтинг больше 100 и они находятся не в Риме.
77 • SELECT rating, cname, city FROM customers WHERE (NOT city = 'Rome') AND (rating > 100);
78
```

Result Grid	Filter Rows:	Export:	Wrap Cell Content:
rating	cname	city	
200	Liu	San Jose	
300	Grass	Berlin	
300	Cisneros	San Jose	

ЗАДАНИЕ 2

1. Создание таблицы

```
79 # Создание и заполнение талицы для части 2 домашнего задания 3
80 • CREATE TABLE workers (
81     id SERIAL PRIMARY KEY,
82     name VARCHAR (12) NOT Null,
83     surname VARCHAR (12) NOT Null,
84     speciality VARCHAR (12),
85     seniority INTEGER,
86     salary DECIMAL (8),
87     age INTEGER
88 );
89
90 • INSERT INTO workers (name, surname, speciality, seniority, salary, age) VALUES
91     ('Вася', 'Васькин', 'начальник', 40, 100000, 60),
92     ('Петя', 'Петькин', 'начальник', 8, 70000, 30),
93     ('Катя', 'Каткина', 'инженер', 2, 70000, 25),
94     ('Саша', 'Сашкин', 'инженер', 12, 50000, 35),
95     ('Иван', 'Иванов', 'рабочий', 40, 30000, 59),
96     ('Петр', 'Петров', 'рабочий', 20, 25000, 40),
97     ('Сидор', 'Сидоров', 'рабочий', 10, 20000, 35),
98     ('Антон', 'Антонов', 'рабочий', 8, 19000, 28),
99     ('Юра', 'Юркин', 'рабочий', 5, 15000, 25),
100     ('Максим', 'Воронин', 'рабочий', 2, 11000, 22),
101     ('Юра', 'Галкин', 'рабочий', 3, 12000, 24),
102     ('Люся', 'Люськина', 'уборщик', 10, 10000, 49);
103
104 • SELECT * FROM workers;
```

105

Result Grid

Filter Rows:

Edit:

Export/Import:

Wrap Cell Content:

id	name	surname	speciality	seniority	salary	age
1	Вася	Васькин	начальник	40	100000	60
2	Петя	Петькин	начальник	8	70000	30
3	Катя	Каткина	инженер	2	70000	25
4	Саша	Сашкин	инженер	12	50000	35
5	Иван	Иванов	рабочий	40	30000	59
6	Петр	Петров	рабочий	20	25000	40
7	Сидор	Сидоров	рабочий	10	20000	35
8	Антон	Антонов	рабочий	8	19000	28
9	Юра	Юркин	рабочий	5	15000	25
10	Максим	Воронин	рабочий	2	11000	22
11	Юра	Галкин	рабочий	3	12000	24

1. Отсортируйте поле “зарплата” в порядке убывания и возрастания

105 # Отсортируйте поле “зарплата” в порядке возрастания

106 • `SELECT * FROM workers ORDER BY salary;`

107 # Отсортируйте поле “зарплата” в порядке убывания

Result Grid | Filter Rows: | Edit: | Export/Import: | Wrap Cell Content: |

	id	name	surname	speciality	seniority	salary	age
▶	12	Люся	Люськина	уборщик	10	10000	49
	10	Максим	Воронин	рабочий	2	11000	22
	11	Юра	Галкин	рабочий	3	12000	24
	9	Юра	Юркин	рабочий	5	15000	25
	8	Антон	Антонов	рабочий	8	19000	28
	7	Сидор	Сидоров	рабочий	10	20000	35
	6	Петр	Петров	рабочий	20	25000	40
	5	Иван	Иванов	рабочий	40	30000	59
	4	Саша	Сашкин	инженер	12	50000	35
	2	Петя	Петькин	начальник	8	70000	30

107 # Отсортируйте поле “зарплата” в порядке убывания

108 • `SELECT * FROM workers ORDER BY salary DESC;`

109

Result Grid | Filter Rows: | Edit: | Export/Import: | Wrap Cell C

	id	name	surname	speciality	seniority	salary	age
▶	1	Вася	Васькин	начальник	40	100000	60
	2	Петя	Петькин	начальник	8	70000	30
	3	Катя	Каткина	инженер	2	70000	25
	4	Саша	Сашкин	инженер	12	50000	35
	5	Иван	Иванов	рабочий	40	30000	59
	6	Петр	Петров	рабочий	20	25000	40
	7	Сидор	Сидоров	рабочий	10	20000	35
	8	Антон	Антонов	рабочий	8	19000	28
	9	Юра	Юркин	рабочий	5	15000	25
	11	Юра	Галкин	рабочий	3	12000	24
	10	Максим	Воронин	рабочий	2	11000	22

2. Отсортируйте по возрастанию поле “Зарплата” и выведите 5 строк с наибольшей заработной платой (возможен подзапрос)

109 # Отсортируйте по возрастанию поле “Зарплата” и выведите 5 строк с наибольшей заработной платой

110 • `SELECT * FROM (SELECT * FROM workers ORDER BY salary DESC LIMIT 5) q ORDER BY q.salary;`

111

Result Grid | Filter Rows: | Export: | Wrap Cell Content: |

	id	name	surname	speciality	seniority	salary	age
▶	5	Иван	Иванов	рабочий	40	30000	59
	4	Саша	Сашкин	инженер	12	50000	35
	2	Петя	Петькин	начальник	8	70000	30
	3	Катя	Каткина	инженер	2	70000	25
	1	Вася	Васькин	начальник	40	100000	60

3. Выполните группировку всех сотрудников по специальности, суммарная зарплата которых превышает 100000

111 # Выполните группировку всех сотрудников по специальности , суммарная зарплата которых превышает 100000

112 • `SELECT speciality, COUNT(*) AS count, SUM(salary) AS accum FROM workers GROUP BY speciality HAVING SUM(salary) > 100000;`

Result Grid	Filter Rows:	Export:	Wrap Cell Content:
speciality	count	accum	
начальник	2	170000	
инженер	2	120000	
рабочий	7	132000	