

Отчет по лабораторной работы №1

Бакаев Никита, 13ПИ

ВАРИАНТ 6.

Уровень 1

1. Дана схема базы данных в виде следующих отношений. С помощью операторов SQL создать логическую структуру соответствующих таблиц для хранения в СУ БД, используя известные средства поддержания целостности (NOT NULL, UNIQUE, и т.д.). Обосновать выбор типов данных и используемые средства поддержания целостности. При выборе подходящих типов данных использовать информацию о конкретных значениях полей БД (см. прил.1)

ПОТРЕБИТЕЛЬ

ИДЕНТИФИКАТОР	НАЗВАНИЕ	АДРЕС ЖИТЕЛЬСТВА	СКИДКА, %
---------------	----------	------------------	-----------

ПОСТАВЩИК

ИДЕНТИФИКАТОР	ФАМИЛИЯ	АДРЕС	КОММИССИОННЫЕ, %
---------------	---------	-------	------------------

ЗАКАЗ

НОМЕР	ДАТА	ПОТРЕБИТЕЛЬ	ПОСТАВЩИК	ДЕТАЛЬ	КОЛ-ВО	СУММА, РУБ
-------	------	-------------	-----------	--------	--------	------------

ДЕТАЛЬ

ИДЕНТИФИКАТОР	НАИМЕНОВАНИЕ	СКЛАД	КОЛ-ВО	ЦЕНА, РУБ
---------------	--------------	-------	--------	-----------

2. Ввести в ранее созданные таблицы конкретные данные (см. прил. 1). Использовать скрипт-файл из операторов INSERT или вспомогательную утилиту.

2) SQL запрос на создание структуры таблиц и заполнение данными

```
1  -- -----
2  -- Table structure for customer
3  -- -----
4  DROP TABLE IF EXISTS "public"."customer";
5  CREATE TABLE "public"."customer" (
6    "id" int4 NOT NULL,
7    "name" varchar(255) COLLATE "default",
8    "address" varchar(255) COLLATE "default",
9    "sale" int4
10 )
11 WITH (OIDS=FALSE)
12
13 ;
14
15 -- -----
16 -- Records of customer
17 -- -----
18 INSERT INTO "public"."customer" VALUES ('1', 'АО ВАРЯ', 'Сормовский', '10');
19 INSERT INTO "public"."customer" VALUES ('2', 'ГАЗ', 'Автозаводский', '7');
20 INSERT INTO "public"."customer" VALUES ('3', 'МП ВЕРА', 'Канавинский', '5');
21 INSERT INTO "public"."customer" VALUES ('4', 'МП', 'Канавинский', '3');
22 INSERT INTO "public"."customer" VALUES ('5', 'АО СТАЛЬ', 'Советский', '0');
23
```

```

24  -- -----
25  -- Table structure for detail
26  -- -----
27  DROP TABLE IF EXISTS "public"."detail";
28  CREATE TABLE "public"."detail" (
29    "id" int4 NOT NULL,
30    "name" varchar(255) COLLATE "default",
31    "stock" varchar(255) COLLATE "default",
32    "quantity" int4,
33    "price" numeric(10,2) NOT NULL
34  )
35  WITH (OIDS=FALSE)
36
37  ;
38
39  -- -----
40  -- Records of detail
41  -- -----
42  INSERT INTO "public"."detail" VALUES ('1', 'Втулка', 'Сормовский', '20000', '5000.00');
43  INSERT INTO "public"."detail" VALUES ('2', 'Болт', 'Сормовский', '40000', '1000.00');
44  INSERT INTO "public"."detail" VALUES ('3', 'Ключ гаечный', 'Канавинский', '5000', '3000.00');
45  INSERT INTO "public"."detail" VALUES ('4', 'Шпилька', 'Автозаводский', '10000', '900.00');
46  INSERT INTO "public"."detail" VALUES ('5', 'Винт', 'Сормовский', '50000', '1500.00');
47  INSERT INTO "public"."detail" VALUES ('6', 'Молоток', 'Канавинский', '1200', '2000.00');
48  INSERT INTO "public"."detail" VALUES ('7', 'Шуруп', 'Сормовский', '30000', '1200.00');
49
50

```

```

50  -- -----
51  -- Table structure for order
52  -- -----
53  DROP TABLE IF EXISTS "public"."order";
54  CREATE TABLE "public"."order" (
55    "number" int4 NOT NULL,
56    "date" varchar COLLATE "default",
57    "customer" int4 NOT NULL,
58    "supplier" int4 NOT NULL,
59    "detail" int4,
60    "quantity" int4,
61    "price" numeric(10,2)
62  )
63  WITH (OIDS=FALSE)
64
65  ;
66
67  -- -----
68  -- Records of order
69  -- -----
70  INSERT INTO "public"."order" VALUES ('1', 'Январь', '5', '4', '3', '7', '21000.00');
71  INSERT INTO "public"."order" VALUES ('2', 'Февраль', '3', '3', '3', '2', '6000.00');
72  INSERT INTO "public"."order" VALUES ('3', 'Февраль', '4', '5', '4', '200', '180000.00');
73  INSERT INTO "public"."order" VALUES ('4', 'Март', '5', '4', '2', '50', '50000.00');
74  INSERT INTO "public"."order" VALUES ('5', 'Апрель', '1', '6', '7', '110', '132000.00');
75  INSERT INTO "public"."order" VALUES ('6', 'Апрель', '4', '4', '1', '150', '750000.00');
76  INSERT INTO "public"."order" VALUES ('7', 'Май', '2', '4', '6', '20', '40000.00');
77  INSERT INTO "public"."order" VALUES ('8', 'Июнь', '1', '3', '7', '2000', '240000.00');
78  INSERT INTO "public"."order" VALUES ('9', 'Июнь', '2', '5', '7', '10000', '1200000.00');
79  INSERT INTO "public"."order" VALUES ('10', 'Июнь', '3', '6', '1', '5', '25000.00');
80  INSERT INTO "public"."order" VALUES ('11', 'Июнь', '4', '3', '3', '1', '3000.00');
81  INSERT INTO "public"."order" VALUES ('12', 'Июнь', '4', '4', '1', '10', '50000.00');
82  INSERT INTO "public"."order" VALUES ('13', 'Июль', '1', '6', '6', '3', '6000.00');
83  INSERT INTO "public"."order" VALUES ('14', 'Июль', '2', '1', '2', '1000', '1000000.00');
84  INSERT INTO "public"."order" VALUES ('15', 'Июль', '2', '2', '1', '100', '5000000.00');
85  INSERT INTO "public"."order" VALUES ('16', 'Июль', '5', '1', '5', '100', '15000.00');
86  INSERT INTO "public"."order" VALUES ('17', 'Август', '1', '4', '7', '12000', '2440000.00');
87

```

```

88  -- -----
89  -- Table structure for supplier
90  -- -----
91  DROP TABLE IF EXISTS "public"."supplier";
92  CREATE TABLE "public"."supplier" (
93  "id" int4 NOT NULL,
94  "lastname" varchar(255) COLLATE "default",
95  "address" varchar(255) COLLATE "default",
96  "commission" int4
97  )
98  WITH (OIDS=FALSE)
99
100 ;
101
102  -- -----
103  -- Records of supplier
104  -- -----
105  INSERT INTO "public"."supplier" VALUES ('1', 'Артюхина', 'Сормовский', '4');
106  INSERT INTO "public"."supplier" VALUES ('2', 'Щепин', 'Приокский', '4');
107  INSERT INTO "public"."supplier" VALUES ('3', 'Власов', 'Канавинский', '5');
108  INSERT INTO "public"."supplier" VALUES ('4', 'Кузнецова', 'Советский', '5');
109  INSERT INTO "public"."supplier" VALUES ('5', 'Цепилева', 'Нижегородский', '3');
110  INSERT INTO "public"."supplier" VALUES ('6', 'Корнилов', 'Нижегородский', '6');
111

```

```

112  -- -----
113  -- Alter Sequences Owned By
114  -- -----
115
116  -- -----
117  -- Primary Key structure for table customer
118  -- -----
119  ALTER TABLE "public"."customer" ADD PRIMARY KEY ("id");
120
121  -- -----
122  -- Primary Key structure for table detail
123  -- -----
124  ALTER TABLE "public"."detail" ADD PRIMARY KEY ("id");
125
126  -- -----
127  -- Primary Key structure for table order
128  -- -----
129  ALTER TABLE "public"."order" ADD PRIMARY KEY ("number");
130
131  -- -----
132  -- Primary Key structure for table supplier
133  -- -----
134  ALTER TABLE "public"."supplier" ADD PRIMARY KEY ("id");
135
136  -- -----
137  -- Foreign Key structure for table "public"."order"
138  -- -----
139  ALTER TABLE "public"."order" ADD FOREIGN KEY ("customer") REFERENCES "public"."customer" ("id") ON DELETE NO ACTION ON UPDATE NO ACTION;
140  ALTER TABLE "public"."order" ADD FOREIGN KEY ("supplier") REFERENCES "public"."supplier" ("id") ON DELETE NO ACTION ON UPDATE NO ACTION;
141  ALTER TABLE "public"."order" ADD FOREIGN KEY ("detail") REFERENCES "public"."detail" ("id") ON DELETE NO ACTION ON UPDATE NO ACTION;
142

```


3. Используя оператор SELECT создать запрос для вывода всех строк каждой таблицы. Проверить правильность ввода. При необходимости произвести коррекцию значений операторами INSERT, UPDATE, DELETE.

Query Builder Query Editor

```
1 SELECT * from customer;
2 SELECT * from detail;
3 SELECT * from orders;
4 SELECT * from supplier;
```

Message	Result1	Result2	Result3	Result4
	id	name	address	sale
►	1	АО ВАРЯ	Сормовский	10
	2	ГАЗ	Автозаводский	7
	3	МП ВЕРА	Канавинский	5
	4	МП	Канавинский	3
	5	АО СТАЛЬ	Советский	0

Query Builder

Query Editor

```
1 SELECT * from customer;  
2 SELECT * from detail;  
3 SELECT * from orders;  
4 SELECT * from supplier;
```

Message

Result1

Result2

Result3

Result4

id	name	stock	quantity	price
1	Втулка	Сормовский	20000	5000
2	Болт	Сормовский	40000	1000
3	Ключ гаечный	Канавинский	5000	3000
4	Шпилька	Автозаводский	10000	900
5	Винт	Сормовский	50000	1500
6	Молоток	Канавинский	1200	2000
7	Шуруп	Сормовский	30000	1200

4. Создать запросы для вывода:
- a) всех различных размеров комиссионных;
 - b) всех различных фамилий поставщиков;
 - c) всех различных наименований деталей.

4a

The screenshot shows a PostgreSQL client interface. On the left is a tree view of the database structure. The 'public' schema is expanded, showing tables: 'customer', 'detail', 'orders', and 'supplier'. The 'supplier' table is selected. The main window has two tabs: 'Query Builder' and 'Query Editor'. The 'Query Editor' is active, showing a SQL query. Below the query editor is a 'Message' and 'Result1' pane. The 'Result1' pane shows the output of the query, which is a single column named 'commission' with four rows of values: 6, 4, 5, and 3.

Objects | customer @hse.public (192.16... | detail @hse.public

Run Stop Explain New Save Sa

Query Builder Query Editor

```
1 SELECT
2 "public".supplier.commission
3 FROM
4 "public".supplier
5 GROUP BY
6 "public".supplier.commission
7
```

Message Result1

commission
6
4
5
3

4b

The screenshot shows a PostgreSQL client interface. On the left is a tree view of the database structure. The 'public' schema is expanded, showing tables: 'customer', 'detail', 'orders', and 'supplier'. The 'supplier' table is selected. The main area is the 'Query Editor' with a SQL query:

```
1 SELECT
2 "public".supplier.lastname
3 FROM
4 "public".supplier
5 GROUP BY
6 "public".supplier.commission,
7 "public".supplier.lastname
8
```

Below the query editor is a 'Message' and 'Result1' pane. The 'Result1' tab is active, showing a table with one column 'lastname' and six rows of Russian surnames:

lastname
Власов
Артюхина
Кузнецова
Цепилева
Щепин
Корнилов

192.168.127.133 DOCKER

hse

public

Tables

customer

detail

orders

supplier

Views

Functions

Queries

Reports

Backups

postgres

Objects

customer @hse.public (192.16...

detail (

Run

Stop

Explain

New

Query Builder

Query Editor

1

SELECT

2

"public".detail."name"

3

FROM

4

"public".detail

5

GROUP BY

6

"public".detail."name"

7

Message

Result1

name

Втулка

Ключ гаечный

Молоток

Шпилька

Шуруп

Болт

Винт

5. Создав запрос получить следующую информацию:
- a) фамилии и адреса поставщиков, имеющих размер комиссионных менее 5%;
 - b) название и адрес склада для деталей, находящихся в количестве менее 1500 шт.;
 - c) название, адрес и размер скидки для предприятий, имеющих в названии слово "МП".
- Отсортировать по адресу и размеру скидки.

5a)

The screenshot shows the PostgreSQL Query Editor interface. On the left is the 'Catalog' tree with 'Tables' expanded, showing 'customer', 'detail', 'orders', and 'supplier'. The 'Query Editor' tab is active, displaying the following SQL query:

```
1 SELECT
2 "public".supplier.lastname,
3 "public".supplier.address
4 FROM
5 "public".supplier
6 WHERE
7 "public".supplier.commission < 5
8
```

Below the query editor, the 'Result' tab shows the query results in a table format:

lastname	address
Артюхина	Сормовский
Щепин	Приокский
Цепилева	Нижегородский

5b

The screenshot shows a PostgreSQL query editor interface. On the left is a database explorer tree with the following structure:

- public
 - Tables
 - customer
 - detail (selected)
 - orders
 - supplier
 - Views
 - Functions
 - Queries
 - Reports
 - Backups
- postgres

The main query editor window contains the following SQL query:

```
1 SELECT
2 "public".detail."name",
3 "public".detail.stock
4 FROM
5 "public".detail
6 WHERE
7 "public".detail.quantity < 1500
8
```

Below the query editor, the results are displayed in a table with two columns: 'name' and 'stock'. The first row of data is highlighted with a mouse cursor.

name	stock
Молоток	Канавинский

The screenshot displays a database management tool interface. On the left, a sidebar shows a tree view of the database structure under the connection '192.168.127.133 DOCKER'. The 'public' schema is expanded, showing tables: 'customer', 'detail', 'orders', and 'supplier'. The 'Queries' folder is selected. The main area is divided into a 'Query Editor' and a 'Result' pane. The 'Query Editor' contains the following SQL query:

```
1 SELECT
2 "public".customer."name",
3 "public".customer.address,
4 "public".customer.sale
5 FROM
6 "public".customer
7 WHERE
8 "public".customer."name" LIKE '%МП%'
9 ORDER BY
10 "public".customer.address ASC,
11 "public".customer.sale ASC
12
```

The 'Result' pane shows the output of the query in a table format:

name	address	sale
МП	Канавинский	3
МП ВЕРА	Канавинский	5

6. На основании данных о заказах вывести все данные в таком формате:
- номер, дата фамилия поставщика, сумма заказа. Отсортировать по фамилиям и сумме заказа;
 - название детали, количество, дата.

6a

The screenshot shows a PostgreSQL query editor interface. On the left is a sidebar with a tree view of the database structure: 'hse' (database), 'public' (schema), and 'Tables' (customer, detail, orders, supplier). The 'supplier' table is selected. The main window has tabs for 'Query Builder' and 'Query Editor'. The 'Query Editor' tab contains the following SQL query:

```
1 SELECT
2 "public".orders."number",
3 "public".orders."date",
4 "public".orders.price,
5 "public".supplier.lastname
6 FROM
7 "public".orders
8 LEFT JOIN "public".supplier
9 ON "public".orders."supplier"="public"."supplier"."id"
10 ORDER BY
11 "public".supplier."lastname" ASC,
12 "public".orders.price ASC
13
```

Below the query editor, there are tabs for 'Message' and 'Result1'. The 'Result1' tab displays the query results in a table with 4 columns: 'number', 'date', 'price', and 'lastname'. The results are sorted by 'lastname' and then 'price'.

number	date	price	lastname
16	Июль	15000	Артюхина
14	Июль	1000000	Артюхина
11	Июнь	3000	Власов
2	Февраль	6000	Власов
8	Июнь	2400000	Власов
13	Июль	6000	Корнилов
10	Июнь	25000	Корнилов
5	Апрель	132000	Корнилов
1	Январь	21000	Кузнецова
7	Май	40000	Кузнецова
4	Март	50000	Кузнецова
12	Июнь	50000	Кузнецова
6	Апрель	750000	Кузнецова
17	Август	24400000	Кузнецова
3	Февраль	180000	Цепилева
9	Июнь	12000000	Цепилева
15	Июль	5000000	Щепин

192.168.127.133 DOCKER

hse

public

Tables

customer

detail

orders

supplier

Views

Functions

Queries

Reports

Backups

postgres

Objects

customer @hse.publi...

*Untitled @hse.publi...

*Untitled @hse.publi...

Run

Stop

Explain

New

Save

Save As

Beautify SQL

Query Builder

Query Editor

```

1 SELECT
2 "public".detail."name",
3 "public".orders."date",
4 "public".detail.quantity
5 FROM
6 "public".orders
7 INNER JOIN "public".detail
8 ON "public".orders.detail = "public".detail."id"
9 ORDER BY
10 "public".orders.price ASC
11

```

Message

Result1

name	date	quantity
Ключ гаечный	Июнь	5000
Молоток	Июль	1200
Ключ гаечный	Февраль	5000
Винт	Июль	50000
Ключ гаечный	Январь	5000
Втулка	Июнь	20000
Молоток	Май	1200
Болт	Март	40000
Втулка	Июнь	20000
Шуруп	Апрель	30000
Шпилька	Февраль	10000
Втулка	Апрель	20000
Болт	Июль	40000
Шуруп	Июнь	30000
Втулка	Июль	20000
Шуруп	Июнь	30000
Шуруп	Август	30000

6b

7. Вывести:
- а) названия и размер скидки организаций-потребителей, куда поставлял детали Щепин, а общая сумма заказа превышала 5000;
 - б) фамилии и размер комиссионных для поставщиков, поставлявших детали предприятиям чужих районов не ранее января. Отсортировать по возрастанию комиссионных;
 - в) название и оставшееся количество деталей, заказывавшихся в количестве более 2 штук предприятиями Автомобильного и Советского районов. В вывод добавить суммарную стоимость соответствующих заказов;
 - г) названия предприятий одного района, заказывавших молотки.

7a

192.168.127.133 DOCKER

- hse
 - public
 - Tables
 - customer
 - detail
 - orders
 - supplier
 - Views
 - Functions
 - Queries
 - Reports
 - Backups
 - postgres

Objects | customer @hse,public (192.16... | orders @hse,public (192.168.12... | *Untitled @hse,public (192.16...

Run Stop Explain New Save Save As Beautify SQL Memo Export Result

Query Builder Query Editor

```
1 SELECT
2 "public".customer."name",
3 "public".customer.sale
4 FROM
5 "public".customer
6 INNER JOIN "public".orders ON "public".orders.customer = "public".customer."id"
7 WHERE
8 "public".orders.supplier = 2 AND
9 "public".orders.price > 5000
10
```

Message Result1

name	sale
ГАЗ	7

7b

192.168.127.133 DOCKER

- hse
 - public
 - Tables
 - customer
 - detail
 - orders
 - supplier
 - Views
 - Functions
 - Queries
 - Reports
 - Backups
 - postgres

Objects | detail @hse.public (192.168.127.133) | *Untitled @hse.public (192.168.127.133) | *Untitled @hse.public (192.168.127.133)

Run Stop Explain New Save Save As Beautify SQL Memo

Query Builder Query Editor

```

1 SELECT DISTINCT a."id", a."lastname", a."commission"
2 FROM (SELECT
3   "public".orders.detail,
4   "public".customer."name",
5   "public".customer.address,
6   "public".supplier.address,
7   "public".supplier.lastname,
8   "public".supplier.commission,
9   "public".supplier."id"
10  FROM
11   "public".orders
12  INNER JOIN "public".customer ON "public".orders.customer = "public".customer."id"
13  INNER JOIN "public".supplier ON "public".orders.supplier = "public".supplier."id"
14
15  WHERE "public".customer.address != "public".supplier.address
16  AND "public".orders."date" != 'Январь'
17 ) AS a
18 ORDER BY a."commission" ASC

```

Message Result1

id	lastname	commission
5	Цепилева	3
2	Щепин	4
1	Артюхина	4
3	Власов	5
4	Кузнецова	5
6	Корнилов	6

No activit

7c

public

- public
 - Tables
 - customer
 - detail
 - orders
 - supplier
 - Views
 - Functions
 - Queries
 - Reports
 - Backups
- postgres

Run Stop Explain New Save Save As Beautify SQL Memo Export Result

Query Builder Query Editor

```

1 SELECT
2   "public".detail."name",
3   A.detail,
4   Sum(A.price) orders_price,
5   Sum(A.quantity) details_quantity,
6   ("public".detail."quantity" - A."quantity") as balance
7 FROM
8   "public".orders as A
9  INNER JOIN "public".detail ON A.detail = "public".detail."id"
10 WHERE
11  A.customer IN ((SELECT id FROM "public".customer WHERE address IN ('Автозаводский', 'Советский')) AND
12  A.quantity > 2
13 GROUP BY A."detail", "public".detail."name", balance
14

```

Message Result1

name	detail	orders_price	details_quantity	balance
Шуруп	7	12000000	10000	20000
Болт	2	1000000	1000	39000
Ключ гаечный	3	21000	7	4993
Молоток	6	40000	20	1180
Втулка	1	5000000	100	19900
Винт	5	15000	100	49900
Болт	2	50000	50	39950

N

7d

192.168.127.133 DOCKER

hse

public

Tables

customer

detail

orders

supplier

Views

Functions

Queries

Reports

Backups

postgres

Objects

orders @hse.public (192.168.12...

detail @hse.public (192.168.12...

*Untitled @hse.public (192.16...

Run

Stop

Explain

New

Save

Save As

Beautify SQL

Memo

Export Result

Query Builder

Query Editor

1

2

3

4

5

6

7

8

9

SELECT

array_to_string(array_agg("public".customer."name") , ',') as customers_names,

"public".customer."address"

FROM

"public".orders

INNER JOIN "public".customer ON "public".orders.customer = "public".customer."id"

WHERE

"public".orders.detail = 6

GROUP BY "public".customer."address"

Message

Result1

customers_names	address
АО ВАРЯ	Сормовский
ГАЗ	Автозаводский

8. Создать запрос для модификации всех значений столбца с суммарной величиной платы, чтобы он содержал истинную сумму, которую заплатил потребитель (с учетом скидки). Вывести новые значения.

The screenshot shows a PostgreSQL query editor interface. On the left is a sidebar with a tree view of the database structure, including tables like 'customer', 'detail', 'orders', and 'supplier'. The main area is divided into a 'Query Builder' tab and a 'Query Editor' tab. The 'Query Editor' contains the following SQL query:

```
1 UPDATE orders
2 SET price = (orders.price * (1 - (P.sale / 100.0) ) )
3 FROM orders AS all_orders
4 INNER JOIN "public".customer AS P ON all_orders."customer" = P."id";
5
6 SELECT * FROM orders
```

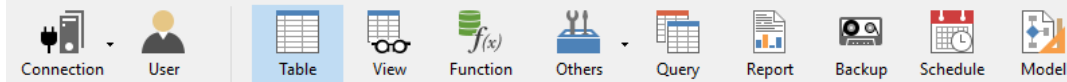
Below the query editor, there is a 'Message' and 'Result1' tab. The 'Result1' tab displays a table with 8 columns: 'number', 'date', 'customer', 'supplier', 'detail', 'quantity', and 'price'. The table contains 17 rows of data, representing orders from January to August.

number	date	customer	supplier	detail	quantity	price
1	Январь	5	4	3	7	18900
2	Февраль	3	3	3	2	5400
3	Февраль	4	5	4	200	162000
4	Март	5	4	2	50	45000
5	Апрель	1	6	7	110	118800
6	Апрель	4	4	1	150	675000
7	Май	2	4	6	20	36000
8	Июнь	1	3	7	2000	2160000
9	Июнь	2	5	7	10000	10800000
10	Июнь	3	6	1	5	22500
11	Июнь	4	3	3	1	2700
12	Июнь	4	4	1	10	45000
13	Июль	1	6	6	3	5400
14	Июль	2	1	2	1000	900000
15	Июль	2	2	1	100	4500000
16	Июль	5	1	5	100	13500
17	Август	1	4	7	12000	21960000

9. Расширить таблицу с данными о заказах столбцом, содержащим величину комиссионных поставщика. Создать запрос для ввода конкретных значений во все строки таблицы.

Untitled @hse.public (192.168.127.133 DOCKER) - Query - Navicat Premium

File View Favorites Tools Window Help



192.168.127.133 DOCKER

- hse
 - public
 - Tables
 - customer
 - detail
 - orders
 - supplier
 - Views
 - Functions
 - Queries
 - Reports
 - Backups
 - postgres

Objects * Untitled @hse.public (192.16... orders @hse.public (192.168.12... orders @hse.public (192.168.12...

Run Stop Explain New Save Save As Beautify SQL Export Result

Query Builder Query Editor

```
1 ALTER TABLE "public"."orders"
2 ADD COLUMN "commission" int4 ;
3
4 UPDATE orders
5 set commission = P.commission from supplier as P WHERE orders."supplier" = P."id"
6
```

Message

[SQL]ALTER TABLE "public"."orders"
ADD COLUMN "commission" int4 ;

UPDATE orders
set commission = P.commission from supplier as P WHERE orders."supplier" = P."id"

Time: 0.005s

Affected rows: 0

Affected rows: 17

Connection	User	Table	view	Function	Others	Query	Report	Backup	Schedule	Model
192.168.127.133 DOCKER		Objects	* Untitled @hse.public (192.168.127.133) orders @hse.public (192.168.127.133) orders @hse.public (192.168.127.133)							
hse		Begin Transaction	Memo Filter Sort Import Export							
public		number	date	customer	supplier	detail	quantity	price	commission	
Tables		1	Январь	5	4	3	7	18900	5	
customer		2	Февраль	3	3	3	2	5400	5	
detail		3	Февраль	4	5	4	200	162000	3	
orders		4	Март	5	4	2	50	45000	5	
supplier		5	Апрель	1	6	7	110	118800	6	
Views		6	Апрель	4	4	1	150	675000	5	
Functions		7	Май	2	4	6	20	36000	5	
Queries		8	Июнь	1	3	7	2000	2160000	5	
Reports		9	Июнь	2	5	7	10000	10800000	3	
Backups		10	Июнь	3	6	1	5	22500	6	
postgres		11	Июнь	4	3	3	1	2700	5	
		12	Июнь	4	4	1	10	45000	5	
		13	Июль	1	6	6	3	5400	6	
		14	Июль	2	1	2	1000	900000	4	
		15	Июль	2	2	1	100	4500000	4	
		16	Июль	5	1	5	100	13500	4	
		17	Август	1	4	7	12000	21960000	5	

Пример после изменения (столбец с комиссией)

Уровень 2

10. Используя операцию IN (NOT IN) реализовать следующие запросы:

- найти всех потребителей, заказывавших болты или винты не менее двух раз;
- найти потребителей, не делавших заказов на сумму менее 500000руб. поставщикам из своего района ;
- запросы задания 7.c и 7.d.

10a)

The screenshot shows a PostgreSQL query editor interface. On the left, a tree view displays the database structure: 'My Connections' -> '192.168.127.133 DOCKER' -> 'hse' -> 'public' -> 'Tables' (containing 'customer', 'detail', 'orders', 'supplier'). The 'Queries' folder is selected. The main area is the 'Query Editor' with the following SQL query:

```
1 SELECT * from (  
2  
3 SELECT  
4 "public".orders.detail,  
5 "public".customer."name",  
6 COUNT(*) as count_all  
7 FROM  
8 "public".orders  
9 INNER JOIN "public".customer ON "public".orders.customer = "public".customer."id"  
10 WHERE  
11 "public".orders.detail IN (2, 5)  
12  
13 GROUP BY "public".orders.detail, "public".customer."name"  
14 ) as P WHERE P.count_all >= 2
```

Below the query editor, the 'Result1' tab shows the query results in a table:

detail	name	count_all
(Null)	(Null)	(Null)

Таких покупателей на такие детали нет. Если добавить другие детали, то будет

192.168.127.133 DOCKER

- hse
 - public
 - Tables
 - customer
 - detail
 - orders
 - supplier
 - Views
 - Functions
 - Queries
 - Reports
 - Backups
 - postgres

Objects: detail @hse.public (192.168.127...), orders @hse.public (192.168.127...), 10-a @hse.public (192.168.127...)

Run Stop Explain New Save Save As Beautify SQL Memo Export Result

Query Builder Query Editor

```
1 SELECT * from (  
2  
3 SELECT  
4 "public".orders.detail,  
5 "public".customer."name",  
6 COUNT(*) as count_all  
7 FROM  
8 "public".orders  
9 INNER JOIN "public".customer ON "public".orders.customer = "public".customer."id"  
10 WHERE  
11 "public".orders.detail IN (1,2, 4)  
12  
13 GROUP BY "public".orders.detail, "public".customer."name"  
14 ) as P WHERE P.count_all >= 2
```

Message Result1

detail	name	count_all
1	МП	2

(пример с добавлением, чтобы был результат)

10b

192.168.127.133 DOCKER

- hse
 - public
 - Tables
 - customer
 - detail
 - orders
 - supplier
 - Views
 - Functions
 - Queries
 - Reports
 - Backups
 - postgres

Query Builder Query Editor

```

1 SELECT *
2 FROM
3 "public".customer
4 INNER JOIN "public".orders ON "public".orders.customer = "public".customer."id"
5 INNER JOIN "public".supplier ON "public".orders.supplier = "public".supplier."id"
6 WHERE
7 "public".orders.price >= 500000 and "public".customer.address IN ( "public".supplier.address )
8

```

10c

- public
 - Tables
 - customer
 - detail
 - orders
 - supplier
 - Views
 - Functions
 - Queries
 - Reports
 - Backups
- postgres

Run Stop Explain New Save Save As Beautify SQL Memo Export Result

Query Builder Query Editor

```

1 SELECT
2 "public".detail."name",
3 A.detail,
4 Sum(A.price) orders_price,
5 Sum(A.quantity) details_quantity,
6 ("public".detail."quantity" - A."quantity") as balance
7 FROM
8 "public".orders as A
9 INNER JOIN "public".detail ON A.detail = "public".detail."id"
10 WHERE
11 A.customer IN ((SELECT id FROM "public".customer WHERE address IN ('Автозаводский', 'Советский') )) AND
12 A.quantity > 2
13 GROUP BY A."detail", "public".detail."name", balance
14

```

Message Result1

name	detail	orders_price	details_quantity	balance
Шуруп	7	12000000	10000	20000
Болт	2	1000000	1000	39000
Ключ гаечный	3	21000	7	4993
Молоток	6	40000	20	1180
Втулка	1	5000000	100	19900
Винт	5	15000	100	49900
Болт	2	50000	50	39950

My Connections

- 192.168.127.133 DOCKER
 - hse
 - public
 - Tables
 - customer
 - detail
 - orders
 - supplier
 - Views
 - Functions
 - Queries
 - Reports
 - Backups
 - postgres

Query Builder Query Editor

```

1 SELECT
2 array_to_string(array_agg(customer."name"), ', ') as customer_names,
3 customer.address
4 FROM
5 orders
6 INNER JOIN customer on orders.customer = customer.id
7 WHERE orders.detail in (6)
8 GROUP BY customer.address

```

Message Result1

customer_names	address
АО ВАРЯ	Сормовский
ГАЗ	Автозаводский

11. Используя операции ALL- ANY реализовать следующие запросы:

- найти поставщика с наименьшими комиссионными, который в мае поставлял детали потребителю, сделавшему заказ максимальной стоимости в апреле;
- найти деталь у которой цена совпадает с ценой какой-либо (но не той же самой) детали, проданной поставщиком из Советского района с максимальными комиссионными;
- найти потребителя, который имеет не максимальный размер скидки и покупал детали у поставщиков из Канавинского района;
- запрос задания 7.а.

11a

192.168.127.133 DOCKER

- hse
 - public
 - Tables
 - customer
 - detail
 - orders
 - supplier
 - Views
 - Functions
 - Queries
 - Reports
 - Backups
 - postgres

Query Builder Query Editor

```

1 SELECT * from supplier WHERE commission <= ALL
2 ( SELECT commission from orders WHERE price >= ANY (SELECT price FROM orders WHERE date = 'Апрель' ) )

```

11b

Query Builder Query Editor

```

1 select id, name, "price" from detail where id IN (
2
3 SELECT detail from orders WHERE detail IN (
4
5 select id from detail as P
6 WHERE price = ANY ( SELECT price from detail WHERE id <> P."id")
7
8 ) and supplier in ( SELECT id from supplier WHERE address = 'Советский' )
9 and commission >= ALL (SELECT commission from supplier WHERE address = 'Советский')
10 ) limit 1 OFFSET 1

```

11c

Query Builder Query Editor

```

1 SELECT * from customer WHERE sale < ANY (SELECT sale from customer )
2 and id IN (
3 SELECT customer from orders WHERE supplier in (
4 SELECT id from supplier WHERE address = 'Канавинский'
5 )
6 )

```

Message Result1

id	name	address	sale
3	МП ВЕРА	Канавинский	5
4	МП	Канавинский	3

11d

my connections

192.168.127.133 DOCKER

hse

public

Tables

customer

detail

orders

supplier

Views

Functions

Queries

10-a

10d(7a)

11-b

11-c

11d

12

13-a

13-b

13-c

13-d

14-a

14-b

14-c

15-b

Query Builder

Query Editor

1

SELECT * from customer as customer_all WHERE id = ANY

2

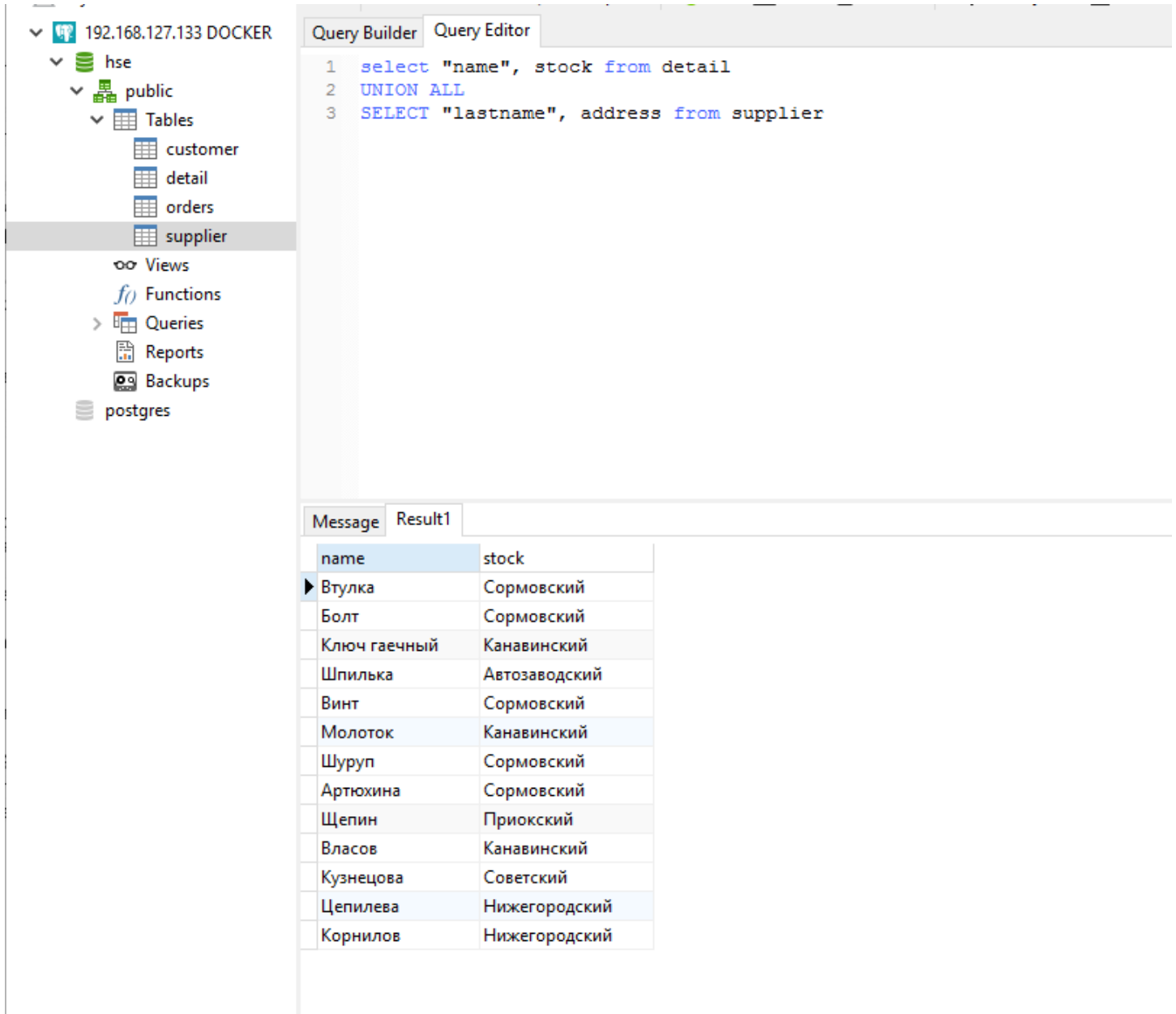
(SELECT customer from orders WHERE supplier = 2 and price > 5000 and customer = customer_all.id)

Message

Result1

id	name	address	sale
2	ГАЗ	Автозаводский	7

12. Используя операцию UNION получить места складирования деталей и места расположения поставщиков.



The screenshot displays a database management interface. On the left, a tree view shows the database structure: 192.168.127.133 DOCKER, hse, public, Tables (customer, detail, orders, supplier), Views, Functions, Queries, Reports, Backups, and postgres. The 'supplier' table is selected. The main area shows the 'Query Editor' with the following SQL query:

```
1 select "name", stock from detail
2 UNION ALL
3 SELECT "lastname", address from supplier
```

Below the query editor, the 'Result1' tab shows the query results in a table:

name	stock
Втулка	Сормовский
Болт	Сормовский
Ключ гаечный	Канавинский
Шпилька	Автозаводский
Винт	Сормовский
Молоток	Канавинский
Шуруп	Сормовский
Артюхина	Сормовский
Щепин	Приокский
Власов	Канавинский
Кузнецова	Советский
Цепилева	Нижегородский
Корнилов	Нижегородский

13. Используя операцию EXISTS (NOT EXISTS) реализовать нижеследующие запросы. В случае, если для текущего состояния БД запрос будет выдавать пустое множество строк, требуется указать какие добавления в БД необходимо провести.

- определить потребителей, заказывавших детали с ценой более 6000руб. у всех поставщиков из Советского или Канавинского районов;
- найти деталь, которую заказывали в количестве одной штуки все потребители;
- какие детали не заказывали потребители с размером скидки менее 5%;
- найти потребителя, заказывавшего все детали, не поставляемые поставщиками из Сормовского района.

13a

The screenshot shows a SQL IDE interface. On the left is a tree view of the database structure. The main window is the 'Query Editor' with a SQL query. Below the editor is a 'Result' pane showing the output of the query.

Database Structure (Left Panel):

- My Connections
 - 192.168.127.133 DOCKER
 - hse
 - public
 - Tables
 - customer
 - detail
 - orders
 - supplier
 - Views
 - Functions
 - Queries
 - 10-a
 - 10d(7a)
 - 11-b
 - 11-c
 - 11d
 - 12
 - 13-a (selected)
 - 13-b
 - 13-c
 - 13-d
 - 14-a
 - 14-b
 - 14-c
 - 15-b
 - 15-c
 - 15-d
 - 7-c
 - 7-d
 - 8
 - 9

SQL Query (Query Editor):

```
1 SELECT * from customer as all_customer WHERE EXISTS(  
2 SELECT * from orders WHERE customer = all_customer."id"  
3 and detail in (SELECT id from detail WHERE price > 6000)  
4 and supplier in (select id from supplier WHERE address in ('Советский', 'Канавинский'))  
5 )  
6 )
```

Result (Result1):

id	name	address	sale
(Null)	(Null)	(Null)	(Null)

Чтобы результат не был пустым нужны товары дороже 6000 (сейчас самая дорогая деталь - 5000)

13b

My Connections

- 192.168.127.133 DOCKER
 - hse
 - public
 - Tables
 - customer
 - detail
 - orders
 - supplier
 - Views
 - Functions
 - Queries
 - 10-a
 - 10d(7a)
 - 11-b
 - 11-c
 - 11d
 - 12
 - 13-a
 - 13-b
 - 13-c
 - 13-d
 - 14-a
 - 14-b
 - 14-c
 - 15-b
 - 15-c
 - 15-d

Query Editor

```

1 SELECT * from detail as all_details WHERE EXISTS
2 (
3 SELECT detail, customer from orders WHERE quantity = 1
4 and detail = all_details.id
5 GROUP BY detail, customer
6 HAVING COUNT(*) >= (SELECT count(*) from customer)
7 )

```

Message Result1

id	name	stock	quantity	price
(Null)	(Null)	(Null)	(Null)	(Null)

Такой детали нет; нужно добавить для каждого клиента (customer) заказ (order) в котором будет деталь с одним общим для все клиентов id

13c

My Connections

- 192.168.127.133 DOCKER
 - hse
 - public
 - Tables
 - customer
 - detail
 - orders
 - supplier
 - Views
 - Functions
 - Queries
 - 10-a
 - 10d(7a)
 - 11-b
 - 11-c
 - 11d
 - 12
 - 13-a
 - 13-b
 - 13-c
 - 13-d
 - 14-a
 - 14-b
 - 14-c
 - 15-b
 - 15-c
 - 15-d
 - 7-c
 - 7-d
 - 8
 - 9

Query Editor

```

1 SELECT * from detail as all_details WHERE NOT EXISTS
2 (SELECT * from orders where detail = all_details.id and customer in (SELECT id from customer as all_customers WHERE sale < 5)
3 )
4
5 -- SELECT * from customer as all_customers WHERE sale < 5

```

Message Result1

id	name	stock	quantity	price
6	Молоток	Канавинск	1200	2000
7	Шурпы	Сормовск	30000	1200

My Connections

192.168.127.133 DOCKER

hse

public

Tables

customer

detail

orders

supplier

Views

Functions

Queries

10-a

10d(7a)

11-b

11-c

11d

12

13-a

13-b

13-c

13-d

14-a

14-b

14-c

15-b

15-c

15-d

7-c

7-d

8

9

orders

Run Stop Explain New Save Save As Beautify SQL Memo Export Result

Query Builder

Query Editor

```
1 SELECT * from customer as all_customers WHERE EXISTS
2 ( SELECT customer, detail from orders WHERE customer = all_customers.id and detail
3 in
4 (SELECT id from detail as all_details WHERE NOT EXISTS
5 ( SELECT detail from orders WHERE supplier not in (SELECT id from supplier WHERE address = 'Сормовский' )
6 and detail = all_details.id
7 )
8 )
9 GROUP BY customer, detail
10 HAVING count(*) >= (SELECT count(*) from detail as all_details WHERE NOT EXISTS
11 ( SELECT detail from orders WHERE supplier not in (SELECT id from supplier WHERE address = 'Сормовский' )
12 and detail = all_details.id
13 )
14 )
15 )
16 )
17
18
```

Message Result1

id	name	address	sale
5	АО СТА/ль	Советский	0

14. Реализовать запросы с использованием агрегатных функций:

- определить суммарную стоимость всех заказов, произведенных потребителями из Канавинского района;
- найти среднее число заказываемых деталей со ценой более 2000;
- найти максимальную скидку среди потребителей, заказывавших детали у поставщиков из своего района;
- какие детали имеют цену за штуку меньше средней.

14a

The screenshot shows the Navicat Cloud interface. On the left, the 'My Connections' tree shows a connection to '192.168.127.133 DOCKER' with a database 'hse' containing tables 'customer', 'detail', 'orders', and 'supplier'. The 'Query Editor' is active, displaying the following SQL query:

```
1 SELECT sum(price) as all_orders_amount from orders WHERE customer in (SELECT id from customer WHERE address = 'Канавинский')
```

The 'Message' pane at the bottom shows the result of the query:

all_orders_amount
912600

14b

The screenshot shows the Navicat Cloud interface. On the left, the 'My Connections' tree shows a connection to '192.168.127.133 DOCKER' with a database 'hse' containing tables 'customer', 'detail', 'orders', and 'supplier'. The 'Query Editor' is active, displaying the following SQL query:

```
1 SELECT avg(quantity) from orders WHERE price > 2000
```

The 'Message' pane at the bottom shows the result of the query:

avg
1515.17647058824

14c

My Connections

- 192.168.127.133 DOCKER
 - hse
 - public
 - Tables
 - customer
 - detail
 - orders
 - supplier
 - Views
 - Functions
 - Queries
 - Reports
 - Backups
 - postgres

Query Builder Query Editor

```
1 SELECT * from (  
2 SELECT * from customer  
3 left OUTER JOIN (SELECT MAX(sale) max_sale_per_address from customer  
4 where id IN (  
5  
6 SELECT  
7 "public".customer.id  
8 FROM  
9 "public".customer ,  
10 "public".supplier  
11 INNER JOIN "public".orders ON "public".orders.customer = "public".customer."id" AND "public".  
12  
13 WHERE customer.address = supplier.address  
14 -)  
15 GROUP BY address ) sub on sub.max_sale_per_address = customer.sale) P  
16 WHERE P.max_sale_per_address IS NOT NULL
```

Message Result1

id	name	address	sale	max_sale_per_address
3	МП ВЕРА	Канавинский	5	5
5	АО СТАЛЬ	Советский	0	0

14d

Navicat Cloud

My Connections

192.168.127.133 DOCKER

- hse
 - public
 - Tables
 - customer
 - detail
 - orders
 - supplier
 - Views
 - Functions
 - Queries
 - Reports
 - Backups
 - postgres

Objects | orders @hse.public (192.168.12... | * Untitled @hse.public (192.16... | * Untitled @hse.public (192.16... | * 14-b @hse.public (192.168...

Run | Stop | Explain | New | Save | Save As | Beautify SQL | Memo | Export Result

Query Builder | Query Editor

1 SELECT * from detail WHERE price < ((SELECT avg(price) from detail))

Message | Result1

	id	name	stock	quantity	price
▶	2	Болт	Сормовский	40000	1000
	4	Шпилька	Автозаводский	10000	900
	5	Винт	Сормовский	50000	1500
	6	Молоток	Канавинский	1200	2000
	7	Шуруп	Сормовский	30000	1200

15. Используя средства группировки реализовать следующие запросы:

- найти для каждой пары "потребитель-поставщик" суммарную величину стоимости произведенных заказов;
- найти детали, которые более трех раз заказывали потребители из Советского района;
- найти месяц, в котором все заказы имели стоимость не менее 10000;
- получить для каждой детали со ценой более 10000 среднее количество заказываемых деталей.

15a

The screenshot shows a PostgreSQL query editor interface. On the left, a tree view displays the database structure, including tables like 'customer', 'detail', 'orders', and 'supplier'. The main window shows a SQL query in the 'Query Editor' tab:

```
1 SELECT customer, supplier, SUM(price) from orders GROUP BY customer, supplier
```

Below the query editor, the 'Result1' tab displays the query results in a table format:

customer	supplier	sum
2	5	10800000
4	3	2700
5	4	63900
1	4	21960000
2	4	36000
3	6	22500
5	1	13500
3	3	5400
1	3	2160000
2	1	900000
2	2	4500000
4	5	162000
4	4	720000
1	6	124200

15b

The screenshot shows the Navicat Cloud interface. On the left, the 'My Connections' tree is expanded to '192.168.127.133 DOCKER', then 'hse', 'public', and 'Tables'. The 'customer' table is selected. The main area shows the 'Query Editor' with the following SQL query:

```
1
2 SELECT * from (
3 SELECT detail, count(*) bought_times from orders WHERE supplier IN (SELECT id from supplier WHERE address = 'Советский' ) GROUP BY detail) P
4 WHERE P.bought_times > 3
```

Below the query editor, the 'Result1' tab is active, showing a table with two columns: 'detail' and 'bought_times'. The first row shows '(Null)' for 'detail' and '(Null)' for 'bought_times'.

15c

The screenshot shows the Navicat Cloud interface. On the left, the 'My Connections' tree is expanded to '192.168.127.133 DOCKER', then 'hse', 'public', and 'Tables'. The 'customer' table is selected. The main area shows the 'Query Editor' with the following SQL query:

```
1 SELECT date from orders GROUP BY date HAVING MIN(price) > 10000;
```

Below the query editor, the 'Result1' tab is active, showing a table with one column: 'date'. The first row shows 'Январь' (January).

15d

Navicat Cloud

My Connections

192.168.127.133 DOCKER

hse

public

Tables

customer

detail

orders

supplier

Views

Functions

Queries

10-a

10d(7a)

11-b

11-c

11d

12

13-a

13-b

13-c

13-d

14-a

14-b

14-c

15-b

15-c

15-d

7-c

7-d

8

n

Query Editor

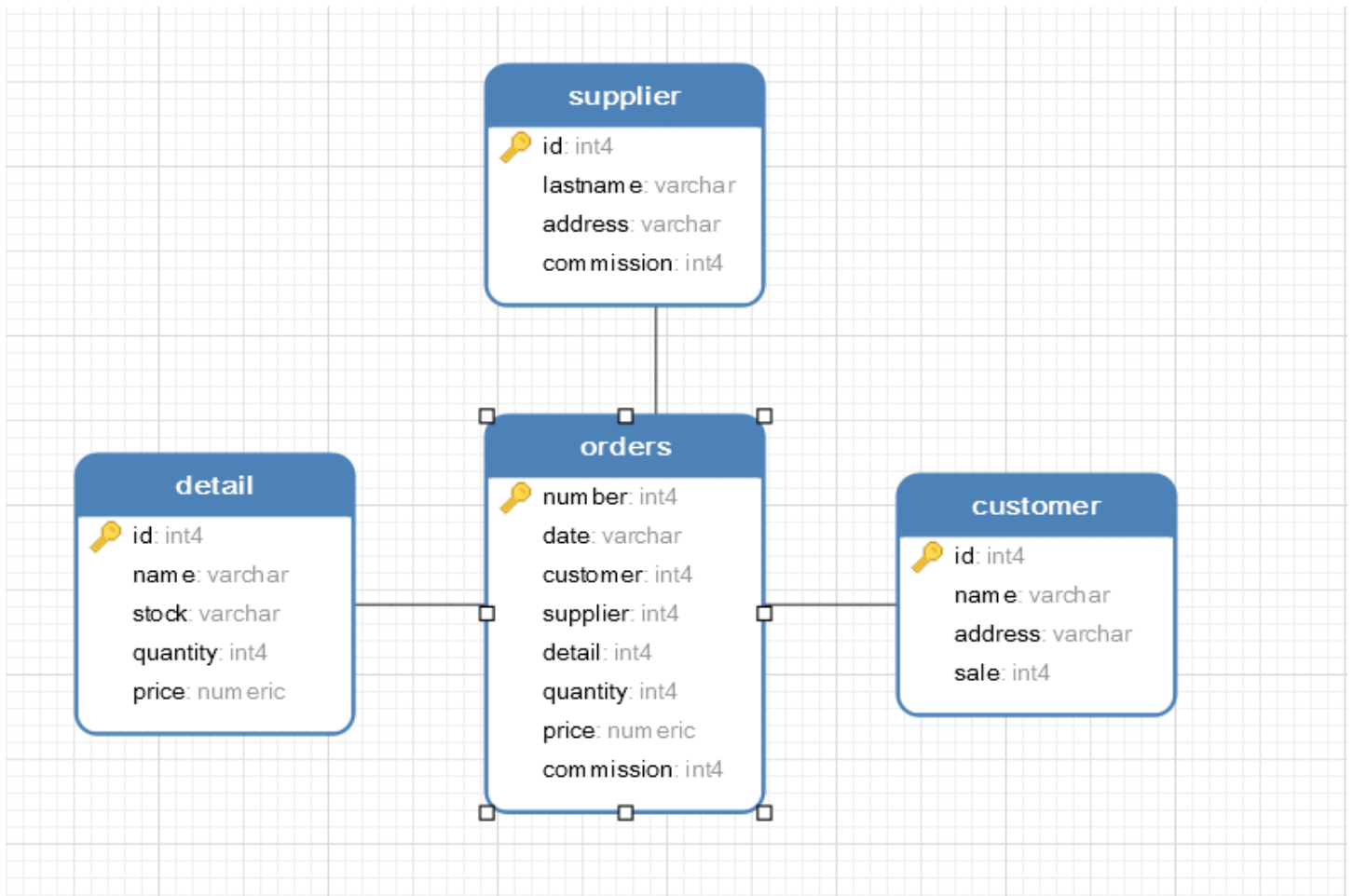
```
1 SELECT detail, AVG(quantity) from orders
2 WHERE detail in (select id from detail WHERE price > 10000)
3 GROUP BY detail
```

Message Result1

detail	avg
(Null)	(Null)

Примечание: но вообще деталей дороже 10000

ERM



Примечания

SQL запросы в формате .sql, изображения для данного .doc документа доступны по адресу:

<https://github.com/NBakaev/hse-all/tree/master/third-course-db-laba-1>

Структура и данные для базы данных

6)

ПОТРЕБИТЕЛЬ

|| ИДЕНТИФИКАТОР | НАЗВАНИЕ | АДРЕС ЖИТЕЛЬСТВА | СКИДКА, % ||
=====

001	АО ВАРЯ	Сормовский	10
002	ГАЗ	Автозаводский	7
003	МП ВЕРА	Канавинский	5
004	МП	Канавинский	3
005	АО СТАЛЬ	Советский	0

ПОСТАВЩИК

|| ИДЕНТИФИКАТОР | ФАМИЛИЯ | АДРЕС | КОММИСИОННЫЕ, % ||
=====

001	Артюхина	Сормовский	4
002	Щепин	Приокский	4
003	Власов	Канавинский	5
004	Кузнецова	Советский	5
005	Цепилева	Нижегородский	3
006	Корнилов	Нижегородский	6

ЗАКАЗ

НОМЕР | ДАТА | ПОТРЕБИТЕЛЬ | ПОСТАВЩИК | ДЕТАЛЬ | КОЛ-ВО | СУММА, РУБ
=====

00001	Январь	005	004	003	7	21000
00002	Февраль	003	003	003	2	6000
00003	Февраль	004	005	004	200	180000
00004	Март	005	004	002	50	50000
00005	Апрель	001	006	007	110	132000
00006	Апрель	004	004	001	150	750000
00007	Май	002	004	006	20	40000
00008	Июнь	001	003	007	2000	2400000
00009	Июнь	002	005	007	10000	12000000
00010	Июнь	003	006	001	5	25000
00011	Июнь	004	003	003	1	3000
00012	Июнь	004	004	001	10	50000
00013	Июль	001	006	006	3	6000
00014	Июль	002	001	002	1000	1000000
00015	Июль	002	002	001	100	5000000
00016	Июль	005	001	005	100	15000
00017	Август	001	004	007	12000	24400000

ДЕТАЛЬ

|| ИДЕНТИФИКАТОР | НАИМЕНОВАНИЕ | СКЛАД | КОЛ-ВО | ЦЕНА, РУБ ||
=====

001	Втулка	Сормовский	20000	5000
002	Болт	Сормовский	40000	1000
003	Ключ гаечный	Канавинский	5000	3000
004	Шпилька	Автозаводский	10000	900
005	Винт	Сормовский	50000	1500
006	Молоток	Канавинский	1200	2000
007	Шуруп	Сормовский	30000	1200