

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ
ФЕДЕРАЦИИ

Федеральное государственное автономное образовательное учреждение
высшего образования «Южно-Уральский государственный университет
(национальный исследовательский университет)»
Высшая школа электроники и компьютерных наук
Кафедра системного программирования

Отчёт по практической работе №6

Тема: «Разработка запросов»

Выполнил:

студент группы КЭ-319

_____ Е. А. Паршикова

_____ 2022 г.

Проверил:

преподаватель кафедры СП

_____ П. Д. Ческидов

_____ 2022 г.

Челябинск

2022

Оглавление

Постановка задачи	3
1 Разработка модели предметной области	4
2 Разработка схемы базы данных и ограничения целостности данных	5
3 Разработка тестовой базы данных	9
4 Разработка представлений.....	14
5 Разработка запросов	18

Постановка задачи

1. Разработать формальную модель предметной области для небольшой строительной фирмы, занимающейся ремонтом помещений. При выполнении проектов в фирме используются детали, закупаемые у поставщиков;
2. Разработать схему базы данных в виде ER-диаграммы. Разработать команды для создания таблиц базы данных в PostgreSQL;
3. Разработать ограничения целостности атрибутов и ограничения ссылочной целостности данных;
4. Разработать тестовую базу данных так, чтобы в каждой таблице содержалось не меньше 20 записей. Заполнить таблицы;
5. Разработать представления для пользователей базы данных: Экономист и Директор.

1 Разработка модели предметной области

Были использованы следующие сущности с атрибутами:

- Сущность деталь с атрибутами: уникальный идентификатор, название, цена (в рублях), цвет, вес (в граммах). Деталь с ценой менее 1000 рублей считается дешевой, иначе – дорогой;
- Сущность поставщик с атрибутами: уникальный идентификатор, название, город, адрес, рейтинг. Рейтинг представляет собой целое число из диапазона 1..10, которое показывает надёжность поставщика. Поставщик, имеющий рейтинг менее 6 считается ненадёжным, иначе – надёжным;
- Сущность поставка детали для проекта, имеющая атрибут «количество деталей»;
- Сущность проект с атрибутами: уникальный идентификатор, название, город, адрес, бюджет (в рублях).

Между сущностями существуют следующие связи:

- Деталь может поставляться различными поставщиками для различных проектов;
- Поставщик может поставлять различные детали для различных проектов;
- В проекте могут использоваться различные детали, поставленные различными поставщиками.

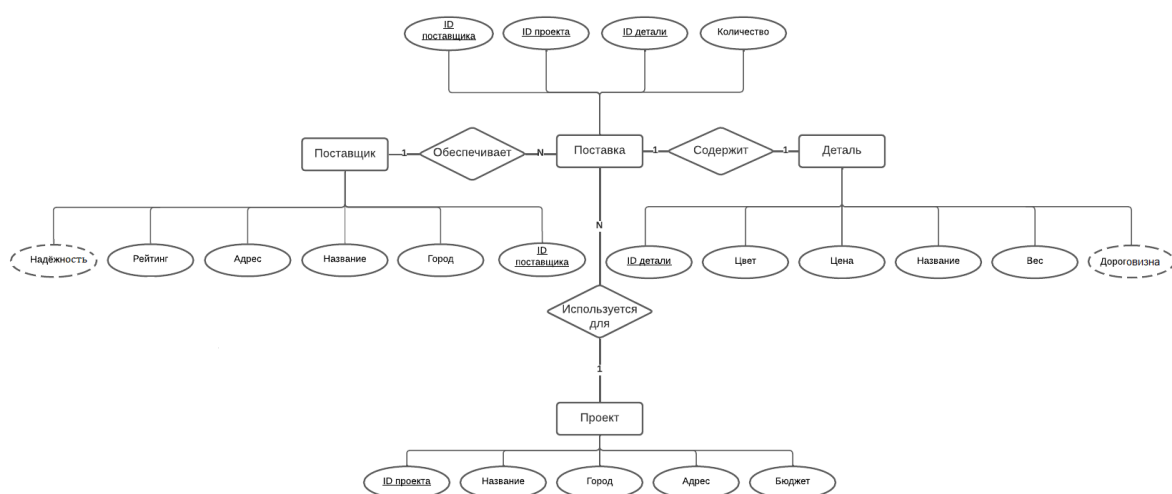


Рис. 1. Разработанная формальная модель предметной области в нотации П. Чена

2 Разработка схемы базы данных и ограничения целостности данных

- Разработана схема базы данных в виде ER-диаграммы;
- Создана учётная запись пользователя в PostgreSQL;
- Разработаны команды SQL для создания реляционных таблиц базы данных;
- Разработаны ограничения целостности атрибутов и ограничения ссылочной

целостности данных:

- Название поставщика уникально в рамках города;
- Название, город и адрес поставщика не могут быть пустыми;
- По умолчанию адрес поставщика «неизвестен»;
- Рейтинг поставщика находится в диапазоне 1..10
- Название и цена детали не могут быть пустыми;
- Вес и цена детали положительны;
- Цвет детали принимает значение из фиксированного списка (белый, серый, чёрный, коричневый, синий, зелёный, красный);
- Название, город и адрес проекта не могут быть пустыми;
- Бюджет проекта положительный;
- Количество деталей в поставке положительно;
- Удаление (изменение) детали, поставщика или проекта инициирует каскадное удаление (изменение) соответствующих поставок.

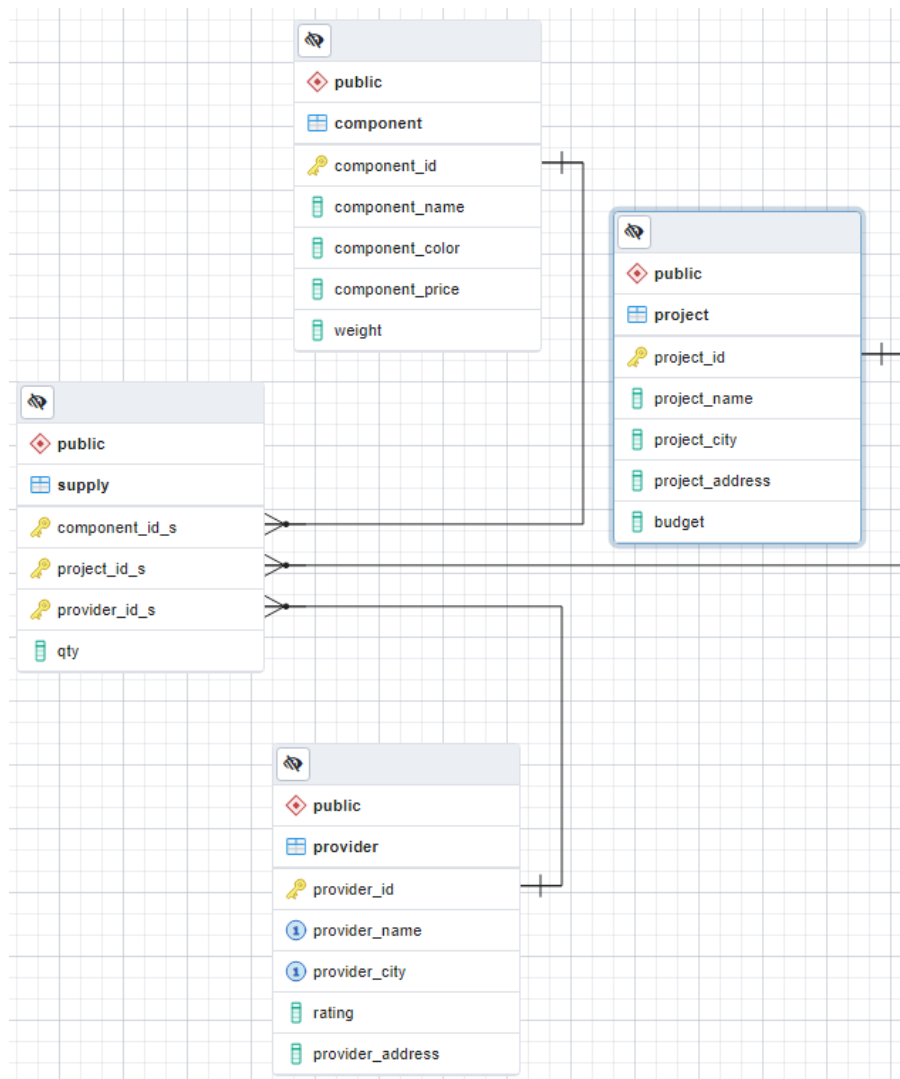


Рис. 2. Схема базы данных в виде ER-диаграммы

```

CREATE TABLE IF NOT EXISTS public.component
(
    component_id character(4) COLLATE pg_catalog."default" NOT NULL,
    component_name character(40) COLLATE pg_catalog."default" NOT NULL,
    component_color character(20) COLLATE pg_catalog."default" NOT NULL,
    component_price integer NOT NULL,
    weight double precision NOT NULL,
    CONSTRAINT component_pkey PRIMARY KEY (component_id),
    CONSTRAINT weight_check CHECK (weight > 0::double precision),
    CONSTRAINT component_price_check CHECK (component_price > 0),
    CONSTRAINT component_color_check CHECK (component_color = ANY (ARRAY['white'::bpchar,
                                                                    'grey'::bpchar,
                                                                    'black'::bpchar,
                                                                    'brown'::bpchar,
                                                                    'green'::bpchar,
                                                                    'blue'::bpchar,
                                                                    'red'::bpchar]))
)
  
```

Рис. 3. Листинг создания таблицы «Деталь» с ограничениями

```

CREATE TABLE IF NOT EXISTS public.project
(
    project_id character(4) COLLATE pg_catalog."default" NOT NULL,
    project_name character(40) COLLATE pg_catalog."default" NOT NULL,
    project_city character(20) COLLATE pg_catalog."default" NOT NULL,
    project_address character(100) COLLATE pg_catalog."default" NOT NULL,
    budget double precision NOT NULL,
    CONSTRAINT project_pkey PRIMARY KEY (project_id),
    CONSTRAINT budget_check CHECK (budget > 0::double precision)
)

```

Рис. 4. Листинг создания таблицы «Проект» с ограничениями

```

CREATE TABLE IF NOT EXISTS public.provider
(
    provider_id character(4) COLLATE pg_catalog."default" NOT NULL,
    provider_name character(40) COLLATE pg_catalog."default" NOT NULL,
    provider_city character(20) COLLATE pg_catalog."default" NOT NULL,
    rating integer NOT NULL,
    provider_address character(25) COLLATE pg_catalog."default" NOT NULL DEFAULT 'unknown'::bpchar,
    CONSTRAINT provider_pkey PRIMARY KEY (provider_id),
    CONSTRAINT uni_city_name UNIQUE (provider_name, provider_city),
    CONSTRAINT rating_check CHECK (rating >= 1 AND rating <= 10)
)

```

Рис. 5. Листинг создания таблицы «Поставщик» с ограничениями

```

CREATE TABLE IF NOT EXISTS public.supply
(
    component_id_s character(4) COLLATE pg_catalog."default" NOT NULL,
    project_id_s character(4) COLLATE pg_catalog."default" NOT NULL,
    provider_id_s character(4) COLLATE pg_catalog."default" NOT NULL,
    qty integer NOT NULL,
    CONSTRAINT supply_pkey PRIMARY KEY (provider_id_s, component_id_s, project_id_s),
    CONSTRAINT supply_component_id_fkey FOREIGN KEY (component_id_s)
        REFERENCES public.component (component_id) MATCH SIMPLE
        ON UPDATE CASCADE
        ON DELETE CASCADE,
    CONSTRAINT supply_project_id_fkey FOREIGN KEY (project_id_s)
        REFERENCES public.project (project_id) MATCH SIMPLE
        ON UPDATE CASCADE
        ON DELETE CASCADE,
    CONSTRAINT supply_provider_id_fkey FOREIGN KEY (provider_id_s)
        REFERENCES public.provider (provider_id) MATCH SIMPLE
        ON UPDATE CASCADE
        ON DELETE CASCADE,
    CONSTRAINT supply_qty_check CHECK (qty > 0)
)

```

Рис. 6. Листинг создания таблицы «Поставка» с ограничениями

```

CREATE OR REPLACE FUNCTION public.isexpensive(
    x integer)
    RETURNS character varying
    LANGUAGE 'sql'
    COST 100
    VOLATILE PARALLEL UNSAFE
AS $BODY$
SELECT CASE WHEN x < 1000 THEN 'Cheap' ELSE 'Expensive' END
$BODY$;

```

Рис. 7. Листинг создания функции для определения дороговизны детали

```

CREATE OR REPLACE FUNCTION public.isreliable(
    x integer)
    RETURNS character varying
    LANGUAGE 'sql'
    COST 100
    VOLATILE PARALLEL UNSAFE
AS $BODY$
SELECT CASE WHEN x < 6 THEN 'Unreliable' ELSE 'Reliable' END
$BODY$;

```

Рис. 8. Листинг создания функции для определения надёжности поставщика

3 Разработка тестовой базы данных

- Заполнена часть ранее созданных таблиц через команду SQL:
insert into ... values ...
- Остальные таблицы заполнены из подготовленных файлов с расширением .csv

Листинг разработки тестовой базы данных

/*Пример заполнения через Query Tool*/

```
INSERT INTO component
```

```
VALUES
```

```
(1, 'Board', 'brown', 268, 6.5),  
(2, 'Nail', 'grey', 2, 0.20),  
(3, 'Sheet metal', 'grey', 2054, 2.916),  
(4, 'PVC panel', 'white', 275, 1.35);
```

/*Пример заполнения импортом*/

```
COPY something_data FROM '*.csv' DELIMITER ';' CSV HEADER;
```

Сущности, заполненные данными

	component_id [PK] character (4)	component_name character (40)	component_color character (20)	component_price integer	weight double precision
1	1	Board	brown	268	6500
2	2	Nail	grey	2	200
3	3	Sheet metal	grey	2054	2916
4	4	PVC panel	white	275	1350
5	5	self-tapping s...	black	1	1
6	6	screw	grey	2	1.5
7	7	dowel	grey	5	4
8	8	furniture faste...	brown	80	250
9	9	rigging	white	150	450
10	10	metric fastener	grey	167	700
11	11	warm floor	brown	70000	28000
12	12	chimney	grey	5610	4200
13	13	tank	grey	12140	57000
14	14	water heater	red	14000	10300
15	15	Radiator	white	15000	44500
16	16	mixer	red	2568	1570
17	17	bath	white	9460	18000
18	18	bath screen	blue	849	5600
19	19	installation	green	478	3300
20	20	siphon	grey	154	780

Рис. 9. Таблица «Деталь»

21	21	corrugation	white	146	500
22	22	toilet	black	14000	11000
23	23	sink	red	13499	8400
24	24	bidet	white	7300	7500
25	25	heated towel r...	grey	9630	4880
26	26	linoleum	brown	750	4800
27	27	parquet board	brown	1079	710
28	28	artificial grass	green	438	768
29	29	pvc tile	blue	178	846
30	30	dirt protection	black	741	1780
31	31	glue packaging	green	893	1500
32	32	plinth	brown	49	66
33	33	corner	brown	12	41
34	34	laminare	brown	749	2320
35	35	lamp	white	72	490
36	36	extension	white	580	650
37	37	insulating tape	blue	99	320
38	38	battery	white	15	44
39	39	cable	white	450	2200
40	40	the wire	black	125	450

Рис. 10. Продолжение таблицы «Деталь»

	project_id [PK] character (4)	project_name character (40)	project_city character (20)	project_address character (100)	budget double precision
1	1	Summer kitch...	Moscow	Highway Ladyg...	4500000
2	2	Winter garden	Sochi	Lomonosov em...	1200000
3	3	Bedroom 1	Novosibirsk	Stalin Street 76...	500000
4	4	Apartment 84	Moscow	Balkanskaya st...	10870000
5	5	aquarius	Moscow	Boulevard Buda...	8521000
6	6	eggplant	Sochi	Domodedovska...	7365000
7	7	physical	Novosibirsk	Bucharestskay...	4500000
8	8	mouser	Moscow	Gogol Avenue 9...	6320000
9	9	calculation	Sochi	Chapaeva stree...	7410000
10	10	embellishment	Yekaterinburg	Jubilee street 2...	9645000
11	11	wrench	Moscow	Parkova street ...	2360000
12	12	wisdom	Moscow	Berezovaya str...	2780000
13	13	theory	Yekaterinburg	Kalinina street ...	9632000
14	14	association	Moscow	Zarechny lane ...	4521000
15	15	inspection	Novosibirsk	New lane 2/39	6021000
16	16	shed	Moscow	School Street 2...	2451000
17	17	thistle	Sochi	Krasnoarmeysk...	2700000
18	18	hold	Novosibirsk	Lake street 10/...	2310420
19	19	ceramic	Moscow	Victory Street 2...	2140640
20	20	carrot	Yekaterinburg	Pervomayskay...	1523000

Рис. 11. Таблица «Проект»

	provider_id [PK] character (4)	provider_name character (40)	provider_city character (20)	rating integer	provider_address character (25)
1	1	LLC Trading h...	Moscow	5	Cosmonauts stre...
2	2	LLC "Stroy-24"	Sochi	7	lane Gagarina 47...
3	3	LLC TK "Uralk...	Yekaterinburg	4	street Druzhnaya ...
4	4	LLC "DonKapit...	Novosibirsk	10	Flower Street 3/2...
5	5	LLC "BVB-Allia...	Moscow	7	Polevoy lane 14/6
6	6	LLC "Lipetskgi...	Yekaterinburg	9	Sulskaya street 2...
7	7	IP Sundeev	Novosibirsk	3	Solnechny Lane ...
8	8	LLC "MCS"	Sochi	8	Leninskaya stree...
9	9	LLC "Elariz"	Moscow	9	Sovkhoznaya Str...
10	10	OOO "TERMO...	Moscow	2	Molodyozhnaya s...
11	11	RBI	Moscow	7	Factory street 23...
12	12	Setl City	Yekaterinburg	6	Central lane 16/2
13	13	GK KVS	Novosibirsk	8	Kirova street 17/...
14	14	Akvilon	Sochi	9	Maxim Gorky Str...
15	15	Normann	Moscow	7	Field Lane 5/7
16	16	Glorax	Yekaterinburg	10	Dzerzhinsky Stre...
17	17	IPS	Moscow	9	Minsk street 3/1
18	18	Navis Group	Novosibirsk	7	Northern street 2...
19	19	AAG	Novosibirsk	6	Communist Stree...
20	20	RANT	Sochi	10	Zaslonoa Street ...

Рис. 12. Таблица «Поставщик»

	component_id_s [PK] character (4)	project_id_s [PK] character (4)	provider_id_s [PK] character (4)	qty integer
1	1	2	2	100
2	2	3	4	200
3	3	4	1	4
4	4	1	1	48
5	1	1	5	1
6	2	2	20	2
7	3	3	18	3
8	4	4	17	4
9	5	5	1	5
10	5	6	2	4
11	6	7	4	3
12	6	8	5	2
13	7	9	8	1
14	7	10	16	2
15	8	11	17	3
16	8	12	11	4
17	9	13	6	5
18	9	14	5	6
19	10	15	18	7
20	10	16	17	6

Рис. 13. Таблица «Поставка»

21	11	17	20	5
22	11	18	13	4
23	12	19	17	3
24	12	20	12	2
25	13	1	9	1
26	13	2	14	2
27	14	3	19	3
28	14	4	15	4
29	15	5	5	5
30	15	6	8	6
31	16	7	7	5
32	16	8	17	4
33	17	9	20	3
34	17	10	12	2
35	18	11	15	1
36	18	12	9	2
37	19	13	16	3
38	19	14	10	4
39	20	15	18	5
40	20	16	15	6

Рис. 14. Продолжение таблицы «Поставка»

41	21	17	14	7
42	21	18	4	8
43	22	19	17	9
44	22	20	6	8
45	23	1	10	7
46	23	2	8	6
47	24	3	13	5
48	24	4	11	4
49	25	5	15	3
50	25	6	14	2
51	26	7	13	1
52	26	8	15	2
53	27	9	14	3
54	27	10	6	4
55	28	11	17	5
56	28	12	10	4
57	29	13	6	3
58	29	14	5	2
59	30	15	18	1
60	30	16	17	2

Рис. 15. Продолжение таблицы «Поставка»

61	31	17	20	3
62	31	18	13	4
63	32	19	17	5
64	32	20	12	4
65	33	1	11	3
66	33	2	20	2
67	34	3	7	1
68	34	4	5	2
69	35	5	11	3
70	35	6	20	4
71	36	7	18	5
72	36	8	10	6
73	37	9	2	7
74	37	10	3	8
75	38	11	15	9
76	38	12	11	10
77	39	13	16	9
78	39	14	10	8
79	40	15	19	7
80	40	16	17	6

Рис. 16. Продолжение таблицы «Поставка»

4 Разработка представлений

Разработаны представления данных со следующими условиями:

Для роли Экономиста

Экономист имеет право осуществлять выборку, добавление, обновление и удаление данных о поставщиках и поставках, а также выборку данных о деталях и проектах.

При выборке данных Поставщик для Экономиста должен быть представлен следующими атрибутами: название, город, адрес, рейтинг, признак «ненадежный» или «надежный» (символьная строка). Данные должны быть отсортированы следующим образом: по городу, названию, убыванию рейтинга.

Деталь для Экономиста должна быть представлена следующими атрибутами: название, цена в *тыс.руб.*, цвет, вес в *килограммах*, признак «дешевая» или «дорогая» (символьная строка). Данные должны быть отсортированы следующим образом: по убыванию цены, названию, цвету, возрастанию веса.

При выборке данных Проект для Экономиста должен быть представлен следующими атрибутами: название, город, адрес, бюджет. Данные должны быть отсортированы следующим образом: по городу, названию, убыванию бюджета.

При выборке данных Поставка для Экономиста должна быть представлена следующими атрибутами: название детали, цвет детали, признак детали «дешевая» или «дорогая» (символьная строка), название поставщика, город поставщика, признак поставщика «ненадежный» или «надежный» (символьная строка), количество деталей, вес поставки в *килограммах*, стоимость поставки в *тыс.руб.* Данные должны быть отсортированы следующим образом: названию детали, названию поставщика, по убыванию стоимости, по убыванию веса.

Для роли Директора

Директор имеет право осуществлять выборку, добавление, обновление и удаление данных о поставщиках, деталях и проектах, а также выборку и удаление данных о поставках.

При выборке данных Поставщик для Директора должен быть представлен следующими атрибутами: название, город, признак «ненадежный» или

«надежный» (символьная строка). Данные должны быть отсортированы следующим образом: по признаку, городу, названию.

При выборке данных Деталь для Директора должна быть представлена следующими атрибутами: название, цена *в тыс.руб.*, цвет, признак «дешевая» или «дорогая» (символьная строка). Данные должны быть отсортированы следующим образом: по убыванию цены, названию.

При выборке данных Проект для Директора должен быть представлен следующими атрибутами: название, город, бюджет. Данные должны быть отсортированы следующим образом: по убыванию бюджета, городу, названию.

При выборке данных Поставка для Директора должна быть представлена следующими атрибутами: название детали, цвет детали, признак детали «дешевая» или «дорогая» (символьная строка), название поставщика, город поставщика, признак поставщика «ненадежный» или «надежный» (символьная строка), количество деталей, вес поставки *в килограммах*, стоимость поставки *в тыс.руб.* Данные должны быть отсортированы следующим образом: по признаку детали, городу детали, признаку поставщика, городу поставщика, названию детали, названию поставщика, по убыванию стоимости, по убыванию веса.

```
CREATE OR REPLACE VIEW public.director_component_view
AS
SELECT component.component_name,
       component.component_price::real / 1000::double precision AS component_price_thousands,
       component.component_color,
       isexpensive(component.component_price) AS is_expensive
FROM component
ORDER BY component.component_price DESC, component.component_name;
```

Рис. 17. Листинг создания представления деталей для директора

```
CREATE OR REPLACE VIEW public.director_project_view
AS
SELECT project.project_name,
       project.project_city,
       project.budget
FROM project
ORDER BY project.budget DESC, project.project_city, project.project_name;
```

Рис. 18. Листинг создания представления проектов для директора

```

CREATE OR REPLACE VIEW public.director_provider_view
AS
SELECT provider.provider_name,
       provider.provider_city,
       isreliable(provider.rating) AS is_reliable
FROM provider
ORDER BY (isreliable(provider.rating)), provider.provider_city, provider.provider_name;

```

Рис. 19. Листинг создания представления поставщиков для директора

```

CREATE OR REPLACE VIEW public.director_supply_view
AS
SELECT component.component_name,
       component.component_color,
       isexpensive(component.component_price) AS is_expensive,
       provider.provider_name,
       provider.provider_city,
       isreliable(provider.rating) AS is_reliable,
       supply.qty,
       round(component.weight::numeric / 1000::numeric * supply.qty::numeric, 3)
       AS weight_supply_kg,
       round(component.component_price::numeric / 1000::numeric * supply.qty::numeric, 3)
       AS component_price_thousands
FROM supply,
     component,
     provider
WHERE supply.component_id_s = component.component_id AND
      supply.provider_id_s = provider.provider_id
ORDER BY (isexpensive(component.component_price)),
       (isreliable(provider.rating)),
       provider.provider_city,
       component.component_name, provider.provider_name,
       (component.component_price::real / 1000::double precision * supply.qty::double precision) DESC,
       (component.weight::real / 1000::double precision * supply.qty::double precision) DESC;

```

Рис. 20. Листинг создания представления поставок для директора

```

CREATE OR REPLACE VIEW public.economist_component_view
AS
SELECT component.component_name,
       component.component_price::real / 1000::double precision AS component_price_thousands,
       component.component_color,
       component.weight::real / 1000::double precision AS weight_kg,
       isexpensive(component.component_price) AS is_expensive
FROM component
ORDER BY component.component_price DESC, component.component_name, component.component_color, component.weight;

```

Рис. 21. Листинг создания представления деталей для экономиста

```

CREATE OR REPLACE VIEW public.economist_project_view
AS
SELECT project.project_name,
       project.project_city,
       project.project_address,
       project.budget
FROM project
ORDER BY project.project_city, project.project_name, project.budget DESC;

```

Рис. 22. Листинг создания представления проектов для экономиста


```

CREATE OR REPLACE VIEW public.economist_provider_view
AS
SELECT provider.provider_name,
       provider.provider_city,
       provider.provider_address,
       provider.rating,
       isreliable(provider.rating) AS is_reliable
FROM provider
ORDER BY provider.provider_city, (isreliable(provider.rating)) DESC, provider.provider_name;

```

Рис. 23. Листинг создания представления поставщиков для экономиста

```

CREATE OR REPLACE VIEW public.economist_supply_view
AS
SELECT component.component_name,
       component.component_color,
       isexpensive(component.component_price) AS is_expensive,
       provider.provider_name,
       provider.provider_city,
       isreliable(provider.rating) AS is_reliable,
       supply.qty,
       round(component.weight::numeric / 1000::numeric * supply.qty::numeric, 3) AS weight_supply_kg,
       round(component.component_price::numeric / 1000::numeric * supply.qty::numeric, 3) AS component_price_thousands
FROM supply,
     component,
     provider
WHERE supply.component_id_s = component.component_id AND supply.provider_id_s = provider.provider_id
ORDER BY component.component_name, provider.provider_name, component.component_price DESC, component.weight DESC;

```

Рис. 24. Листинг создания представления поставок для экономиста

	component_name character (40)	component_color character (20)	is_expensive character varying	provider_name character (40)	provider_city character (20)	is_reliable character varying	qty integer	weight_supply_kg numeric	component_price_thousands numeric
1	artificial grass	green	Cheap	IPS	Moscow	Reliable	5	3.840	2.190
2	bath screen	blue	Cheap	LLC "Elariz"	Moscow	Reliable	2	11.200	1.698
3	bath screen	blue	Cheap	Normann	Moscow	Reliable	1	5.600	0.849
4	battery	white	Cheap	Normann	Moscow	Reliable	9	0.396	0.135
5	battery	white	Cheap	RBI	Moscow	Reliable	10	0.440	0.150
6	Board	brown	Cheap	LLC "BVB-Allia...	Moscow	Reliable	1	6.500	0.268
7	corner	brown	Cheap	RBI	Moscow	Reliable	3	0.123	0.036
8	dirt protection	black	Cheap	IPS	Moscow	Reliable	2	3.560	1.482
9	furniture faste...	brown	Cheap	IPS	Moscow	Reliable	3	0.750	0.240
10	furniture faste...	brown	Cheap	RBI	Moscow	Reliable	4	1.000	0.320
11	laminate	brown	Cheap	LLC "BVB-Allia...	Moscow	Reliable	2	4.640	1.498
12	lamp	white	Cheap	RBI	Moscow	Reliable	3	1.470	0.216
13	linoleum	brown	Cheap	Normann	Moscow	Reliable	2	9.600	1.500
14	metric fastener	grey	Cheap	IPS	Moscow	Reliable	6	4.200	1.002
15	plinth	brown	Cheap	IPS	Moscow	Reliable	5	0.330	0.245
16	PVC panel	white	Cheap	IPS	Moscow	Reliable	4	5.400	1.100
17	pvc tile	blue	Cheap	LLC "BVB-Allia...	Moscow	Reliable	2	1.692	0.356
18	rigging	white	Cheap	LLC "BVB-Allia...	Moscow	Reliable	6	2.700	0.900
19	screw	grey	Cheap	LLC "BVB-Allia...	Moscow	Reliable	2	0.003	0.004
20	siphon	grey	Cheap	Normann	Moscow	Reliable	6	4.680	0.924

Рис. 25. Таблица представления поставок для экономиста

5 Разработка запросов

Разработаны запросы к базе данных в форме функций. Для удобства созданы дополнительные представления, содержащие необходимую для отчётов информацию.

Для экономиста

1. Список всех, надежных, ненадежных поставщиков.
2. Список всех, дорогих, дешевых деталей.
3. Список всех проектов, проектов из заданного города, проектов с бюджетом в заданном диапазоне.
4. Список всех поставок, поставок дорогих и дешевых деталей, поставок от надежных и ненадежных поставщиков, поставок со стоимостью из заданного диапазона.

Для директора

1. Список всех поставщиков, список поставщиков с рейтингом не ниже/не выше среднего, список поставщиков с максимальным/минимальным рейтингом – с указанием у каждого поставщика общей суммы, сделанной им поставок (непрерывное условие для всех списков).
2. Список всех деталей, список деталей с ценой не ниже/не выше средней, список деталей максимальной/минимальной ценой – с указанием у каждой детали общей суммы ее поставок (непрерывное условие для всех списков).
3. Список всех проектов, список проектов с бюджетом не ниже/не выше среднего, список проектов с максимальным/минимальным бюджетом – с указанием общей суммы поставок по данному проекту (непрерывное условие для всех списков).
4. Сумма всех поставок, сумма поставок дорогих деталей, сумма поставок дешевых деталей, сумма поставок от надежных поставщиков, сумма поставок от ненадежных поставщиков, сумма поставок от заданного поставщика.

```

CREATE OR REPLACE VIEW public.director_component_view2
AS
SELECT component.component_id,
       component.component_name,
       component.component_price::real / 1000::double precision AS component_price_thousands,
       component.component_color,
       isexpensive(component.component_price) AS is_expensive,
       sum(round(component.component_price::numeric / 1000::numeric * supply.qty::numeric, 3)) AS total
FROM component,
     supply
WHERE component.component_id = supply.component_id_s
GROUP BY component.component_id
ORDER BY component.component_price DESC, component.component_name;

```

Рис. 26. Создание дополнительного представления деталей для директора

```

CREATE OR REPLACE VIEW public.director_project_view2
AS
SELECT project.project_id,
       project.project_name,
       project.project_city,
       project.budget,
       sum(round(component.component_price::numeric / 1000::numeric * supply.qty::numeric, 3)) AS total
FROM project,
     component,
     supply
WHERE component.component_id = supply.component_id_s AND project.project_id = supply.project_id_s
GROUP BY project.project_id
ORDER BY project.budget DESC, project.project_city, project.project_name;

```

Рис. 27. Создание дополнительного представления проектов для директора

```

CREATE OR REPLACE VIEW public.director_provider_view2
AS
SELECT provider.provider_id,
       provider.provider_name,
       provider.provider_city,
       provider.rating,
       isreliable(provider.rating) AS is_reliable,
       sum(round(component.component_price::numeric / 1000::numeric * supply.qty::numeric, 3)) AS total
FROM provider,
     supply,
     component
WHERE component.component_id = supply.component_id_s AND provider.provider_id = supply.provider_id_s
GROUP BY provider.provider_id
ORDER BY (isreliable(provider.rating)), provider.provider_city, provider.provider_name;

```

Рис. 28. Создание дополнительного представления поставщиков для директора

```

CREATE OR REPLACE FUNCTION public.report_economist_provider(
)
RETURNS TABLE(provider_name character, provider_city character,
                provider_address character, rating integer, is_reliable character varying)
LANGUAGE 'plpgsql'
COST 100
VOLATILE PARALLEL UNSAFE
ROWS 1000

AS $BODY$
BEGIN
RETURN QUERY (
    SELECT * FROM economist_provider_view
);
END;
$BODY$;

```

Рис. 29. Функция запроса всех поставщиков для экономиста

```

CREATE OR REPLACE FUNCTION public.report_economist_reliable_provider(
reliable character varying)
RETURNS TABLE(provider_name character, provider_city character,
                provider_address character, rating integer,
                is_reliable character varying)
LANGUAGE 'plpgsql'
COST 100
VOLATILE PARALLEL UNSAFE
ROWS 1000

AS $BODY$
BEGIN
RETURN QUERY (
    SELECT economist_provider_view.provider_name, economist_provider_view.provider_city,
    economist_provider_view.provider_address, economist_provider_view.rating,
    economist_provider_view.is_reliable
    FROM economist_provider_view
    WHERE economist_provider_view.is_reliable = reliable
);
END;
$BODY$;

```

Рис. 30. Функция запроса (не)надёжных поставщиков для экономиста

```

CREATE OR REPLACE FUNCTION public.report_economist_component(
)
RETURNS TABLE(component_name character, component_price double precision,
               component_color character, weight_kg double precision,
               is_expensive character varying)
LANGUAGE 'plpgsql'
COST 100
VOLATILE PARALLEL UNSAFE
ROWS 1000

AS $BODY$
BEGIN
RETURN QUERY (
    SELECT * FROM economist_component_view
);
END;
$BODY$;

```

Рис. 31. Функция запроса всех деталей для экономиста

```

CREATE OR REPLACE FUNCTION public.report_economist_expensive_component(expensive character varying
)
RETURNS TABLE(component_name character, component_price double precision,
               component_color character, weight_kg double precision,
               is_expensive character varying)
LANGUAGE 'plpgsql'
COST 100
VOLATILE PARALLEL UNSAFE
ROWS 1000

AS $BODY$
BEGIN
RETURN QUERY (
    SELECT economist_component_view.component_name,
           economist_component_view.component_price_thousands,
           economist_component_view.component_color,
           economist_component_view.weight_kg,
           economist_component_view.is_expensive
    FROM economist_component_view
    WHERE economist_component_view.is_expensive = expensive);
END;
$BODY$;

```

Рис. 32. Функция запроса для (не)дорогих деталей для экономиста

```

CREATE OR REPLACE FUNCTION public.report_economist_project(
)
RETURNS TABLE(project_name character, project_place character, project_address character,
               budget double precision)
LANGUAGE 'plpgsql'
COST 100
VOLATILE PARALLEL UNSAFE
ROWS 1000

AS $BODY$
BEGIN
RETURN QUERY (
    SELECT * FROM economist_project_view
);
END;
$BODY$;

```

Рис. 33. Функция запроса всех проектов для экономиста

```

CREATE OR REPLACE FUNCTION public.report_economist_project_city(
    town character)
    RETURNS TABLE(project_name character, project_place character,
        project_address character, budget double precision)
    LANGUAGE 'plpgsql'
    COST 100
    VOLATILE PARALLEL UNSAFE
    ROWS 1000

AS $BODY$
BEGIN
RETURN QUERY (
    SELECT * FROM economist_project_view where project_city = town
);
END;
$BODY$;

```

Рис. 34. Функция запроса проектов из определённого города для экономиста

```

CREATE OR REPLACE FUNCTION public.report_economist_project_budget(
    x1 integer,
    x2 integer)
    RETURNS TABLE(project_name character, project_place character,
        project_address character, project_budget double precision)
    LANGUAGE 'plpgsql'
    COST 100
    VOLATILE PARALLEL UNSAFE
    ROWS 1000

AS $BODY$
BEGIN
RETURN QUERY (
    SELECT * FROM economist_project_view where (budget >= x1) and (budget <= x2)
);
END;
$BODY$;

```

Рис. 35. Функция запроса проектов с ценой из заданного интервала для экономиста

```

CREATE OR REPLACE FUNCTION public.report_economist_all_supplies(
)
RETURNS TABLE(component_name character, component_color character,
               is_expensive character varying, provider_name character,
               provider_city character, is_reliable character varying,
               qty integer, weight_supplies numeric, components_price numeric)
LANGUAGE 'plpgsql'
COST 100
VOLATILE PARALLEL UNSAFE
ROWS 1000

AS $BODY$
BEGIN
RETURN QUERY (
    SELECT * FROM economist_supply_view
);
END;
$BODY$;

```

Рис. 36. Функция запроса всех поставок для экономиста

```

CREATE OR REPLACE FUNCTION public.report_economist_supplies_expensive(
expensive character varying)
RETURNS TABLE(component_name character, component_color character,
               isexpensive character varying, provider_name character,
               provider_city character, isreliable character varying,
               qty integer, weight_supplies numeric, components_price numeric)
LANGUAGE 'plpgsql'
COST 100
VOLATILE PARALLEL UNSAFE
ROWS 1000

AS $BODY$
BEGIN
RETURN QUERY (
    SELECT * FROM economist_supply_view where is_expensive = expensive
);
END;
$BODY$;

```

Рис. 37. Функция запроса (не)дорогих поставок для экономиста

```

CREATE OR REPLACE FUNCTION public.report_economist_supplies_reliable(
    reliable character varying)
    RETURNS TABLE(component_name character, component_color character,
        isexpensive character varying, provider_name character,
        provider_city character, isreliable character varying,
        qty integer, weight_supplies numeric, components_price numeric)
    LANGUAGE 'plpgsql'
    COST 100
    VOLATILE PARALLEL UNSAFE
    ROWS 1000

AS $BODY$
BEGIN
RETURN QUERY (
    SELECT * FROM economist_supply_view where is_reliable = reliable
);
END;
$BODY$;

```

Рис. 38. Функция запроса поставок от (не) надёжных поставщиков для экономиста

```

CREATE OR REPLACE FUNCTION public.report_economist_supplies_price(
    p1 numeric,
    p2 numeric)
    RETURNS TABLE(component_name character, component_color character,
        isexpensive character varying, provider_name character,
        provider_city character, isreliable character varying,
        qty integer, weight_supplies numeric, components_price numeric)
    LANGUAGE 'plpgsql'
    COST 100
    VOLATILE PARALLEL UNSAFE
    ROWS 1000

AS $BODY$
BEGIN
RETURN QUERY (
    SELECT * FROM economist_supply_view
    where (component_price_thousands >= p1) and (component_price_thousands <= p2)
);
END;
$BODY$;

```

Рис. 39. Функция запроса поставок со стоимостью из заданного диапазона для экономиста


```

CREATE OR REPLACE FUNCTION public.report_director_all_provider(
)
RETURNS TABLE(provider_name character, provider_city character,
               is_reliable character varying, sum_price_sullies numeric)
LANGUAGE 'plpgsql'
COST 100
VOLATILE PARALLEL UNSAFE
ROWS 1000

AS $BODY$
BEGIN
RETURN QUERY (
    SELECT director_provider_view2.provider_name,
           director_provider_view2.provider_city,
           director_provider_view2.is_reliable, director_provider_view2.total
    FROM director_provider_view2);
END;
$BODY$;

```

Рис. 40. Функция запроса всех поставщиков для директора

```

CREATE OR REPLACE FUNCTION public.report_director_less_rating_provider(
)
RETURNS TABLE(provider_name character, provider_city character,
               is_reliable character varying, sum_price_sullies numeric)
LANGUAGE 'plpgsql'
COST 100
VOLATILE PARALLEL UNSAFE
ROWS 1000

AS $BODY$
BEGIN
RETURN QUERY (
    SELECT director_provider_view2.provider_name,
           director_provider_view2.provider_city,
           director_provider_view2.is_reliable, director_provider_view2.total
    FROM director_provider_view2
    WHERE director_provider_view2.rating <= (select avg(director_provider_view2.rating)
                                             from director_provider_view2)
);
END;
$BODY$;

```

Рис. 41. Функция запроса поставщиков с рейтингом не выше среднего для директора

```

CREATE OR REPLACE FUNCTION public.report_director_more_rating_provider(
)
RETURNS TABLE(provider_name character, provider_city character,
                is_reliable character varying, sum_price_sullies numeric)
LANGUAGE 'plpgsql'
COST 100
VOLATILE PARALLEL UNSAFE
ROWS 1000

AS $BODY$
BEGIN
RETURN QUERY (
    SELECT director_provider_view2.provider_name,
           director_provider_view2.provider_city,
           director_provider_view2.is_reliable, director_provider_view2.total
    FROM director_provider_view2
    WHERE director_provider_view2.rating >= (select avg(director_provider_view2.rating)
                                           from director_provider_view2)
);
END;
$BODY$;

```

Рис. 42. Функция запроса поставщиков с рейтингом не ниже среднего для директора

```

CREATE OR REPLACE FUNCTION public.report_director_max_rating_provider(
)
RETURNS TABLE(provider_name character, provider_city character,
                is_reliable character varying, sum_price_sullies numeric)
LANGUAGE 'plpgsql'
COST 100
VOLATILE PARALLEL UNSAFE
ROWS 1000

AS $BODY$
BEGIN
RETURN QUERY (
    SELECT director_provider_view2.provider_name,
           director_provider_view2.provider_city,
           director_provider_view2.is_reliable, director_provider_view2.total
    FROM director_provider_view2
    WHERE director_provider_view2.rating = (select max(director_provider_view2.rating)
                                           from director_provider_view2)
);
END;
$BODY$;

```

Рис. 43. Функция запроса поставщиков с максимальным рейтингом для директора

```

CREATE OR REPLACE FUNCTION public.report_director_min_rating_provider(
)
RETURNS TABLE(provider_name character, provider_city character,
                is_reliable character varying, sum_price_sullies numeric)
LANGUAGE 'plpgsql'
COST 100
VOLATILE PARALLEL UNSAFE
ROWS 1000

AS $BODY$
BEGIN
RETURN QUERY (
    SELECT director_provider_view2.provider_name,
           director_provider_view2.provider_city,
           director_provider_view2.is_reliable, director_provider_view2.total
    FROM director_provider_view2
    WHERE director_provider_view2.rating = (select min(director_provider_view2.rating)
                                           from director_provider_view2)
);
END;
$BODY$;

```

Рис. 44. Функция запроса поставщиков с минимальным рейтингом для директора

```

CREATE OR REPLACE FUNCTION public.report_director_all_component(
)
RETURNS TABLE(component_name character,
                component_price double precision,
                component_color character, is_expensive character varying,
                sum_components numeric)
LANGUAGE 'plpgsql'
COST 100
VOLATILE PARALLEL UNSAFE
ROWS 1000

AS $BODY$
BEGIN
RETURN QUERY (
    SELECT director_component_view2.component_name,
           director_component_view2.component_price_thousands,
           director_component_view2.component_color,
           director_component_view2.is_expensive, director_component_view2.total
    FROM director_component_view2
);
END;
$BODY$;

```

Рис. 45. Функция запроса всех деталей для директора

```

CREATE OR REPLACE FUNCTION public.report_director_less_price_component(
)
RETURNS TABLE(component_name character, component_price double precision,
                component_color character, is_expensive character varying,
                sum_components numeric)
LANGUAGE 'plpgsql'
COST 100
VOLATILE PARALLEL UNSAFE
ROWS 1000

AS $BODY$
BEGIN
RETURN QUERY (
    SELECT director_component_view2.component_name,
           director_component_view2.component_price_thousands,
           director_component_view2.component_color,
           director_component_view2.is_expensive, director_component_view2.total
    FROM director_component_view2
    WHERE director_component_view2.component_price_thousands <=
           (select AVG(director_component_view2.component_price_thousands)
            from director_component_view2)
);
END;
$BODY$;

```

Рис. 46. Функция запроса деталей с ценой не выше средней для директора

```

CREATE OR REPLACE FUNCTION public.report_director_more_price_component(
)
RETURNS TABLE(component_name character,
                component_price double precision,
                component_color character,
                is_expensive character varying, sum_components numeric)
LANGUAGE 'plpgsql'
COST 100
VOLATILE PARALLEL UNSAFE
ROWS 1000

AS $BODY$
BEGIN
RETURN QUERY (
    SELECT director_component_view2.component_name,
           director_component_view2.component_price_thousands,
           director_component_view2.component_color,
           director_component_view2.is_expensive, director_component_view2.total
    FROM director_component_view2
    WHERE director_component_view2.component_price_thousands >=
           (select AVG(director_component_view2.component_price_thousands)
            from director_component_view2)
);
END;
$BODY$;

```

Рис. 47. Функция запроса деталей с ценой не ниже средней для директора

```

CREATE OR REPLACE FUNCTION public.report_director_max_price_component(
)
RETURNS TABLE(component_name character,
                component_price double precision,
                component_color character,
                is_expensive character varying, sum_components numeric)
LANGUAGE 'plpgsql'
COST 100
VOLATILE PARALLEL UNSAFE
ROWS 1000

AS $BODY$
BEGIN
RETURN QUERY (
    SELECT director_component_view2.component_name,
           director_component_view2.component_price_thousands,
           director_component_view2.component_color,
           director_component_view2.is_expensive, director_component_view2.total
    FROM director_component_view2
    WHERE director_component_view2.component_price_thousands =
      (select max(director_component_view2.component_price_thousands)
       from director_component_view2)
);
END;
$BODY$;

```

Рис. 48. Функция запроса деталей максимальной ценой для директора

```

CREATE OR REPLACE FUNCTION public.report_director_min_price_component(
)
RETURNS TABLE(component_name character, component_price double precision,
                component_color character, is_expensive character varying,
                sum_components numeric)
LANGUAGE 'plpgsql'
COST 100
VOLATILE PARALLEL UNSAFE
ROWS 1000

AS $BODY$
BEGIN
RETURN QUERY (
    SELECT director_component_view2.component_name,
           director_component_view2.component_price_thousands,
           director_component_view2.component_color,
           director_component_view2.is_expensive, director_component_view2.total
    FROM director_component_view2
    WHERE director_component_view2.component_price_thousands =
      (select min(director_component_view2.component_price_thousands)
       from director_component_view2)
);
END;
$BODY$;

```

Рис. 49. Функция запроса деталей минимальной ценой для директора

```

CREATE OR REPLACE FUNCTION public.report_director_all_project(
)
  RETURNS TABLE(project_name character, project_city character,
                 budget double precision, sum_price_sullies numeric)
  LANGUAGE 'plpgsql'
  COST 100
  VOLATILE PARALLEL UNSAFE
  ROWS 1000

AS $BODY$
BEGIN
RETURN QUERY (
  SELECT director_project_view2.project_name, director_project_view2.project_city,
         director_project_view2.budget, director_project_view2.total
  FROM director_project_view2

);
END;
$BODY$;

```

Рис. 50. Функция запроса всех проектов для директора

```

CREATE OR REPLACE FUNCTION public.report_director_less_budget_project(
)
  RETURNS TABLE(project_name character, project_city character,
                 budget double precision, sum_price_sullies numeric)
  LANGUAGE 'plpgsql'
  COST 100
  VOLATILE PARALLEL UNSAFE
  ROWS 1000

AS $BODY$
BEGIN
RETURN QUERY (
  SELECT director_project_view2.project_name,
         director_project_view2.project_city, director_project_view2.budget,
         director_project_view2.total
  FROM director_project_view2
  where director_project_view2.budget <= (select avg(director_project_view2.budget)
                                         from director_project_view2)

);
END;
$BODY$;

```

Рис. 51. Функция запроса проектов с бюджетом не выше среднего для директора

```

CREATE OR REPLACE FUNCTION public.report_director_more_budget_project(
)
  RETURNS TABLE(project_name character, project_city character,
                budget double precision, sum_price_sullies numeric)
  LANGUAGE 'plpgsql'
  COST 100
  VOLATILE PARALLEL UNSAFE
  ROWS 1000

AS $BODY$
BEGIN
RETURN QUERY (
  SELECT director_project_view2.project_name,
    director_project_view2.project_city, director_project_view2.budget,
    director_project_view2.total
  FROM director_project_view2
  where director_project_view2.budget >= (select avg(director_project_view2.budget)
    from director_project_view2)

);
END;
$BODY$;

```

Рис. 52. Функция запроса проектов с бюджетом не ниже среднего для директора

```

CREATE OR REPLACE FUNCTION public.report_director_max_budget_project(
)
  RETURNS TABLE(project_name character, project_city character,
                budget double precision, sum_price_sullies numeric)
  LANGUAGE 'plpgsql'
  COST 100
  VOLATILE PARALLEL UNSAFE
  ROWS 1000

AS $BODY$
BEGIN
RETURN QUERY (
  SELECT director_project_view2.project_name,
    director_project_view2.project_city, director_project_view2.budget,
    director_project_view2.total
  FROM director_project_view2
  where director_project_view2.budget = (select max(director_project_view2.budget)
    from director_project_view2)

);
END;
$BODY$;

```

Рис. 53. Функция запроса проектов с максимальным бюджетом для директора

```

CREATE OR REPLACE FUNCTION public.report_director_min_budget_project(
)
RETURNS TABLE(project_name character, project_city character,
                budget double precision, sum_price_sullies numeric)
LANGUAGE 'plpgsql'
COST 100
VOLATILE PARALLEL UNSAFE
ROWS 1000

AS $BODY$
BEGIN
RETURN QUERY (
    SELECT director_project_view2.project_name, director_project_view2.project_city,
    director_project_view2.budget, director_project_view2.total
    FROM director_project_view2
    where director_project_view2.budget = (select min(director_project_view2.budget)
                                         from director_project_view2)

);
END;
$BODY$;

```

Рис. 54. Функция запроса проектов с максимальным бюджетом для директора

```

CREATE OR REPLACE FUNCTION public.report_director_sum_all_supplies(
)
RETURNS TABLE(sum_all_supplies numeric)
LANGUAGE 'plpgsql'
COST 100
VOLATILE PARALLEL UNSAFE
ROWS 1000

AS $BODY$
BEGIN
RETURN QUERY (
    select sum(component_price_thousands) as total
    from director_supply_view
);
END;
$BODY$;

```

Рис. 55. Функция запроса суммы всех поставок для директора


```

CREATE OR REPLACE FUNCTION public.report_director_sum_expensive_or_cheap_supplies(
    expensive character varying)
    RETURNS TABLE(sum_all_supplies numeric)
    LANGUAGE 'plpgsql'
    COST 100
    VOLATILE PARALLEL UNSAFE
    ROWS 1000

AS $BODY$
BEGIN
RETURN QUERY (
    select sum(component_price_thousands) as total
    from director_supply_view
    where is_expensive = Expensive
);
END;
$BODY$;

```

Рис. 56. Функция запроса суммы (не)дорогих поставок для директора

```

CREATE OR REPLACE FUNCTION public.report_director_sum_un_reliable_supplies(
    reliable character varying)
    RETURNS TABLE(sum_all_supplies numeric)
    LANGUAGE 'plpgsql'
    COST 100
    VOLATILE PARALLEL UNSAFE
    ROWS 1000

AS $BODY$
BEGIN
RETURN QUERY (
    select sum(component_price_thousands) as total
    from director_supply_view
    where is_reliable = Reliable
);
END;
$BODY$;

```

Рис. 57. Функция запроса суммы поставок от (не)надёжных поставщиков для директора

```

CREATE OR REPLACE FUNCTION public.report_director_sum_provider_supplies(
    provider character)
    RETURNS TABLE(sum_all_supplies numeric)
    LANGUAGE 'plpgsql'
    COST 100
    VOLATILE PARALLEL UNSAFE
    ROWS 1000

AS $BODY$
BEGIN
RETURN QUERY (
    select sum(component_price_thousands) as total
    from director_supply_view
    where provider_name = provider
);
END;
$BODY$;

```

Рис. 58. Функция запроса суммы поставок от заданного поставщика для директора

	component_name character	component_price double precision	component_color character	is_expensive character varying	sum_components numeric
1	warm floor	70	brown	Expensive	630.000
2	Radiator	15	white	Expensive	165.000
3	toilet	14	black	Expensive	238.000
4	water heater	14	red	Expensive	98.000
5	sink	13.499	red	Expensive	175.487
6	tank	12.14	grey	Expensive	36.420
7	heated towel rail	9.63	grey	Expensive	48.150
8	bath	9.46	white	Expensive	47.300
9	bidet	7.3	white	Expensive	65.700
10	chimney	5.61	grey	Expensive	28.050

Рис. 59. Пример использования функции запроса всех деталей для директора






	component_name character 	component_price double precision 	component_color character 	weight_kg double precision 	is_expensive character varying 
1	warm floor	70	brown	28	Expensive
2	Radiator	15	white	44.5	Expensive
3	toilet	14	black	11	Expensive
4	water heater	14	red	10.3	Expensive
5	sink	13.499	red	8.4	Expensive
6	tank	12.14	grey	57	Expensive
7	heated towel rail	9.63	grey	4.88	Expensive
8	bath	9.46	white	18	Expensive
9	bidet	7.3	white	7.5	Expensive
10	chimney	5.61	grey	4.2	Expensive
11	mixer	2.568	red	1.57	Expensive
12	Sheet metal	2.054	grey	2.916	Expensive
13	parquet board	1.079	brown	0.71	Expensive
14	glue packaging	0.893	green	1.5	Cheap
15	bath screen	0.849	blue	5.6	Cheap

Рис. 60. Пример использования функции запроса всех деталей для экономиста