1 Цель работы

Выработка и закрепление практических навыков по работе с реляционными базами данных, изучить основы языка запросов SQL, научиться создавать таблицы и осуществлять элементарные выборки.

2 Ход выполнения работы

Был выдан вариант 3.

2.1 Часть 1

Для базы данных была составлена ER-диаграмма, представленная на рисунке 1.

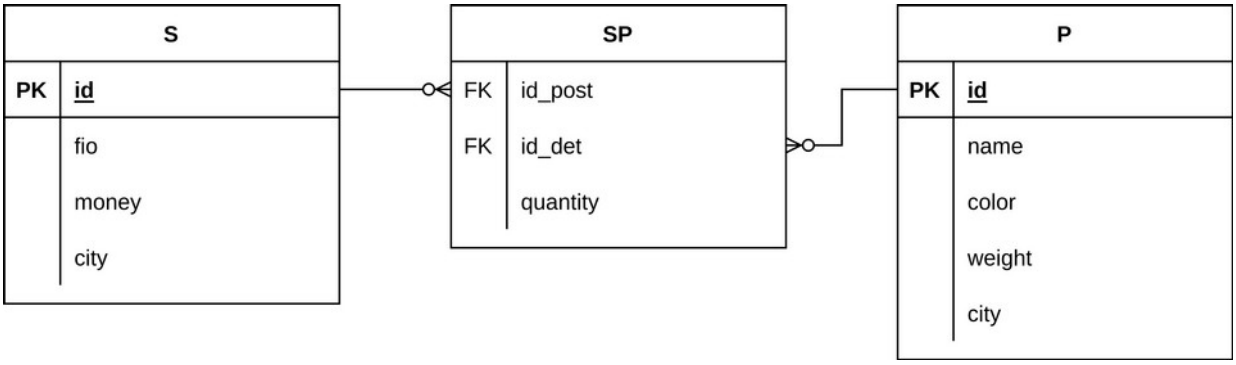


Рисунок 1 — ER-диаграмма для базы данных 1

Задача 1: создать базу данных.

Запрос:

```
CREATE DATABASE './company.fdb'
    USER 'sysdba' PASSWORD 'masterkey'
    page_size = 4096
    DEFAULT CHARACTER SET UTF8;
```

Результат выполнения запроса на рисунке 2.

```
 /public_folder isql-fb
Use CONNECT or CREATE DATABASE to specify a database
SQL> CREATE DATABASE './company.fdb'
CON>     USER 'sysdba' PASSWORD 'masterkey'
CON>     page_size = 4096
CON>     DEFAULT CHARACTER SET UTF8;
SQL> ^D
 /public_folder ll
итого 784
drwxrwxrwt 1 root     root         22 мая 12 12:57 ./
drwxr-xr-x 1 root     root        252 мая 12 12:55 ../
-rw-rw---- 1 firebird firebird 802816 мая 12 12:57 company.fdb
```

Рисунок 2 — Результат выполнения запроса задачи 1

Задача 2: создать таблицы и заполнить их значениями, представленными в таблицах S (suppliers), P (details), SP (supplier_detail).

Транзакция создания таблиц:

```
CREATE TABLE suppliers (
    id INTEGER NOT NULL,
    fio VARCHAR(20) NOT NULL,
    money INTEGER NOT NULL,
    city VARCHAR(15) NOT NULL,
    PRIMARY KEY (id)
);
CREATE TABLE details (
    id INTEGER NOT NULL,
    name VARCHAR(20) NOT NULL,
    color CHAR(6) NOT NULL,
    weight INTEGER NOT NULL,
    city VARCHAR(15) NOT NULL,
        PRIMARY KEY (id)
);
CREATE TABLE supplier_detail (
    id_s INTEGER NOT NULL,
    id_d INTEGER NOT NULL,
    quantity INTEGER NOT NULL,
        FOREIGN KEY (id_s) REFERENCES suppliers(id)
            ON DELETE CASCADE
            ON UPDATE CASCADE,
        FOREIGN KEY (id_d) REFERENCES details(id)
            ON DELETE CASCADE
            ON UPDATE CASCADE
);
```

Результат выполнения запроса на рисунке 3.

```
SQL> SHOW TABLES;
       DETAILS                                 SUPPLIERS
       SUPPLIER_DETAIL
```

Рисунок 3 — Результат выполнения запроса задачи 2

Транзакция заполнения таблиц данными:

```
INSERT INTO suppliers VALUES (1, 'Иванов И.И.',   50000,  'Лондон');
INSERT INTO suppliers VALUES (2, 'Петров П.П.',   20000,  'Москва');
INSERT INTO suppliers VALUES (3, 'Сидоров С.С.',  150000, 'Париж');
INSERT INTO suppliers VALUES (4, 'Васильев В.В.', 300000, 'Лондон');
INSERT INTO suppliers VALUES (5, 'Семенов С.С.',  70000,  'Киев');
INSERT INTO suppliers VALUES (6, 'К. Ласки',      650000, 'Нью-Йорк');
INSERT INTO suppliers VALUES (7, 'Ф. де Костер',  10000,  'Нью-Йорк');
INSERT INTO suppliers VALUES (8, 'Дзержинский Ф.Э.', 10000,  'Москва');
INSERT INTO suppliers VALUES (123, 'Ульянов В.И.',  300000,  'Севастополь');
INSERT INTO details   VALUES (1, 'Болт',        '000000', 34, 'Киев');
INSERT INTO details   VALUES (4, 'Винт',        'ff0000', 67, 'Киев');
INSERT INTO details   VALUES (3, 'Га*ка малая', 'ffff00', 17, 'Лондон');
INSERT INTO details   VALUES (2, 'Гвоздь',      'ffffff', 56, 'Париж');
INSERT INTO details   VALUES (5, 'Петля',       '00ff00', 90, 'Москва');
INSERT INTO details   VALUES (6, 'Га*ка большая','00ff00',25, 'Москва');
INSERT INTO details   VALUES (7, 'Пор*ень',     '00ff00', 120, 'Нью-Йорк');
INSERT INTO details   VALUES (8, 'Вал',         '00ff00', 230, 'Нью-Йорк');
INSERT INTO details   VALUES (9, 'Труба',       '00ff00', 50, 'Киев');
INSERT INTO details   VALUES (10, 'Трубка',     '00ff00', 13, 'Нью-Йорк');
INSERT INTO details   VALUES (11, 'Винт',       'ffffff', 50, 'Севастополь');
INSERT INTO details   VALUES (12, 'Болт',       'ffffff', 50, 'Москва');
INSERT INTO supplier_detail VALUES (2, 3, 120);
INSERT INTO supplier_detail VALUES (2, 4, 190);
INSERT INTO supplier_detail VALUES (2, 5, 15);
INSERT INTO supplier_detail VALUES (1, 4, 18);
INSERT INTO supplier_detail VALUES (3, 5, 190);
INSERT INTO supplier_detail VALUES (5, 5, 200);
INSERT INTO supplier_detail VALUES (1, 5, 100);
INSERT INTO supplier_detail VALUES (2, 2, 20);
INSERT INTO supplier_detail VALUES (3, 1, 150);
INSERT INTO supplier_detail VALUES (6, 1, 621);
INSERT INTO supplier_detail VALUES (6, 2, 34);
INSERT INTO supplier_detail VALUES (7, 4, 69);
INSERT INTO supplier_detail VALUES (4, 3, 30);
INSERT INTO supplier_detail VALUES (5, 4, 18);
INSERT INTO supplier_detail VALUES (2, 1, 25);
INSERT INTO supplier_detail VALUES (1, 3, 14);
INSERT INTO supplier_detail VALUES (4, 5, 22);
INSERT INTO supplier_detail VALUES (1, 2, 67);
INSERT INTO supplier_detail VALUES (3, 3, 15);
INSERT INTO supplier_detail VALUES (4, 1, 18);
INSERT INTO supplier_detail VALUES (5, 1, 217);
INSERT INTO supplier_detail VALUES (8, 8, 200);
INSERT INTO supplier_detail VALUES (2, 12, 926);
INSERT INTO supplier_detail VALUES (123, 11, 590);
INSERT INTO supplier_detail VALUES (123, 5, 621);
INSERT INTO supplier_detail VALUES (123, 2, 267);
INSERT INTO supplier_detail VALUES (123, 1, 409);
```

Задача 3: выдать номера всех поставляемых деталей.

Запрос:

```
SELECT id FROM details ORDER BY id;
```

Результат выполнения запроса на рисунке 4.

```
SQL> SELECT id FROM details ORDER BY id;

          ID
============
           1
           2
           3
           4
           5
           6
           7
           8
           9
          10
          11
          12
```

Рисунок 4 — Результат выполнения запроса задачи 3

Задача 4: добавить столбец «материал» в таблицу P.

Запрос:

```
ALTER TABLE details
    ADD material VARCHAR(15);
```

Результат выполнения запроса на рисунке 5.

```
SQL> ALTER TABLE details
CON>     ADD material VARCHAR(15);
SQL> SHOW TABLE details;
ID                              INTEGER Not Null
NAME                            VARCHAR(20) Not Null
COLOR                           CHAR(6) Not Null
WEIGHT                          INTEGER Not Null
CITY                            VARCHAR(15) Not Null
MATERIAL                        VARCHAR(15) Nullable
CONSTRAINT INTEG_11:
  Primary key (ID)
```

Рисунок 5 — Результат выполнения запроса задачи 4

Задача 5: выделить всех поставщиков, не проживающих в Париже.

Запрос:

```
SELECT id, city FROM suppliers
    WHERE city != 'Париж';
```

Результат выполнения запроса на рисунке 6.

```
SQL> SELECT id, city FROM suppliers
CON>     WHERE city != 'Париж';

          ID CITY
============ ======================
           1 Лондон
           2 Москва
           4 Лондон
           5 Киев
           6 Нью-Йорк
           7 Нью-Йорк
           8 Москва
         123 Севастополь
```

Рисунок 6 — Результат выполнения запроса задачи 5

Задача 6: выдать номера и состояния для поставщиков, находящихся в Лондоне.

Запрос:

```
SELECT id, money FROM suppliers
    WHERE city = 'Лондон';
```

Результат выполнения запроса на рисунке 7.

```
SQL> SELECT id, money FROM suppliers
CON>     WHERE city = 'Лондон';

          ID        MONEY
============ ============
           1        50000
           4       300000
```

Рисунок 7 — Результат выполнения запроса задачи 6

Задача 7: выдать номера и состояния для поставщиков, проживающих в Москве, и состояние которых меньше 30 тыс.$ (в порядке убывания состояний).

Запрос:

```
SELECT id, money FROM suppliers
    WHERE city = 'Москва' AND
          money < 30000
    ORDER BY money DESC;
```

Результат выполнения запроса на рисунке 8.

```
SQL> SELECT id, money FROM suppliers
CON>     WHERE city = 'Москва' AND
CON>           money < 30000
CON>     ORDER BY money DESC;

           ID        MONEY
============ ============
            2        20000
            8        10000
```

Рисунок 8 — Результат выполнения запроса задачи 7

Задача 8: выдать полные характеристики всех поставщиков.

Запрос:

```
SELECT 'Поставщик ' || suppliers.fio || ' поставляет ' || details.name || ' из '
|| details.city FROM suppliers JOIN supplier_detail ON suppliers.id =
supplier_detail.id_s JOIN details  ON details.id = supplier_detail.id_d;
```

Результат выполнения запроса на рисунке 9.

```
SQL> SELECT 'Поставщик ' || suppliers.fio || ' поставляет ' || details.name || ' из ' || details.city
CON>     FROM suppliers JOIN supplier_detail ON suppliers.id = supplier_detail.id_s
CON>            JOIN details  ON details.id = supplier_detail.id_d;

CONCATENATION

================================================================================
Поставщик Иванов И.И. поставляет Винт из Киев

Поставщик Иванов И.И. поставляет Петля из Москва

Поставщик Иванов И.И. поставляет Га*ка малая из Лондон

Поставщик Иванов И.И. поставляет Гвоздь из Париж

Поставщик Петров П.П. поставляет Га*ка малая из Лондон

Поставщик Петров П.П. поставляет Винт из Киев

Поставщик Петров П.П. поставляет Петля из Москва

Поставщик Петров П.П. поставляет Гвоздь из Париж
```

Рисунок 9 — Результат выполнения запроса задачи 8

Задача 9: выдать сведения о деталях, вес которых в диапазоне от 20 до 50.

Запрос:

```
SELECT * FROM details WHERE weight BETWEEN 20 AND 50;
```

Результат выполнения запроса на рисунке 10.

```
SQL> SELECT * FROM details
CON>     WHERE weight BETWEEN 20 AND 50;

          ID NAME                                                                                COLOR                WEIGHT CITY
============ ================================================================================ ==================== ============ =========
           1 Болт                                                                               000000                34 Киев
           6 Га*ка большая                                                                    00ff00               25 Москва
           9 Труба                                                                              00ff00                50 Киев
          11 Винт                                                                               ffffff                50 Севастополь
          12 Болт                                                                               ffffff                50 Москва
```

Рисунок 10 — Результат выполнения запроса задачи 9

Задача 10: выдать номер и вес каждой детали в граммах для всех деталей, если вес указан в фунтах (454 гр.).

Запрос:

```
SELECT id, weight * 454 FROM details;
```

Результат выполнения запроса на рисунке 11.

```
SQL> SELECT id, weight * 454
CON>     FROM details;

          ID             MULTIPLY
============ ====================
           1                15436
           4                30418
           3                 7718
           2                25424
           5                40860
           6                11350
           7                54480
           8               104420
           9                22700
          10                 5902
          11                22700
          12                22700
```

Рисунок 11 — Результат выполнения запроса задачи 10

Задача 11: выдать номера деталей, вес которых 13, 17, 25.

Запрос:

```
SELECT id FROM details WHERE weight IN (13, 17, 25);
```

Результат выполнения запроса на рисунке 12.

```
SQL> SELECT id FROM details
CON>     WHERE weight IN (13, 17, 25);

          ID
============
           3
           6
          10
```

Рисунок 12 — Результат выполнения запроса задачи 11

Задача 12: выдать все детали, название которых начинается с буквы «В».

Запрос:

```
SELECT id FROM details WHERE (name LIKE '%B%');
```

Результат выполнения запроса на рисунке 13.

```
SQL> SELECT id FROM details
CON>     WHERE (name LIKE '%B%');

          ID
============
           4
           8
          11
```

Рисунок 13 — Результат выполнения запроса задачи 12

Задача 13: выдать все детали, название которых заканчивается на букву «Я».

Запрос:

```
SELECT id FROM details WHERE (name LIKE '%я');
```

Результат выполнения запроса на рисунке 14.

```
SQL> SELECT id FROM details
CON>     WHERE (name LIKE '%я');

           ID
============
            3
            5
            6
```

Рисунок 14 — Результат выполнения запроса задачи 13

Задача 14: выдать все детали, название которых содержит в середине букву «а».

Запрос:

```
SELECT id FROM details WHERE (name LIKE '%a%');
```

Результат выполнения запроса на рисунке 15.

```
SQL> SELECT id FROM details
CON>     WHERE (name LIKE '%a%');

           ID
============
            3
            6
            8
            9
           10
```

Рисунок 15 — Результат выполнения запроса задачи 14

Задача 15: выдать все детали, название которых содержит третью букву «з».

Запрос:

```
SELECT id FROM details WHERE (name LIKE '___з%');
```

Результат выполнения запроса на рисунке 16.

```
SQL> SELECT id FROM details
CON>     WHERE (name LIKE '___з%');

          ID
============
           2
```

Рисунок 16 — Результат выполнения запроса задачи 15

Задача 16: выдать все детали, название которых содержит «_».

Запрос:

```
SELECT id FROM details WHERE (name LIKE '% %');
```

Результат выполнения запроса на рисунке 17.

```
SQL> SELECT id FROM details
CON>     WHERE (name LIKE '% %');

          ID
============
           3
           6
```

Рисунок 17 — Результат выполнения запроса задачи 16

Задача 17: Выдать все комбинации информации о таких поставщиках и деталях, которые размещены в одном и том же городе.

Запрос:

```
SELECT details.id, suppliers.id
    FROM details JOIN suppliers ON details.city = suppliers.city;
```

Результат выполнения запроса на рисунке 18.

```
SQL> SELECT details.id, suppliers.id
CON>     FROM details
CON>         JOIN suppliers ON details.city = suppliers.city;

           ID           ID
============ ============
            1            5
            4            5
            9            5
            3            1
            3            4
            5            2
            5            8
            6            2
            6            8
           12            2
           12            8
            7            6
            7            7
            8            6
            8            7
           10            6
           10            7
            2            3
           11          123
```

Рисунок 18 — Результат выполнения запроса задачи 17

Задача 18: изменить цвет детали 2 на жёлтый(#FFFF00), увеличить её вес на три и установить значение города – неизвестен.

Запрос:

```
UPDATE details SET city='Неизвестно' WHERE id = 2;
UPDATE details SET weight=(weight+3) WHERE id = 2;
UPDATE details SET color='ffff00'    WHERE id = 2;
```

Результат выполнения запроса на рисунке 19.

```
SQL> UPDATE details SET city='Неизвестно' WHERE id = 2;
TE details SET weight=(weSQL> UPDATE details SET weight=(weight+3) WHERE id = 2;
DATE details SETSQL> UPDATE details SET color='ffff00'    WHERE id = 2;
SQL> SELECT * FROM details WHERE id = 2;

         ID NAME                                                                                COLOR                WEIGHT CITY

============ ================================================================================ ==================== ============ =====
==================================================
          2 Гвоздь                                                                            ffff00                 59 Неизвестно
```

Рисунок 19 — Результат выполнения запроса задачи 18

Задача 19: удалить столбец «материал» из таблицы P.

Запрос:

```
ALTER TABLE details DROP material;
```

Результат выполнения запроса на рисунке 20.

```
SQL> ALTER TABLE details
CON>     DROP material;
SQL> SHOW TABLE details;
ID                              INTEGER Not Null
NAME                            VARCHAR(20) Not Null
COLOR                           CHAR(6) Not Null
WEIGHT                          INTEGER Not Null
CITY                            VARCHAR(15) Not Null
CONSTRAINT INTEG_11:
  Primary key (ID)
```

Рисунок 20 — Результат выполнения запроса задачи 19

Задача 20: увеличить в четыре раза состояние всех поставщиков, находящихся в Париже.

Запрос:

```
UPDATE suppliers
    SET money = money * 4 WHERE city = 'Париж';
```

Результат выполнения запроса на рисунке 21.

```
SQL> UPDATE suppliers
CON>     SET money = money * 4
CON>     WHERE city = 'Париж';
SQL>
```

Рисунок 21 — Результат выполнения запроса задачи 20

Задача 21: удалить всех поставщиков из Парижа.

Запрос:

```
DELETE FROM suppliers WHERE city = 'Париж';
```

Результат выполнения запроса на рисунке 22.

```
SQL> DELETE FROM suppliers WHERE city = 'Париж';
SQL> SELECT * FROM suppliers WHERE city = 'Париж';
SQL> _
```

Рисунок 22 — Результат выполнения запроса задачи 21

Задача 22: добавить в таблицу P запись: деталь – 6, город – Севастополь, цвет – #FF9900, название – лента, вес – неизвестен.

Запрос:

```
INSERT INTO details (id, city, color, name, weight) VALUES (13, 'Севастополь',
'ff9900', 'Лента', 0);
```

Результат выполнения запроса на рисунке 23.

```
SQL> INSERT INTO details (id, city, color, name, weight) VALUES (13, 'Севастополь', 'ff9900', 'Лента', 0);
SQL> SELECT id, name FROM details WHERE id = 13;

          ID NAME
============ ================================================================================
          13 Лента
```

Рисунок 23 — Результат выполнения запроса задачи 22

Задача 23: выведите номера деталей, для которых не определено значение веса.

Запрос:

```
SELECT id FROM details WHERE weight = 0;
```

Результат выполнения запроса на рисунке 24.

```
SQL> SELECT id FROM details WHERE weight = 0;

          ID
============
          13
```

Рисунок 24 — Результат выполнения запроса задачи 23

Задача 24: выдать общее количество поставщиков, поставляющих в настоящее время детали.

Запрос:

```
SELECT COUNT(*) FROM suppliers;
```

Результат выполнения запроса на рисунке 25.

```
SQL> SELECT COUNT(*) FROM suppliers;

       COUNT
============
           8
```

Рисунок 25 — Результат выполнения запроса задачи 24

Задача 25: для каждой поставляемой детали выдать её номер и объем поставок.

Запрос:

```
SELECT details.id, supplier_detail.quantity
    FROM details JOIN supplier_detail
        ON details.id = supplier_detail.id_d;
```

Результат выполнения запроса на рисунке 26.

```
SQL> SELECT details.id, supplier_detail.quantity
CON>     FROM details JOIN supplier_detail
CON>         ON details.id = supplier_detail.id_d;

          ID     QUANTITY
============ ============
           1          621
           1           25
           1           18
           1          217
           1          409
           4          190
           4           18
```

Рисунок 26 — Результат выполнения запроса задачи 25

Задача 26: для каждой детали выдать её номер и объем поставок за исключением поставщика под номером 3.

Запрос:

```
SELECT details.id, supplier_detail.quantity
    FROM details JOIN supplier_detail
        ON details.id = supplier_detail.id_d
    WHERE supplier_detail.id_s != 3;
```

Результат выполнения запроса на рисунке 27.

```
SQL> SELECT details.id, supplier_detail.quantity
CON>     FROM details JOIN supplier_detail
CON>         ON details.id = supplier_detail.id_d
CON>     WHERE supplier_detail.id_s != 3;

          ID     QUANTITY
============ ============
           3          120
           4          190
           5           15
           4           18
           5          200
           5          100
           2           20
           1          621
           2           34
           4           69
```

Рисунок 27 — Результат выполнения запроса задачи 26

Задача 27: выдать номера деталей для всех деталей, поставляемых более чем одним поставщиком.

Запрос:

```
SELECT sd.id_d
    FROM supplier_detail sd
    GROUP BY sd.id_d
    HAVING COUNT(DISTINCT sd.id_s) > 1;
```

Результат выполнения запроса на рисунке 28.

```
SQL> SELECT sd.id_d
CON>     FROM supplier_detail sd
CON>     GROUP BY sd.id_d
CON>     HAVING COUNT(DISTINCT sd.id_s) > 1;

        ID_D
============
           1
           2
           3
           4
           5
```

Рисунок 28 — Результат выполнения запроса задачи 27

Задача 28: установить связи между таблицами S, P и SP (внешние ключи).

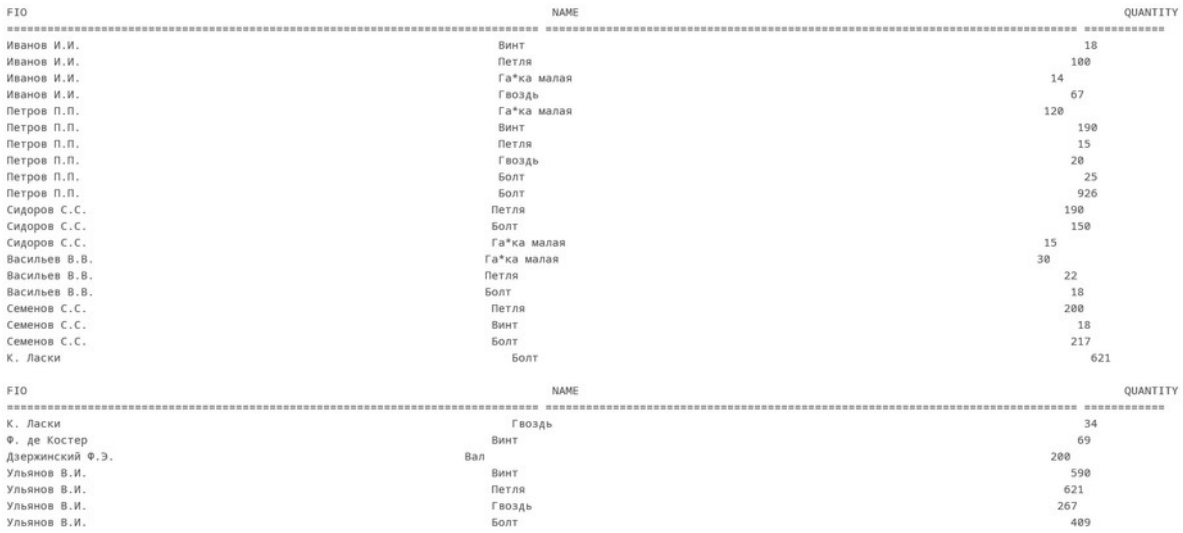
Связи между таблицами были установлены во время их создания.

Задача 29: вывести информацию об именах поставщиков и о том, в каком количестве и какие детали они поставляют.

Запрос:

```
SELECT s.fio, d.name, sd.quantity
    FROM suppliers s
            JOIN supplier_detail sd ON s.id = sd.id_s
        JOIN details d ON d.id = sd.id_d;
```

Результат выполнения запроса на рисунке 29.



FIO	NAME	QUANTITY
Иванов И.И.	Винт	18
Иванов И.И.	Петля	100
Иванов И.И.	Га*ка малая	14
Иванов И.И.	Гвоздь	67
Петров П.П.	Га*ка малая	120
Петров П.П.	Винт	190
Петров П.П.	Петля	15
Петров П.П.	Гвоздь	20
Петров П.П.	Болт	25
Петров П.П.	Болт	926
Сидоров С.С.	Петля	190
Сидоров С.С.	Болт	150
Сидоров С.С.	Га*ка малая	15
Васильев В.В.	Га*ка малая	30
Васильев В.В.	Петля	22
Васильев В.В.	Болт	18
Семенов С.С.	Петля	200
Семенов С.С.	Винт	18
Семенов С.С.	Болт	217
К. Ласки	Болт	621

FIO	NAME	QUANTITY
К. Ласки	Гвоздь	34
Ф. де Костер	Винт	69
Дзержинский Ф.Э.	Вал	200
Ульянов В.И.	Винт	590
Ульянов В.И.	Петля	621
Ульянов В.И.	Гвоздь	267
Ульянов В.И.	Болт	409

Рисунок 29 — Результат выполнения запроса задачи 29

Задача 30: вывести информацию об имени поставщика и названии деталей, которые он поставляет.

Запрос:

```
SELECT s.fio, d.name
    FROM (suppliers s JOIN supplier_detail sd ON s.id = sd.id_s)
        JOIN details d ON d.id = sd.id_d;
```

Результат выполнения запроса на рисунке 30.

```
SQL> SELECT s.fio, d.name
CON>     FROM (suppliers s JOIN supplier_detail sd ON s.id = sd.id_s)
CON>         JOIN details d ON d.id = sd.id_d;

FIO                                                                              NAME
================================================================================ ================
Иванов И.И.                                                                      Винт
Иванов И.И.                                                                      Петля
Иванов И.И.                                                                      Га*ка малая
Иванов И.И.                                                                      Гвоздь
Петров П.П.                                                                      Га*ка малая
Петров П.П.                                                                      Винт
Петров П.П.                                                                      Петля
Петров П.П.                                                                      Гвоздь
Петров П.П.                                                                      Болт
Петров П.П.                                                                      Болт
Сидоров С.С.                                                                     Петля
Сидоров С.С.                                                                     Болт
Сидоров С.С.                                                                     Га*ка малая
Васильев В.В.                                                                    Га*ка малая
Васильев В.В.                                                                    Петля
```

Рисунок 30 — Результат выполнения запроса задачи 30

Задача 31: вывести все пары поставщиков, живущих в одном городе. Исключить комбинации продавцов с ними же, а также дубликаты строк, выводимые в обратном порядке.

Запрос:

```
SELECT DISTINCT s1.fio, s2.fio
    FROM suppliers s1, suppliers s2
    WHERE s1.city = s2.city
          AND s1.fio != s2.fio;
```

Результат выполнения запроса на рисунке 31.

```
SQL> SELECT DISTINCT s1.fio, s2.fio
CON>     FROM suppliers s1, suppliers s2
CON>     WHERE s1.city = s2.city
CON>           AND s1.fio != s2.fio;

FIO                                                                              FIO
================================================================================ ===============
Васильев В.В.                                                                    Иванов И.И.
Дзержинский Ф.Э.                                                                 Петров П.П.
Иванов И.И.                                                                      Васильев В.В.
К. Ласки                                                                         Ф. де Костер
Петров П.П.                                                                      Дзержинский Ф.Э.
Ф. де Костер                                                                     К. Ласки
```

Рисунок 31 — Результат выполнения запроса задачи 31

Задача 32: вывести фамилии всех поставщиков, которые поставляют детали с номером 1.

Запрос:

```
SELECT s.fio
    FROM suppliers s JOIN supplier_detail sd ON s.id = sd.id_s;
```

Результат выполнения запроса на рисунке 32.

```
SQL> SELECT s.fio
CON>     FROM suppliers s JOIN supplier_detail sd ON s.id = sd.id_s;

FIO
============================================================
Иванов И.И.
Иванов И.И.
Иванов И.И.
Иванов И.И.
Петров П.П.
Петров П.П.
Петров П.П.
Петров П.П.
Петров П.П.
Петров П.П.
Сидоров С.С.
Сидоров С.С.
Сидоров С.С.
Васильев В.В.
```

Рисунок 32 — Результат выполнения запроса задачи 32

Задача 33: вывести фамилии поставщиков, которые поставляют, по крайней мере, одну красную деталь.

Запрос:

```
яSELECT s.id, s.fio
      FROM suppliers s
            INNER JOIN supplier_detail sd
                  ON s.id = sd.id_s
            INNER JOIN details d
                  ON d.id = sd.id_d
      WHERE d.color = 'ff0000';
```

Результат выполнения запроса на рисунке 33.

```
                ID FIO
=================== ==========================
                 2 Петров П.П.
                 1 Иванов И.И.
                 7 Ф. де Костер
                 5 Семенов С.С.
```

Рисунок 33 — Результат выполнения запроса задачи 33

Задача 34: Вывести номера и фамилии поставщиков, которые поставляют по крайней мере одну деталь, поставляемую поставщиком 3.

Запрос:

```
SELECT DISTINCT sd_outer.id_s
      FROM supplier_detail sd_outer
      WHERE sd_outer.id_d IN (
        SELECT sd_inner.id_d
            FROM supplier_detail sd_inner
            WHERE sd_inner.id_s = 3
    ) AND sd_outer.id_s != 3
      ORDER BY sd_outer.id_s DESC;
```

Результат выполнения запроса на рисунке 34.

```
SQL> SELECT DISTINCT sd_outer.id_s
CON> FROM supplier_detail sd_outer
CON> WHERE sd_outer.id_d IN (
CON>         SELECT sd_inner.id_d
CON>             FROM supplier_detail sd_inner
CON>             WHERE sd_inner.id_s = 3
CON>     ) AND sd_outer.id_s != 3
CON> ORDER BY sd_outer.id_s DESC;

        ID_S
============
         123
           6
           5
           4
           2
           1
```

Рисунок 34 — Результат выполнения запроса задачи 34

Задача 35: вывести номера поставщиков, для которых в базе существует информация о номерах поставляемых деталей.

Запрос:

```
SELECT id_s
      FROM supplier_detail
      GROUP BY id_s
      HAVING count(id_d) > 0;
```

Результат выполнения запроса на рисунке 35.

```
SQL> SELECT id_s
CON> FROM supplier_detail
CON> GROUP BY id_s
CON> HAVING count(id_d) > 0;

        ID_S
============
           1
           2
           3
           4
           5
           6
           7
           8
         123
```

Рисунок 35 — Результат выполнения запроса задачи 35

Задача 36: вывести номера поставщиков, для которых не существует информация о номерах поставляемых деталей.

Запрос:

```
SELECT s.id, s.fio
      FROM suppliers s
            LEFT JOIN supplier_detail sd
            ON sd.id_s = s.id
      WHERE sd.id_d IS NULL;
```

Результат выполнения запроса на рисунке 36.

```
SQL> SELECT s.id, s.fio
CON> FROM suppliers s LEFT JOIN supplier_detail sd ON sd.id_s = s.id WHERE sd.id_d IS NULL;
SQL> _
```

Рисунок 36 — Результат выполнения запроса задачи 36

Задача 37: вывести номера поставщиков, которые поставляют все детали.

Запрос:

```
SELECT sd.id_s
      FROM supplier_detail sd
      GROUP BY sd.id_s
      HAVING (
        SELECT COUNT(d.id) FROM details d
      ) <= COUNT(sd.id_d);
```

Результат выполнения запроса на рисунке 37.

```
SQL> SELECT sd.id_s
CON> FROM supplier_detail sd
CON> GROUP BY sd.id_s
CON> HAVING (
CON>         SELECT COUNT(d.id) FROM details d
CON> ) <= COUNT(sd.id_d);
SQL>
```

Рисунок 37 — Результат выполнения запроса задачи 37

Задача 38: вывести номера только тех деталей, которые не поставляет поставщик 1.

Запрос:

```
SELECT DISTINCT id_d
    FROM supplier_detail
    WHERE id_s != 1;
```

Результат выполнения запроса на рисунке 38.

```
SQL> SELECT DISTINCT id_d
CON>     FROM supplier_detail
CON>     WHERE id_s != 1;

        ID_D
============
           1
           2
           3
           4
           5
           8
          11
          12
```

Рисунок 38 — Результат выполнения запроса задачи 38

Задача 39: вывести номера деталей, которые имеют вес больше 34 или поставляются поставщиком 3.

Запрос:

```
SELECT DISTINCT d.id
      FROM details d
```

```
    INNER JOIN supplier_detail sd
    ON sd.id_d = d.id
WHERE d.weight > 34
    OR sd.id_s = 3
ORDER BY d.id DESC;
```

Результат выполнения запроса на рисунке 39.

```
SQL> SELECT DISTINCT d.id
CON> FROM details d
CON>     INNER JOIN supplier_detail sd
CON>     ON sd.id_d = d.id
CON> WHERE d.weight > 34
CON>     OR sd.id_s = 3
CON> ORDER BY d.id DESC;

          ID
============
          12
          11
           8
           5
           4
           3
           2
           1
```

Рисунок 39 — Результат выполнения запроса задачи 39

Задача 40: вывести названия деталей, которые поставляет поставщик с номером 2.

Запрос:

```
SELECT d.name
      FROM details d
        INNER JOIN supplier_detail sd
        ON sd.id_d = d.id
      WHERE sd.id_s = 2
      ORDER BY d.id ASC;
```

Результат выполнения запроса на рисунке 40.

```
SQL> SELECT d.name
CON> FROM details d
CON>         INNER JOIN supplier_detail sd
CON>         ON sd.id_d = d.id
CON> WHERE sd.id_s = 2
CON> ORDER BY d.id ASC;

NAME
================================================================================
Болт
Гвоздь
Га*ка малая
Винт
Петля
Болт
```

Рисунок 40 — Результат выполнения запроса задачи 40

Задача 41: вывести номера поставщиков с состоянием меньшим, чем текущее максимальное состояние в таблице поставщиков.

Запрос:

```
SELECT id
    FROM suppliers
    WHERE money < (
        SELECT MAX(money)
            FROM suppliers
    );
```

Результат выполнения запроса на рисунке 41.

```
SQL> SELECT id
CON>     FROM suppliers
CON>     WHERE money < (
CON>         SELECT MAX(money)
CON>             FROM suppliers
CON>     );

          ID
============
           1
           2
           3
           4
           5
           7
           8
         123
```

Рисунок 41 — Результат выполнения запроса задачи 41

Задача 42: выдать номера, состояние и город для всех поставщиков, у которых состояние равно или больше среднего по городу.

Запрос:

```
SELECT s1.id, s1.fio, s1.city
    FROM suppliers s1
    WHERE s1.money >= (
        SELECT AVG(s2.money)
            FROM suppliers s2
            WHERE s2.city = s1.city
    );
```

Результат выполнения запроса на рисунке 42.

```
SQL> SELECT s1.id, s1.fio, s1.city
CON>     FROM suppliers s1
CON>     WHERE s1.money >= (
CON>         SELECT AVG(s2.money)
CON>             FROM suppliers s2
CON>             WHERE s2.city = s1.city
CON>     );

          ID FIO                                                                              CITY
============ ================================================================================ ========
           2 Петров П.П.                                                                       Москва
           3 Сидоров С.С.                                                                      Париж
           4 Васильев В.В.                                                                     Лондон
           5 Семенов С.С.                                                                      Киев
           6 К. Ласки                                                                             Нью-Йорк
         123 Ульянов В.И.                                                                      Севастополь
```

Рисунок 42 — Результат выполнения запроса задачи 42

Задача 43: для каждой поставляемой детали получить её номер и общий объём поставок. Сохранить результат в новой таблице.

Запрос:

```
SELECT d.id, MAX(sd.quantity)
    FROM details d
        LEFT JOIN supplier_detail sd
        ON sd.id_D = d.id
    GROUP BY d.id;
```

Результат выполнения запроса на рисунке 43.

```
SQL> SELECT d.id, MAX(sd.quantity)
CON>     FROM details d
CON>         LEFT JOIN supplier_detail sd
CON>         ON sd.id_D = d.id
CON>     GROUP BY d.id;

          ID          MAX
============ ============
           1          621
           2          267
           3          120
           4          190
           5          621
           6       <null>
           7       <null>
           8          200
           9       <null>
          10       <null>
          11          590
          12          926
```

Рисунок 43 — Результат выполнения запроса задачи 43

Задача 44: вывести имена поставщиков и названия деталей, которые поставляются в наибольшем объеме.

Запрос:

```
SELECT s.fio, d.name
    FROM suppliers s
         INNER JOIN supplier_detail sd_outer
         ON sd_outer.id_s = s.id
            INNER JOIN details d
                ON sd_outer.id_d = d.id
    WHERE sd_outer.quantity = (
        SELECT MAX(sd_inner.quantity)
            FROM supplier_detail sd_inner
            WHERE sd_inner.id_d = d.id
    );
```

Результат выполнения запроса на рисунке 44.

```
SQL> SELECT s.fio, d.name
CON>     FROM suppliers s
CON>          INNER JOIN supplier_detail sd_outer
CON>          ON sd_outer.id_s = s.id
CON>             INNER JOIN details d
CON>                 ON sd_outer.id_d = d.id
CON>     WHERE sd_outer.quantity = (
CON>         SELECT MAX(sd_inner.quantity)
CON>             FROM supplier_detail sd_inner
CON>             WHERE sd_inner.id_d = d.id
CON>     );

FIO                                                                              NAME
================================================================================ ===========
Петров П.П.                                                                     Га*ка малая
Петров П.П.                                                                     Винт
Петров П.П.                                                                     Болт
К. Ласки                                                                          Болт
Дзержинский Ф.Э.                                                              Вал
Ульянов В.И.                                                                    Винт
Ульянов В.И.                                                                    Петля
Ульянов В.И.                                                                    Гвоздь
```

Рисунок 44 — Результат выполнения запроса задачи 44

2.2 Часть 2

Задача 1: выдать № белых деталей, поставляемых поставщиками Севастополя и Москвы, с фамилиями, начинающимися на букву «П».

Запросы на языке SQL представлены в таблице ниже, а результат выполнения запроса — на рисунке 45.

С использованием JOIN	Без JOIN

`SELECT d.id`
`    FROM suppliers s`
`        JOIN supplier_detail sd`
`        ON sd.id_s = s.id`
`        JOIN details d`
`        ON d.id = sd.id_d`
`    WHERE   (d.color = 'ffffff')`
`        AND (d.city IN ('Севастополь', 'Москва'))`
`        AND (s.fio LIKE '%П%');`
	`SELECT d.id`
	`    FROM suppliers s, supplier_detail sd, details d`
	`    WHERE s.id = sd.id_s AND d.id = sd.id_d`
	`        AND (d.color = 'ffffff')`
	`        AND (d.city IN ('Севастополь', 'Москва'))`
	`        AND (s.fio LIKE '%П%');`

```
                ID
============
                12
```

Рисунок 45 — Результат выполнения запроса задачи 1 части 2

Задача 2: выдать номера всех поставляемых деталей поставщиками Москвы при объеме поставок до 12 300.

Запросы на языке SQL представлены в таблице ниже, а результат выполнения запроса — на рисунке 46.

С использованием JOIN	Без JOIN

`SELECT DISTINCT sd.id_d`
`FROM suppliers s`
`JOIN supplier_detail sd`
`ON sd.id_s = s.id`
`WHERE s.city = 'Москва' AND (`
`SELECT SUM(sd_inner.quantity)`
`FROM supplier_detail sd_inner`
`WHERE sd.id_d = sd_inner.id_d`
`) <= 12300;`
	`SELECT DISTINCT sd.id_d`
	`FROM suppliers s, supplier_detail sd`
	`WHERE sd.id_s = s.id`
	`AND s.city = 'Москва' AND (`
	`SELECT SUM(sd_inner.quantity)`
	`FROM supplier_detail sd_inner`
	`WHERE sd.id_d = sd_inner.id_d`
	`) <= 12300;`

```
SQL> SELECT DISTINCT sd.id_d
CON>     FROM suppliers s
CON>         JOIN supplier_detail sd
CON>         ON sd.id_s = s.id
CON>     WHERE s.city = 'Москва' AND (
CON>         SELECT SUM(sd_inner.quantity)
CON>             FROM supplier_detail sd_inner
CON>             WHERE sd.id_d = sd_inner.id_d
CON>     ) <= 12300;

        ID_D
============
           1
           2
           3
           4
           5
           8
          12
```

Рисунок 46 — Результат выполнения запроса задачи 2 части 2

Задача 3: выдать фамилии поставщиков, находящихся в Париже, поставляющих зеленые детали, в названии которых есть символ «*».

Запросы на языке SQL представлены в таблице ниже, а результат выполнения запроса — на рисунке 47.

С использованием JOIN	Без JOIN
SELECT s.fio FROM details d JOIN supplier_detail sd ON sd.id_d = d.id JOIN suppliers s ON s.id = sd.id_s WHERE d.name LIKE '%*%' AND s.city = 'Париж';	SELECT s.fio FROM details d, supplier_detail sd, suppliers s WHERE d.id = sd.id_d AND s.id = sd.id_s AND d.name LIKE '%*%' AND s.city = 'Париж';

```
SQL> SELECT s.fio
CON>     FROM details d
CON>         JOIN supplier_detail sd
CON>         ON sd.id_d = d.id
CON>         JOIN suppliers s
CON>         ON s.id = sd.id_s
CON>     WHERE   d.name LIKE '%*%'
CON>         AND s.city = 'Париж';

FIO
====================================
Сидоров С.С.
```

Рисунок 47 — Результат выполнения запроса задачи 3 части 2

Задача 4: выдать количество деталей, поставляемых поставщиком №123 в порядке возрастания количества деталей.

Запросы на языке SQL представлены в таблице ниже, а результат выполнения запроса — на рисунке 48.

С использованием JOIN	Без JOIN
SELECT sd.id_d, SUM(sd.quantity) FROM suppliers s JOIN supplier_detail sd ON sd.id_s = s.id WHERE s.id = 123 GROUP BY sd.id_d ORDER BY SUM(sd.quantity) ASC;	SELECT sd.id_d, SUM(sd.quantity) FROM suppliers s, supplier_detail sd WHERE sd.id_s = s.id AND s.id = 123 GROUP BY sd.id_d ORDER BY SUM(sd.quantity) ASC;

```
SQL> SELECT sd.id_d, SUM(sd.quantity)
CON>     FROM suppliers s
CON>         JOIN supplier_detail sd
CON>         ON sd.id_s = s.id
CON>     WHERE s.id = 123
CON>     GROUP BY sd.id_d
CON>     ORDER BY SUM(sd.quantity) ASC;

        ID_D                  SUM
============ ====================
           2                  267
           1                  409
          11                  590
           5                  621
```

Рисунок 48 — Результат выполнения запроса задачи 4 части 2

2.3 Часть 3

Была составлена ER-диаграмма, представленная на рисунке 49.

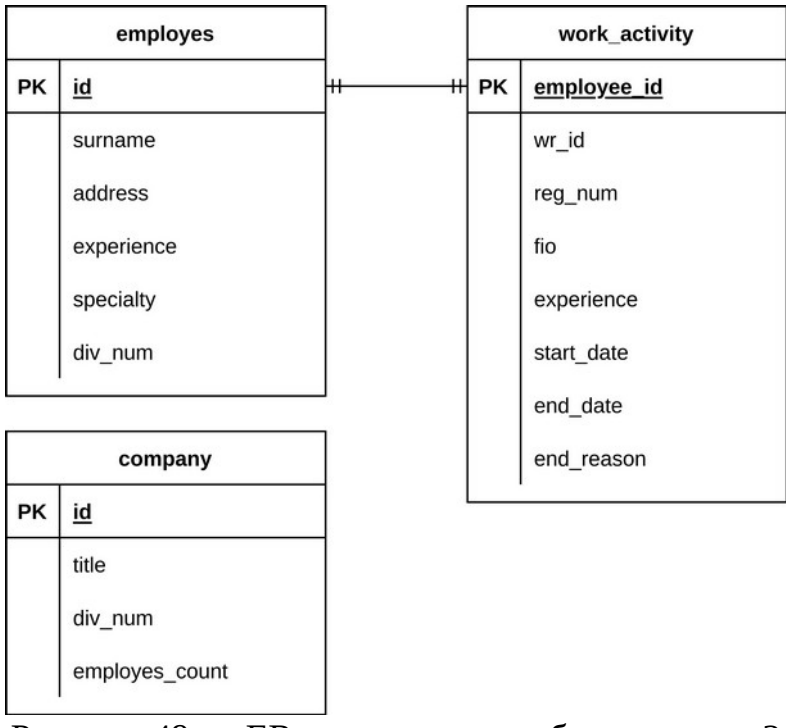


Рисунок 49 — ER-диаграмма для базы данных 2

Перед началом выполнения поставленных задач были сформированы запросы на создание требуемых таблиц:

```
CREATE TABLE employes (
    id INTEGER NOT NULL,
    surname VARCHAR(20) NOT NULL,
```

```
    address VARCHAR(20) NOT NULL,
    experience INTEGER NOT NULL,
    specialty VARCHAR(10) NOT NULL,
    div_num INTEGER NOT NULL,
    PRIMARY KEY (id)
);
CREATE TABLE work_activity (
    employee_id INTEGER NOT NULL,
    wr_id INTEGER NOT NULL,
    reg_num INTEGER NOT NULL,
    fio VARCHAR(20) NOT NULL,
    experience INTEGER NOT NULL,
    start_date DATE NOT NULL,
    end_date DATE,
    end_reason VARCHAR(10),

    PRIMARY KEY (employee_id),
    UNIQUE (wr_id),
    FOREIGN KEY (employee_id) REFERENCES employes(id)
);
CREATE TABLE company (
    id INTEGER NOT NULL,
    title VARCHAR(10) NOT NULL,
    div_num INTEGER NOT NULL,
    employes_count INTEGER NOT NULL,

    PRIMARY KEY (id)
);
```

После таблицы были заполненны следующими данными:

```
INSERT INTO employes VALUES (1, 'Михайлов',  'ул. Первая',    4, 'рыболов', 9);
INSERT INTO employes VALUES (2, 'Руднев',    'ул. Вторая',    3, 'слесарь', 5);
INSERT INTO employes VALUES (3, 'Копащенко', 'ул. Третья',    2, 'токарь',  9);
INSERT INTO employes VALUES (4, 'Ализ',      'ул. Четвёртая', 1, 'лесоруб', 1);
INSERT INTO work_activity VALUES (1, 1500, 1515, 'Михайлов', 4, '1982-01-01',
'1983-01-01', 'слишкомтру');
INSERT INTO work_activity VALUES (3, 1501, 1525, 'Михайлов', 2, '1982-01-01',
'1984-01-01', 'алкоголь');
INSERT INTO company VALUES (789, 'Оз',         5, 15);
INSERT INTO company VALUES (123, 'Морфеус',    9, 20);
INSERT INTO company VALUES (456, 'РосСельМаш', 1, 10);
```

Задача 1: выдать список служащих, уволенных и не проработавших по году на предприятии с №123.

Составленный запрос:

```
SELECT e.id, e.surname
    FROM employes e
        JOIN work_activity wa
        ON wa.employee_id = e.id
        JOIN company c
        ON c.div_num = e.div_num
    WHERE wa.end_reason IS NOT NULL
        AND wa.end_date - wa.start_date <= 366
        AND c.id = 123;
```

Результат выполнения запроса представлен на рисунке 50.

```
SQL> SELECT e.id, e.surname
CON>     FROM employes e
CON>         JOIN work_activity wa
CON>         ON wa.employee_id = e.id
CON>         JOIN company c
CON>         ON c.div_num = e.div_num
CON>     WHERE wa.end_reason IS NOT NULL
CON>         AND wa.end_date - wa.start_date <= 366
CON>         AND c.id = 123;

          ID SURNAME
============ ========================================
           1 Михайлов
```

Рисунок 50 — Результат выполнения запроса задачи 1 части 3

Задача 2: в таблицу предприятие добавить столбец «№ счета в банке» и найти предприятие с максимальным числом служащих.

Составленная транзакция:

```
ALTER TABLE company
    ADD bank_account VARCHAR(20) DEFAULT '0000-0000-0000-0000';
SELECT title
    FROM company
    WHERE employes_count = (SELECT MAX(employes_count) FROM company);
```

Результаты выполнения запросов представлены на рисунках 51 и 52.

```
SQL> ALTER TABLE company
CON>     ADD bank_account VARCHAR(20) DEFAULT '0000-0000-0000-0000';
SQL> SHOW TABLE company;
ID                              INTEGER Not Null
TITLE                           VARCHAR(10) Not Null
DIV_NUM                         INTEGER Not Null
EMPLOYES_COUNT                  INTEGER Not Null
BANK_ACCOUNT                    VARCHAR(20) Nullable DEFAULT '0000-0000-0000-0000'
CONSTRAINT INTEG_37:
  Primary key (ID)
SQL>
```

Рисунок 51 — Добавление атрибута в таблицу

```
SQL> SELECT title
CON>     FROM company
CON>     WHERE employes_count = (
CON>         SELECT MAX(employes_count)
CON>             FROM company
CON>     );

TITLE
========================================
Морфеус
```

Рисунок 52 — Выполнение запроса по заданию