



Министерство науки и высшего образования Российской Федерации  
Федеральное государственное бюджетное образовательное учреждение  
высшего образования  
«Московский государственный технический университет  
имени Н. Э. Баумана  
(национальный исследовательский университет)»  
(МГТУ им. Н. Э. Баумана)

ФАКУЛЬТЕТ \_\_\_\_\_ Фундаментальные науки  
КАФЕДРА \_\_\_\_\_ Математическое моделирование

## Лабораторная работа № 1

*по курсу:*

*“Методы и алгоритмы обработки больших  
даннных”*

Студент \_\_\_\_\_  
ФН12-21М  
(Группа)

\_\_\_\_\_  
(Подпись, дата)

А. А. Кононенко  
(И. О. Фамилия)

Руководитель научной работы  
\_\_\_\_\_

\_\_\_\_\_  
(Подпись, дата)

М. С. Виноградова  
(И. О. Фамилия)

**Оглавление**

<b>Введение</b> . . . . .	3
---------------------------	---

## Введение

Написать *SQL*-запросы, требуемые в задачах из презентаций семинарских занятий. При создании запросов воспользуемся СУБД *MySQL*.

```
-- Запрос на создание базы данных
CREATE DATABASE employee;
```

Статистика 1

CREATE DATABASE employee | Введите SQL выражение

Name	Value
Updated Rows	1
Query	-- Запрос на создание базы данных CREATE DATABASE employee
Finish time	Wed Apr 17 07:43:20 GMT+03:00 2024

Рис. 1.

```
-- Выбор созданной базы данных для дальнейших операций
USE employee;
```

Статистика 1

USE employee | Введите SQL выражение чтобы отфильтровать результат

Name	Value
Updated Rows	0
Query	-- Выбор созданной базы данных для дальнейших операций USE employee
Finish time	Wed Apr 17 07:43:44 GMT+03:00 2024

Рис. 2.

```
-- Создание таблицы department
CREATE TABLE department (
    departmentID INT NOT NULL AUTO_INCREMENT PRIMARY KEY,
    name VARCHAR(30)
) ENGINE=InnoDB;
```

Статистика 1

CREATE TABLE department ( departmentID INT NOT NULL AUTO\_INCREMENT PRIMARY KEY, name VARCHAR(30) ) ENGINE=InnoDB | Введите SQL выражение

Name	Value
Updated Rows	0
Query	-- Создание таблицы department CREATE TABLE department ( departmentID INT NOT NULL AUTO_INCREMENT PRIMARY KEY, name VARCHAR(30) ) ENGINE=InnoDB
Finish time	Wed Apr 17 07:44:11 GMT+03:00 2024

Рис. 3.

-- эта таблица аналогичная таблице employee, мы её потом удаляем

```
create table employee1
(employeeID int not null auto_increment primary key,
name1 varchar (80),
job varchar(30),
departmentID int not null
references department(departmentID)
) ENGINE=InnoDB;
```

Статистика 1

create table employee1 (employeeID int not null auto\_increment primary key | Введите SQL выражение

Name	Value
Updated Rows	0
Query	-- эта таблица аналогична таблице employee, мы её потом удаляем create table employee1 (employeeID int not null auto_increment primary key, name1 varchar (80), job varchar(30), departmentID int not null references department(departmentID) ) ENGINE=InnoDB
Finish time	Wed Apr 17 07:44:42 GMT+03:00 2024

Рис. 4.

-- эта таблица аналогична таблице employee1, мы её потом не удаляем

```
create table employee
(employeeID int not null auto_increment primary key,
name1 varchar (80),
job varchar(30),
departmentID int not null,
CONSTRAINT `DEP` FOREIGN KEY (departmentID)
REFERENCES department(departmentID)
ON DELETE CASCADE ON UPDATE CASCADE
) ENGINE=InnoDB;
```

Статистика 1

create table employee1 (employeeID int not null auto\_increment primary key | Введите SQL выражение

Name	Value
Updated Rows	0
Query	-- эта таблица аналогична таблице employee, мы её потом удаляем create table employee1 (employeeID int not null auto_increment primary key, name1 varchar (80), job varchar(30), departmentID int not null references department(departmentID) ) ENGINE=InnoDB
Finish time	Wed Apr 17 07:44:42 GMT+03:00 2024

Рис. 5.

-- занимаемся добавлением элементов в созданные таблицы

```
INSERT INTO department (departmentID, name)
VALUES
(1, 'Dep_analit'),
(2, 'Dep_prog'),
(3, 'Dep_admin');
```

Статистика 1

INSERT INTO department (departmentID, name) VALUES (1, 'Dep\_analit'), ( | Введите SQL выражение

Name	Value
Updated Rows	3
Query	-- занимаемся добавлением элементов в созданные таблицы INSERT INTO department (departmentID, name) VALUES (1, 'Dep_analit'), (2, 'Dep_prog'), (3, 'Dep_admin')
Finish time	Wed Apr 17 07:45:39 GMT+03:00 2024

Рис. 6.

Рис. 7.

```

    SELECT
    VALUES
    (100, 'Smit N', 'Programmer', 2),
    (101, 'Stone J.', 'manager', 3),
    (102, 'Asser M.', 'analitic', 1),
    (103, 'Wood N.', 'Programmer', 2),
    (104, 'Thomson L.', 'Programmer', 2);
  
```

Статистика 1

INSERT INTO employee (employeeID, name1, job, departmentID) VALUES (100, 'Smit N', 'Programmer', 2), (101, 'Stone J.', 'manager', 3), (102, 'Asser M.', 'analitic', 1), (103, 'Wood N.', 'Programmer', 2), (104, 'Thomson L.', 'Programmer', 2);
Name Value
Updated Rows 5
Query
Finish time Wed Apr 17 07:48:00 GMT+03:00 2024

Рис. 8.

```

    SELECT
    VALUES
    (100, 'Smit N', 'Programmer', 2),
    (101, 'Stone J.', 'manager', 3),
    (102, 'Asser M.', 'analitic', 1),
    (103, 'Wood N.', 'Programmer', 2),
    (104, 'Thomson L.', 'Programmer', 2);
  
```

Статистика 1

INSERT INTO employee1 (employeeID, name1, job, departmentID) VALUES (100, 'Smit N', 'Programmer', 2), (101, 'Stone J.', 'manager', 3), (102, 'Asser M.', 'analitic', 1), (103, 'Wood N.', 'Programmer', 2), (104, 'Thomson L.', 'Programmer', 2);
Name Value
Updated Rows 5
Query
Finish time Wed Apr 17 07:46:16 GMT+03:00 2024

Рис. 9.

```

    -- Создание таблицы employeeSkills
    CREATE TABLE employeeSkills (
        employeeID int not null
        references employee(employeeID),
        skill VARCHAR(15),
        PRIMARY KEY (employeeID, skill)
    ) ENGINE=InnoDB;
  
```

Статистика 1

CREATE TABLE employeeSkills ( employeeID int not null
Name Value
Updated Rows 0
Query -- Создание таблицы employeeSkills
CREATE TABLE employeeSkills ( employeeID int not null
references employee(employeeID),
skill VARCHAR(15),
PRIMARY KEY (employeeID, skill)
) ENGINE=InnoDB

-- Создание таблицы client

```
CREATE TABLE client (
    clientID int not null auto_increment primary key,
    name VARCHAR(40),
    address VARCHAR(100),
    contactPerson VARCHAR(80),
    contactNumber VARCHAR(80)
) ENGINE=InnoDB;
```

Статистика 1

CREATE TABLE client ( clientID int not null auto\_increment primary key, na |

Name	Value
Updated Rows	0
Query	-- Создание таблицы client CREATE TABLE client ( clientID int not null auto_increment primary key, name VARCHAR(40), address VARCHAR(100), contactPerson VARCHAR(80), contactNumber VARCHAR(80) ) ENGINE=InnoDB

Finish time Wed Apr 17 07:53:16 GMT+03:00 2024

Рис. 10.

-- Создание таблицы assignment

```
CREATE TABLE assignment (
    clientID int not null
        references client(clientID),
    employeeID int not null
        references employee(employeeID),
    workdate date,
    hours float,
    PRIMARY KEY (clientID, employeeID, workdate)
) ENGINE=InnoDB;
```

Статистика 1

CREATE TABLE assignment ( clientID int not null references client(clientID), |

Name	Value
Updated Rows	0
Query	-- Создание таблицы assignment CREATE TABLE assignment ( clientID int not null references client(clientID), employeeID int not null references employee(employeeID), workdate date, hours float, PRIMARY KEY (clientID, employeeID, workdate) ) ENGINE=InnoDB

Finish time Wed Apr 17 07:53:42 GMT+03:00 2024

Рис. 11.

INSERT INTO employeeSkills (employeeID, skill)

```
VALUES
    (101 , 'Basic'),
    (103 , 'Python'),
    (102 , 'SQL'),
    (100 , 'C++'),
    (100 , 'Pascal'),
    (104 , 'Delphi');
```

Статистика 1

INSERT INTO employeeSkills (employeeID, skill) VALUES (101 , 'Basic'), (103 |

Name	Value
Updated Rows	6
Query	INSERT INTO employeeSkills (employeeID, skill) VALUES (101 , 'Basic'), (103, 'Python'), (102, 'SQL'), (100, 'C++'), (100, 'Pascal'), (104, 'Delphi')

Finish time Wed Apr 17 07:54:19 GMT+03:00 2024

Рис. 12.

The screenshot shows the SQL Server Management Studio interface. On the left, there's a tree view with 'Output' selected. In the main pane, a query window displays:

```
INSERT INTO client (clientID, name, adress, contactPerson, contactNumber)
VALUES
    (1100, 'ACER', 'M. 12 st.', 'Nora', 112233445566),
    (1101, 'MTS', 'S.P.11 st', 'Lena', 665544332211),
    (1102, 'Dog', 'N.N 13 st.', 'Ivan', 123456123456),
    (1103, 'Cat', 'K. 14 st.', 'Petr', 654321123456);
```

Below the query window is a 'Статистика 1' (Statistics 1) window showing the execution plan. At the bottom, a status bar indicates:

Введите SQL выражение чтобы...  
Finish time Wed Apr 17 07:54:37 GMT+03:00 2024

Рис. 13.

The screenshot shows the SQL Server Management Studio interface. On the left, there's a tree view with 'Output' selected. In the main pane, a query window displays:

```
INSERT INTO assignment (clientID, employeeID, workdate, hours)
VALUES
    (1100, 100, '2009-01-10', 120),
    (1101, 101, '2008-11-01', 10),
    (1102, 102, '2009-12-10.', 70),
    (1103, 102, '2009-02-01', 100);
```

Below the query window is a 'Статистика 1' (Statistics 1) window showing the execution plan. At the bottom, a status bar indicates:

Введите SQL выражение чтобы...  
Finish time Wed Apr 17 07:54:54 GMT+03:00 2024

Рис. 14.

The screenshot shows the SQL Server Management Studio interface. On the left, there's a tree view with 'Output' selected. In the main pane, a query window displays:

```
SELECT * FROM department;
```

Below the query window is a 'department 1' (department 1) results grid showing:

departmentID	name
1	Dep_analit
2	Dep_prog
3	Dep_admin

Рис. 15.

```
-- Получить имена всех клиентов и имена сотрудников для
-- контакта из отношения client
SELECT name, contactPerson
FROM client WHERE contactNumber='665544332211';

client 1
SELECT name, contactPerson FROM client WHERE contactNumber='665544332211' | Введите SQL выражение чтобы


|   | name | contactPerson |
|---|------|---------------|
| 1 | MTS  | Lena          |


```

Рис. 16.

```
-- 2 Получить для всех служащих отдела № 2 имя сотрудника и
-- его должность (атрибут job). МГ
select name1, job
from employee Left join department
on employee.departmentID = department.departmentID
where employee.departmentID = 2;

employee 1
select name1, job from employee Left join department on employee.departmentID = 2 | Введите SQL выражение чтобы


|   | name1      | job        |
|---|------------|------------|
| 1 | Smit N     | Programmer |
| 2 | Wood N.    | Programmer |
| 3 | Thomson L. | Programmer |


```

Рис. 17.

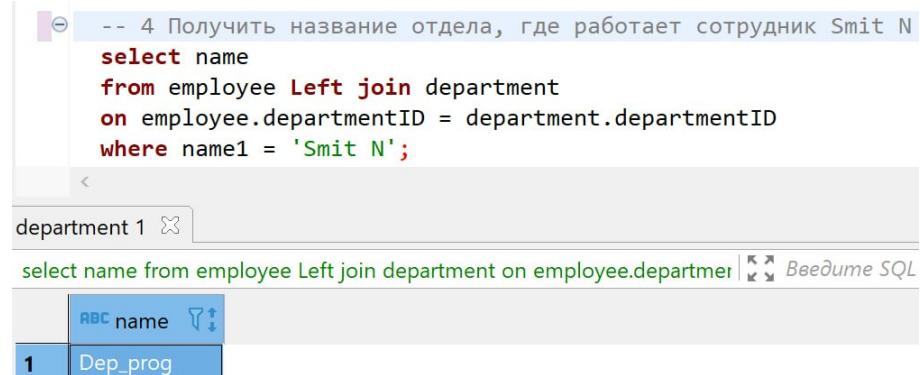
```
-- 3 Получить имя служащего, которому было поручено работать
-- с клиентом 'MTS'.
select contactPerson
from client
where name = 'MTS';

client 1
`select contactPerson from client where name = 'MTS'` | Введите SQL выражение чтобы


|   | contactPerson |
|---|---------------|
| 1 | Lena          |


```

Рис. 18.



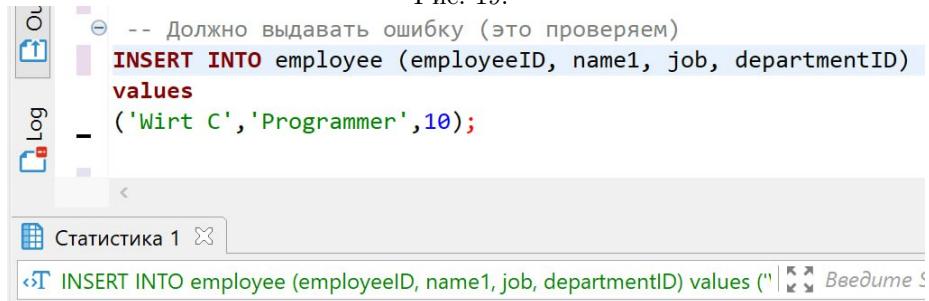
```
-- 4 Получить название отдела, где работает сотрудник Smit N
select name
from employee Left join department
on employee.departmentID = department.departmentID
where name1 = 'Smit N';
```

department 1

	ABC name
1	Dep_prog

select name from employee Left join department on employee.departmentID | Введите SQL

Рис. 19.

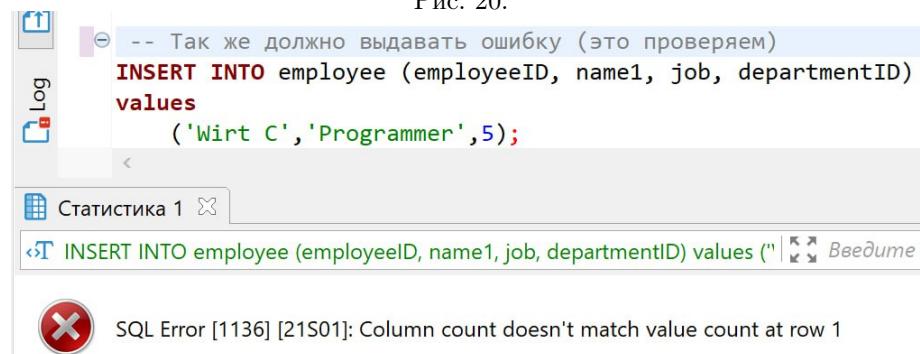


```
-- Должно выдавать ошибку (это проверяем)
INSERT INTO employee (employeeID, name1, job, departmentID)
values
('Wirt C', 'Programmer', 10);
```

Статистика 1

SQL Error [1136] [21S01]: Column count doesn't match value count at row 1

Рис. 20.



```
-- Так же должно выдавать ошибку (это проверяем)
INSERT INTO employee (employeeID, name1, job, departmentID)
values
('Wirt C', 'Programmer', 5);
```

Статистика 1

SQL Error [1136] [21S01]: Column count doesn't match value count at row 1

Рис. 21.

-- Без ошибки (удаляем строку)  
**INSERT INTO** department (departmentID, name)  
**values**  
(10, 'Test DELETE');

Статистика 1

INSERT INTO department (departmentID, name) values (10,'Test DEL')

Name	Value
Updated Rows	1
Query	-- Без ошибки (удаляем строку) <b>INSERT INTO</b> department (departmentID, name) <b>values</b> (10,'Test DELETE')
Finish time	Wed Apr 17 07:49:29 GMT+03:00 2024

Рис. 22.

-- 4 Получить название отдела, где работает сотрудник Smit N  
**select** name  
**from** employee **Left join** department  
**on** employee.departmentID = department.departmentID  
**where** name1 = 'Smit N';

department 1

select name from employee Left join department on employee.departmentID | Введите SQL

ABC name
1 Dep_prog

Рис. 23.

-- 5 Выяснить имена служащих, работающих вместе со Smit N, Smit N  
-- в результат не включать  
**SELECT** e2.name1 **FROM** employee **as** e1, employee **as** e2 **WHERE**  
e1.name1 = 'Smit N' **AND** e1.departmentID = e2.departmentID  
**and** e2.name1 <> 'Smit N';

employee 1

'SELECT e2.name1 FROM employee as e1, employee as e2 WHERE e1.name1 | Введите SQL выражение

ABC name1
1 Wood N.
2 Thomson L.

Рис. 24.

-- 6 Создайте запрос, возвращающий имя служащего и всю информацию о его (или ее) квалификации (skill).

```
select name1, skill
from employee as emp, employeeskills as empsk
where emp.employeeID = empsk.employeeID;
```

employee(+ 1)

select name1, skill from employee as emp, employeeskills as empsk where

	ABC name1	ABC skill
1	Smit N	C++
2	Smit N	Pascal
3	Stone J.	Basic
4	Asser M.	SQL
5	Wood N.	Python
6	Thomson L.	Delphi

Рис. 25.

/\*7

Создайте запрос, возвращающий имя служащего и всю информацию о его (или ее) квалификации (skill).

Создайте запрос, использующий LEFT JOIN и возвращающий список клиентов, для которых не было выделено служащих, работающих с ними.

```
select c.name
from client as c left join
assignment as a on c.clientID = a.clientID
where employeeID is null;
```

client 1

select c.name from client as c left join assignment as a on c.clientID = a.cli

ABC name

Рис. 26.

-- 8 (а) Добавить в таблицу employee столбец AGE (возраст), salary (зарплату), perks (надбавки)

```
-- ALTER TABLE employee ADD COLUMN Age INT, ADD COLUMN
-- salary INT, ADD COLUMN perks INT;
```

alter table employee
add column age int,
add column salary int,
add column perks int;

Статистика 1

alter table employee add column age int, add column salary int, add colur

Name	Value
Updated Rows	0
Query	-- 8 (а) Добавить в таблицу employee столбец AGE (возраст), salary -- (зарплату), perks (надбавки) -- ALTER TABLE employee ADD COLUMN Age INT, ADD COLUMN -- salary INT, ADD COLUMN perks INT; alter table employee add column age int, add column salary int, add column perks int
Finish time	Wed Apr 17 08:05:29 GMT+03:00 2024

Рис. 27.

```

    /* 8(b)
     Заполнить новые столбцы данными (зарплата: 20000 -50000,
     надбавки: 1000 - 5000, возраст 20-45 лет).
     Например:
     UPDATE employee SET Age=25,salary=35000, perks=4000 WHERE
     employeeID=100;
    */
update employee
set age = 25, salary = 35000, perks=4000 WHERE
    employeeID=100;
update employee
set age = 30, salary = 45000, perks=2000 WHERE
    employeeID=101;
update employee
set age = 33, salary = 33000, perks=3600 WHERE
    employeeID=102;
update employee
set age = 48, salary = 30000, perks=3000 WHERE
    employeeID=103;
update employee
set age = 27, salary = 47000, perks=4500 WHERE
    employeeID=104;

```

Рис. 28.

```

    -- 9 Вывести все идентификационные номера и имена сотрудников
    -- возраст которых от 32 до 40 лет (включительно).
    select employeeID, name
    from employee e
    where age between 32 and 40;

```

	employeeID	name
1	102	Asser M.
2	105	Grishin Egor
3	106	Kiselev Artem

Рис. 29.

```

    /* 10 Выбрать имена всех сотрудников, которые не являются
    программистами (любыми, системными, прикладными и т.п.).
    Указание. Использовать NOT LIKE '%прогр%'; */
    select name1
    from employee
    where job not like "%прогр%";

```

	name1
1	Stone J.
2	Asser M.

Рис. 30.

The screenshot shows a MySQL Workbench interface. In the top-left pane, there is a code editor with the following SQL query:

```
/* 11 Выбрать имена всех сотрудников в возрасте 25 лет, которые не являются программистами. */
select name1
from employee
where job not like "%progr%"
and age = 25;
```

In the bottom-right pane, there is a results table titled "employee 1" with one row:

	RBC name1
1	

Рис. 31.

The screenshot shows a MySQL Workbench interface. On the left, there is a sidebar with tabs for "Output" and "Log". The main area contains the following complex SQL query:

```
/* 12 Найти зарплаты и надбавки для каждого программиста и каждого аналитика.
SELECT Job,salary,perks FROM employee WHERE Job='programmer'
OR Job='analitic';
update employee
set salary = salary*1.1, perks = perks*1.05
where job = "programmer";
update employee
set salary = salary*1.1, perks = perks*1.05
where job = "analitic";
select salary, perks from employee
where job = "programmer" or job = "analitic";
select name1, salary + perks from employee
where job = "programmer";
```

Рис. 32.

The screenshot shows a MySQL Workbench interface. The main area contains the following SQL query:

```
/* 13 а) Вычислить средний возраст сотрудников компании. Вычислить среднюю зарплату.*/
select avg(age), avg(salary) from employee;
```

In the bottom-right pane, there is a results table titled "Результат 1" with two rows:

	avg(age)	avg(salary)
1	33,7143	42 885,7143

Рис. 33.

```
/* 14 Сколько компания тратит на зарплату сотрудников по подразделениям */
select departmentID, sum(salary)
from employee group by departmentID;
```

employee 1

	departmentID	sum(salary)
1	1	82 000
2	2	173 200
3	3	45 000

Рис. 34.

```
/* 15 Подсчитать число сотрудников имеющих определенную должность*/
select count(*) as "number"
from employee
where job = "programmer";
```

Результат 1

	number
1	3

Рис. 35.

```
/* 16 Вывести сумму всех возрастов сотрудников, работающих в компании*/
select sum(age)
from employee;
```

Результат 1

	sum(age)
1	236

Рис. 36.

/\* 17 Вычислите сумму зарплат и средний возраст сотрудников, которые занимают должность "программист".\*/

```
select sum(salary), avg(age)
from employee
where job = "programmer";
```

Результат 1

	sum(salary)	avg(age)
1	123 200	33,3333

Рис. 37.

/\* 18 Что делает следующий оператор?

```
SELECT (SUM(perks)/SUM(salary) * 100) FROM employee; */
select (sum(perks)/sum(salary)*100)
from employee ;
```

Результат 1

	(sum(perks)/sum(salary)*100)
1	8,8857

Рис. 38.

/\* 19 Подсчитайте количество сотрудников в группах одного возраста.\*/

```
select age, count(*) as "number"
from employee
group by age;
```

employee 1

select age, count(\*) as "number" from employee group by age

	age	number
1	25	1
2	30	1
3	33	1
4	48	1
5	27	1
6	35	1
7	38	1

Рис. 39.

```

    /* 20 Найдите средний возраст сотрудников в различных
подразделениях (должностях).*/
select departmentID, avg(age)
from employee
group by departmentID;
<

employee 1
| select departmentID, avg(age) from employee group by departmentID | Введи
| 1 departmentID 1 avg(age) |
| 1 35,5 |
| 2 33,75 |
| 3 30 |

```

Рис. 40.

```

    /* 21 Подсчитайте средний возраст сотрудников по должностям с
использованием псевдонима столбца, отсортируйте по
возрасту.
Задать псевдоним для столбца, содержащего среднее
значение возраста надо, чтобы его можно было сортировать*/
select job, avg(age) as avg_age
from employee
group by job
order by avg_age;
<

employee 1
| select job, avg(age) as avg_age from employee group by job order by avg | Введи
| RBC job 1 avg_age |
| 1 manager 30 |
| 2 analitic 33 |
| 3 Programmer 33,3333 |
| 4 syst.programmer 35 |
| 5 admin.Programmer 38 |

```

Рис. 41.

```

    /* 22 Разница между приведенными ниже запросами 1 и 2
заключается в том, что
а) никакой разницы нет;
б) они возвращают разные данные;
в) они возвращают одни и те же данные, но LEFT JOIN (1), скорее
всего, будет выполняться быстрее;
г) они возвращают одни и те же данные, но подзапрос (2),
скорее всего, будет выполняться быстрее.
Запрос 1:
SELECT employee.name , e.employeeID FROM employee
left join assignment
on employee.employeeID = assignment.employeeID
WHERE clientID is null;
Запрос 2:
SELECT e.name, e.employeeID FROM employee e
WHERE not exists
(SELECT *FROM assignment WHERE employeeID = e.employeeID);*/

-- 1 запрос
select employee.name1, employee.employeeID
from employee left join assignment
on employee.employeeID = assignment.employeeID
where clientID is null;
<

employee 1
| select employee.name1, employee.employeeID from employee left join as | Введи
| RBC name1 1 employeeID |
| 1 Wood N. 103 |
| 2 Thomson L. 104 |
| 3 Grishin Egor 105 |
| 4 Kiselev Artem 106 |

```

Рис. 42.

```
-- 2 запрос
select e.name1, e.employeeID
from employee e
where not exists
(select * from assignment
where employeeID = e.employeeID);
```

employee 1

	ABC name1	employeeID
1	Wood N.	103
2	Thomson L.	104
3	Grishin Egor	105
4	Kiselev Artem	106

Рис. 43.

```
/* 23 Создайте запрос, возвращающий имя служащего и всю
информацию о его (или ее) квалификации (skill)*/
select e.name1, e2.skill
from employee e, employeeskills e2
where e.employeeID = e2.employeeID;
```

employee(+) 1

	ABC name1	skill
1	Smit N	C++
2	Smit N	Pascal
3	Stone J.	Basic
4	Asser M.	SQL
5	Wood N.	Python
6	Thomson L.	Delphi

Рис. 44.

```
/* 24 Создайте запрос, использующий ключевое слово EXISTS и
возвращающий список клиентов, для которых не было
выделено служащих, работающих с ними.*/
select c.name
from client as c
where not exists
( select a.employeeID
from assignment a
where a.clientID = c.clientID);
```

client 1

	ABC name
--	----------

Рис. 45.

```

/* 25 Создайте запрос возвращающий имя, возраст и должность
самого старого служащего, с использованием группирующей
функции MAX в подзапросе*/
select name1 , age , job
from employee
where age = (select max(age) from employee);


```

employee 1

	RBC name1	123 age	ABC job
1	Wood N.	48	Programmer

Рис. 46.

```

/* 26 Снять надбавки со служащих, которые не имели внешних
заданий (использовать UPDATE и SELECT). */
update employee
set perks = 0
where not exists
(select name1 where employeeID in
(select employeeID from assignment));


```

Статистика 1

Name	Value
Updated Rows	4

Query

```

/* 26 Снять надбавки со служащих, которые не имели внешних
заданий (использовать UPDATE и SELECT). */
update employee
set perks = 0
where not exists
(select name1 where employeeID in
(select employeeID from assignment))

```

Finish time

Рис. 47.

```

/* 27 Добавить в таблицу department столбец ManagerID (номер
руководителя отдела). */
alter table department
add column
ManagerID int not null references employee(employeeID);
update department
set ManagerID = 111
where departmentID in (1, 2);
update department
set ManagerID = 101 where departmentID = 3;


```

Статистика 1

Name	Value
Updated Rows	0

Query

```

/* 27 Добавить в таблицу department столбец ManagerID (номер
руководителя отдела). */
alter table department
add column
ManagerID int not null references employee(employeeID)

```

Finish time

Рис. 48.

```

/* 28 Найти номера, имена, номера отделов и имена руководителей
отделов служащих, размер заработной платы которых меньше
25000 руб.*/
select e.employeeID, e.name1, d.departmentID, e2.name1
from employee e, department d, employee e2
where e.departmentID = d.departmentID and
d.departmentID = e2.departmentID and
e.salary < 25000;

```

employee(+) 1

employeeID	name1	departmentID	name1

Рис. 49.

```

/* 29 Найти номера отделов и имена сотрудников, размер заработной
платы которых превышает размер заработной платы
руководителя отдела */
select e.departmentID, e.name1
from employee e, department d, employee e2
where e.departmentID = d.departmentID and
d.ManagerID = e2.employeeID and
e.salary > e2.salary;

```

employee 1

departmentID	name1

Рис. 50.

```

/* 31 Найти номера отделов, все сотрудники которых имеют разный
возраст, т.е возраст каждого сотрудника уникален
select e.departmentID */
select e.departmentID
from employee as e,
(select departmentID, count(*) as emp_n
from employee group by departmentID) as empln,
(select e2.departmentID, count(e2.age) as un_n
from employee as e2
where not exists
(select e2.employeeID
from employee as e3
where e3.age = e2.age and
e3.employeeID != e2.employeeID)
group by e2.departmentID) as un
where e.departmentID = empln.departmentID and
empln.departmentID = un.departmentID and
empln.emp_n = un_n
group by e.departmentID;

```

employee 1

departmentID
1
2
3

Рис. 51.

```

/* 32 Найти номера отделов, сотрудников которых можно различить по
имени и возрасту, т.е каждый сотрудник имеет уникальное имя и
возраст. */
select e.departmentID
from employee as e,
(select departmentID, count(*) as emp_n
from employee group by departmentID) as empln,
(select e2.departmentID, count(e2.name1) as un_n
from employee as e2
where not exists
(select e2.employeeID
from employee as e3
where e3.age = e2.age and
e3.employeeID != e2.employeeID)
group by e2.departmentID) as un
where e.departmentID = empln.departmentID and
empln.departmentID = un.departmentID and
empln.emp_n = un_n
group by e.departmentID;

```

employee 1

select e.departmentID from empln | Введите SQL выражение чтобы отфильтровать результат

	departmentID
1	1
2	2
3	3

Рис. 52.

```

/* 33. Найти номера и имена сотрудников-однофамильцев (в пределах пред-
приятия. */
select e.employeeID, e.name1
from employee e
where e.name1 = some
(select e2.name1
from employee e2
where e2.employeeID != e.employeeID);

```

employee 1

select e.employeeID, e.name1 fi | Введите SQL выражение чтобы отфильтровать результат

employeeID	name1

Рис. 53.

```

/* 34. Найти номера и имена сотрудников, не имеющих однофамильцев (в
пределах предприятия. */
select e.employeeID, e.name1
from employee e
where e.name1 <> some
(select e2.name1 from employee e2
where e2.employeeID != e.employeeID);

```

employee 1

select e.employeeID, e.name1 fi | Введите SQL выражение чтобы отфильтровать результат

employeeID	name1
1	Smit N
2	Stone J.
3	Asser M.
4	Wood N.
5	Thomson L.
6	Grishin Egor
7	Kiselev Artem

Рис. 54.

```
/* 34. Найти номера отделов и средний возраст служащих для отделов с
минимальным средним возрастом служащих */
select avg_e.departmentID, avg_e.age as avg_age
from (select e.departmentID, avg(e.age) as age
from employee as e
group by e.departmentID) as avg_e
where avg_e.age = (select min(apd.age)
from (select e2.departmentID, avg(e2.age) as age
from employee e2
group by e2.departmentID) as apd);
```

employee 1

	departmentID	avg_age
1	3	30

Рис. 55.

```
/* 35. Найти общее число служащих и максимальный размер зарплаты в
отделах с одинаковым максимальным размером зарплаты */
select sum(t1.departmentID) as number_emp, max(t1.salary) as max_salary
from (select e2.departmentID, max(e2.salary) as salary
from employee e2
group by e2.departmentID) as t1,
(select e2.departmentID, max(e2.salary) as salary
from employee e2
group by e2.departmentID) as t2
where t1.salary = t2.salary and
t1.departmentID != t2.departmentID;
```

Результат 1

	number_emp	max_salary
1	[NULL]	[NULL]

Рис. 56.

```
/* 36. Найти имена, номера отделов и имена руководителей отделов в которых,
размер заработной платы служащих меньше 55000 руб. */
select d.name, d.departmentID, e.name1
from department as d, employee as e
where d.ManagerID = e.employeeID and
exists (select e2.name1
from employee e2
where e2.salary < 55000 and
e2.departmentID = d.departmentID);
```

department(+) 1

	d.name	d.departmentID	e.name1

Рис. 57.

Таким образом был произведён процесс анализа данных и манипуляциям с таблицами по средствам языка *SQL*: удаление/создание базы данных/таблиц, изменение таблиц (добавление/удаление/изменение строк). Покажем итоговые результаты:

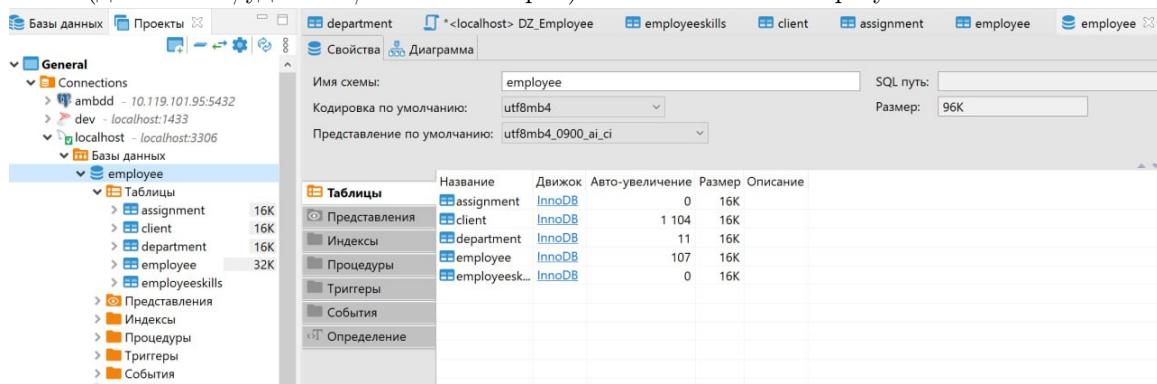


Рис. 58.

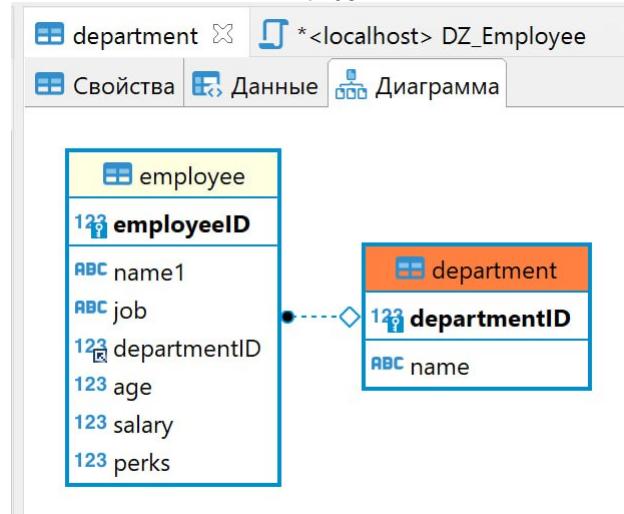


Рис. 59.

The screenshot shows the 'employeeskills' table data. The table has columns: employeeID (PK), skill (FK), and level. The data is as follows:

employeeID	skill	level
1	C++	100
2	Pascal	100
3	Basic	101
4	SQL	102
5	Python	103
6	Delphi	104

Рис. 60.

**department** | Введите SQL выражение чтобы отфильтровать результаты

Таблица	departmentID	name	ManagerID
1	1	Dep_analit	111
2	2	Dep_prog	111
3	3	Dep_admin	105

Рис. 61.

**client** | Введите SQL выражение чтобы отфильтровать результаты

Таблица	clientID	name	adress	contactPerson	contactNumber
1	1 100	ACER	M. 12 st.	Nora	112233445566
2	1 101	MTS	S.P.11 st	Lena	665544332211
3	1 102	Dog	N.N 13 st.	Ivan	123456123456
4	1 103	Cat	K. 14 st.	Petr	654321123456

Рис. 62.

**assignment** | Введите SQL выражение чтобы отфильтровать результаты

Таблица	clientID	employeeID	workdate	hours
1	1 100	100	2009-01-10	120
2	1 101	101	2008-11-01	10
3	1 102	102	2009-12-10	70
4	1 103	102	2009-02-01	100

Рис. 63.

**employee** | Введите SQL выражение чтобы отфильтровать результаты

Таблица	employeeID	name1	job	departmentID	age	salary	perks
1	100	Smit N	Programmer	2	25	38 500	4 200
2	101	Stone J.	manager	3	30	45 000	2 000
3	102	Asser M.	analitic	1	33	36 300	3 780
4	103	Wood N.	Programmer	2	48	33 000	0
5	104	Thomson L.	Programmer	2	27	51 700	0
6	105	Grishin Egor	syst.programmer	2	35	50 000	0
7	106	Kiselev Artem	admin.Programmer	1	38	49 000	0

Рис. 64.