Министерство образования Республики Беларусь

Учреждение образования

«БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ ИНФОМАТИКИ И РАДИОЭЛЕКТРОНИКИ»

Факультет компьютерного проектирования

Кафедра проектирования информационно-компьютерных систем

ОТЧЁТ

Индивидуальное задание

«Работа с БД»

Проверил:

МЕЛЬНИКОВ Дмитрий Васильевич

Выполнил:

ГОМАН Павел Павлович

Студент группы № 014301

Минск 2023

Задача:

Учет выполнения договоров

Компания осуществляет деятельность по выполнению проектов на заказ. По каждому проекту составляется договор с Заказчиком (в 2-х экземплярах для каждой стороны). По каждому договору оформляется два счета - на предоплату и остаток. После выполнения проекта подписывается Акт выполненных работ (в 2-х экземплярах для каждой стороны). Каждый договор сопровождается менеджером со стороны компании, который обеспечивает коммуникации между заказчиком и руководителем проекта. Проекты состоят из нескольких заданий (этапов), каждый из которых имеет свою стоимость в рамках всего проекта. Для каждого задания проекта руководитель проекта составляет график контроля выполнения, включающий несколько дат для каждого задания. По итогу контроля хранится информация о выполнении к дате контроля задания (в процентах), поясняющий комментарий о причинах невыполнения или отставания выполнения задания. Каждый проект имеет руководителя проекта из числа сотрудников. Каждый сотрудник может участвовать в одном или нескольких проектах, или временно не участвовать ни в каких проектах. Над каждым проектом может работать несколько сотрудников отделов, или временно проект может быть приостановлен, тогда над ним не работает ни один сотрудник. Над каждым заданием (этапом) в проекте может работать несколько сотрудников. Для участия в проекте с каждым сотрудником заключается договор на выполнение проектных работ. Каждый сотрудник числится в одном отделе по основной должности согласно штатному расписанию отдела. Сотрудник может работать в другом отделе на условиях штатного совместительства на 0,5 ставки. БД должна содержать следующий минимальный набор сведений: Номер сотрудника. Фамилия сотрудника. Имя сотрудника. Отчество сотрудника. Должность сотрудника. Оклад сотрудника. Название организации-заказчика. Номер организации. Адрес организации. Контактное лицо и его контакты. Номер телефона отдела. Номер отдела. Название отдела. Код проекта. Название проекта. Сроки выполнения проекта. Руководитель проекта. Статус выполнения. Статус оплаты. Номер задания. Дата начала выполнения задания. Срок выполнения задания. Статус выполнении задания. Вознаграждение за задание сотруднику. Статус выполнения задания каждым сотрудником. Дата контроля выполнения задания. Причина невыполнения или отставания выполнения задания.

Постановка задачи:

Разработать систему учета выполнения договоров для компании, занимающейся выполнением проектов на заказ. Система должна обеспечивать хранение, обработку и предоставление информации по всем аспектам работы с договорами, включая фиксацию этапов выполнения работ, финансовых операций и взаимодействия с сотрудниками. Основная цель системы – обеспечение прозрачности и контролируемости процесса выполнения проектных работ, а также оперативного доступа к актуальным данным о ходе выполнения договорных обязательств.

Описание предметной области:

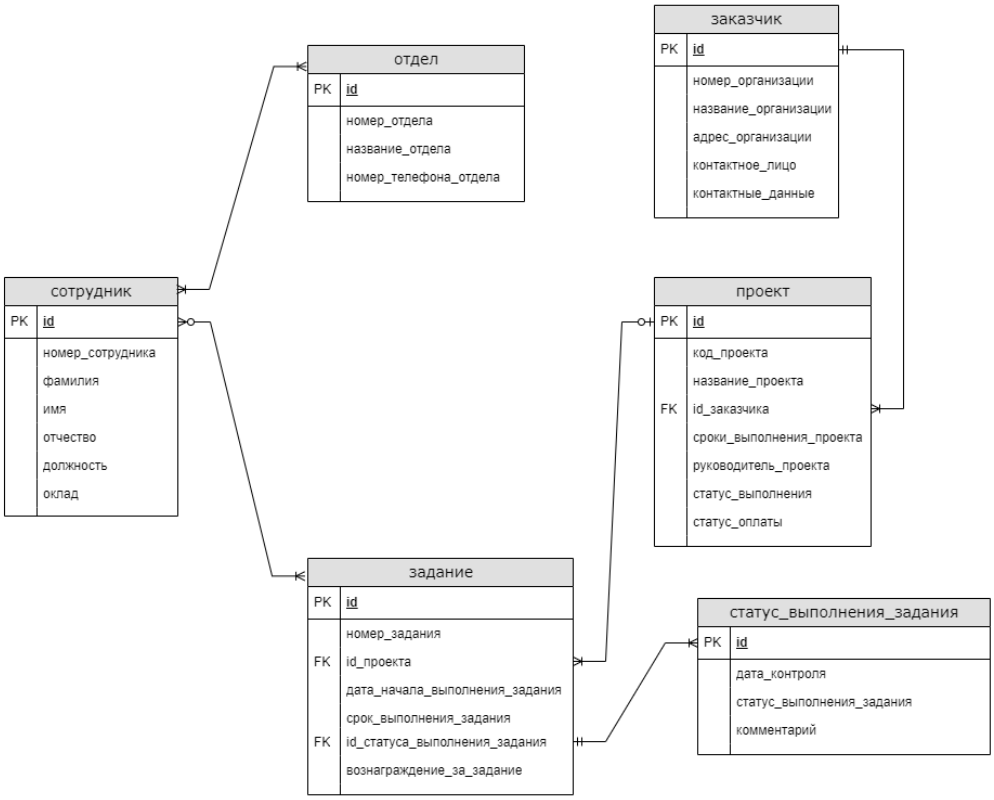
Компания работает по принципу проектного менеджмента, выполняя заказы на разработку и внедрение проектов для внешних клиентов. Проект разбивается на задания (этапы), каждый из которых имеет собственные сроки и стоимость. Управление проектами осуществляется менеджерами, которые отвечают за координацию работы сотрудников и коммуникацию с заказчиками. Важную роль играет планирование и контроль сроков выполнения каждого этапа, а также обеспечение финансовых расчетов между компанией и заказчиком. Эффективность работы компании зависит от точности учета трудозатрат сотрудников и своевременности выполнения заданий.

Участники процесса:

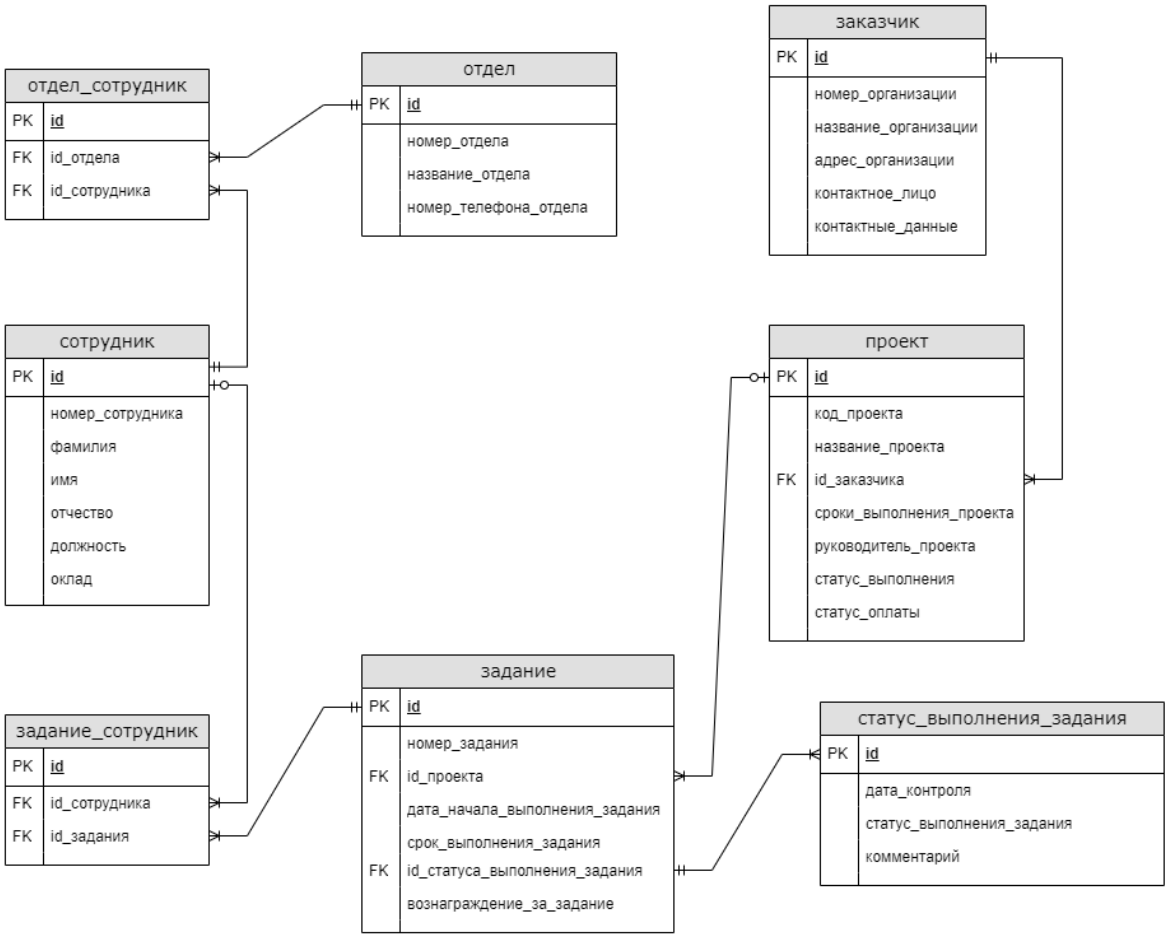
- Проект: набор взаимосвязанных заданий, направленных на достижение определенных целей для заказчика.
- Задание (Этап): часть проекта с определенным сроком выполнения и стоимостью.
- Сотрудник: лицо, привлекаемое к выполнению заданий в рамках проектов.
- Отдел: структурное подразделение компании, в котором числятся сотрудники.
- Заказчик: организация или индивидуальный предприниматель, заключающий договор на выполнение проекта.

Задание 1. ER диаграмма.

Задание 2. Логическая модель БД.



Задание 3. Нормализация реляционных отношений до 3НФ.



Задание 4. Структура таблиц.

DESKTOP-TJA2076\...DB - dbo.customer | DESKTOP-TJA2076\S...- dbo.department

Column Name	Data Type	Allow Nulls
id	int	☐
company_number	nvarchar(50)	☐
company_name	nvarchar(100)	☐
company_address	nvarchar(255)	☐
contact_person	nvarchar(100)	☐
telephone	nvarchar(15)	☑
email	nvarchar(100)	☑

DESKTOP-TJA2076\S...- dbo.department | DESKTOP-TJA2076\...anyDB - Diagram_0

Column Name	Data Type	Allow Nulls
id	int	☐
department_number	int	☐
department_name	nvarchar(100)	☐
telephone	nvarchar(15)	☑

DESKTOP-TJA2076\...B - dbo.employee | DESKTOP-TJA2076\...anyDB - Diagram_0

Column Name	Data Type	Allow Nulls
id	int	☐
employee_number	int	☐
lastname	nvarchar(50)	☐
firstname	nvarchar(50)	☐
surname	nvarchar(50)	☑
position	nvarchar(50)	☐
salary	decimal(15, 2)	☐

DESKTOP-TJA2076\...ployee_department | DESKTOP-TJA2076\...B - dbo.employee

Column Name	Data Type	Allow Nulls
id	int	☐
id_department	int	☑
id_employee	int	☑

DESKTOP-TJA2076\...dbo.employee_task | DESKTOP-TJA2076\...ployee_department

Column Name	Data Type	Allow Nulls
task_id	int	☐
employee_id	int	☐

DESKTOP-TJA2076\S...yDB - dbo.project | DESKTOP-TJA2076\...dbo.employee_task

Column Name	Data Type	Allow Nulls
id	int	☐
project_number	nvarchar(50)	☐
project_name	nvarchar(100)	☐
customer_id	int	☑
deadline	date	☐
project_status	nvarchar(50)	☐
payment_status	nvarchar(50)	☐

DESKTOP-TJA2076\...panyDB - dbo.task | DESKTOP-TJA2076\S...yDB - dbo.project

Column Name	Data Type	Allow Nulls
id	int	☐
task_number	nvarchar(50)	☐
project_id	int	☑
start_date	date	☐
deadline	date	☐
task_status_id	int	☑
reward	decimal(15, 2)	☑

DESKTOP-TJA2076\SQ...- dbo.task_status | DESKTOP-TJA2076\...panyDB - dbo.task

Column Name	Data Type	Allow Nulls
id	int	☐
control_date	date	☐
comment	nvarchar(255)	☑

Задание 5. Данные в таблицах.

DESKTOP-TJA2076\...DB - dbo.customer

id	company_n...	company_n...	company_a...	contact_pe...	telephone	email
1	C001	TechCorp	123 Tech St.	John Doe	+1234567892	johndoe@t...
2	C002	BuildInc	456 Build A...	Jane Smith	+1234567893	janesmith@...
3	C003	SoftSolutions	789 Softwar...	Bill Gates	+1234567894	bill@softsol...
4	C004	NetTech	101 Networ...	Elon Musk	+1234567895	elon@nette...
5	C005	EcoPower	202 Energy ...	Greta Thunb...	+1234567896	greta@eco...
6	C006	HealthFirst	303 Medical...	Ben Carson	+1234567897	ben@health...
7	C007	FoodDelight	404 Tasty Bl...	Gordon Ra...	+1234567898	gordon@fo...
8	C008	AutoDrive	505 Cars Ave.	Jeremy Clar...	+1234567899	jeremy@aut...
9	C009	MobiTech	606 Mobile ...	Steve Jobs	+1234567800	steve@mob...
10	C010	EduLearn	707 Educati...	Albert Einst...	+1234567801	albert@edu...

DESKTOP-TJA2076\S...- dbo.department | DESKTOP-TJA207

id	departmen...	departmen...	telephone
1	1	HR	+1234567890
2	2	Engineering	+1234567891
3	3	Sales	+1234567892
4	4	Marketing	+1234567893
5	5	Finance	+1234567894
6	6	IT Support	+1234567895
7	7	Product Ma...	+1234567896
8	8	Research an...	+1234567897
9	9	Customer S...	+1234567898
10	10	Operations	+1234567899

DESKTOP-TJA2076\...B - dbo.employee | DESKTOP-TJA2076\S...- dbo.department | DESKTOP-TJA

id	employee_...	lastname	firstname	surname	position	salary
1	1001	Ivanov	Ivan	Ivanovich	Manager	50000.00
2	1002	Petrov	Petr	Petrovich	Engineer	40000.00
3	1003	Sidorov	Sergey	Sergeevich	Sales Rep	45000.00
4	1004	Smirnov	Alexey	Alexeevich	Marketing S...	47000.00
5	1005	Popov	Dmitry	Dmitrievich	Finance Ana...	48000.00
6	1006	Vasiliev	Vasily	Vasilievich	IT Support	42000.00
7	1007	Zaitsev	Zahar	Zakharevich	Product Ma...	53000.00
8	1008	Kuznetsov	Konstantin	Konstantino...	R&D Engine...	51000.00
9	1009	Morozov	Maxim	Maximovich	Customer S...	41000.00
10	1010	Pavlov	Pavel	Pavlovich	Operations ...	52000.00

DESKTOP-TJA2076\...ployee_department

id	id_departm...	id_employee
1	1	1
2	1	2
3	2	3
4	2	4
5	3	5
6	3	6
7	4	7
8	4	8
9	5	9
10	5	10

DESKTOP-TJA2076\...dbo.employee_task

	task_id	employee_id
▶	1	7
	2	8
	3	10
	4	1
	5	2
	6	2
	7	3
	8	4
	9	5
	10	6

DESKTOP-TJA2076\S...yDB - dbo.project | DESKTOP-TJA2076\...dbo.employee_task | DESKTOP-T

	id	project_nu...	project_na...	customer_id	deadline	project_sta...	payment_st...
▶	1	P001	Project Alpha	1	2023-12-31	Active	Paid
	2	P002	Project Beta	2	2023-06-30	On Hold	Unpaid
	3	P003	Project Gam...	3	2023-05-15	Active	Partial
	4	P004	Project Delta	4	2023-07-20	Completed	Paid
	5	P005	Project Epsil...	5	2023-09-10	Active	Unpaid
	6	P006	Project Zeta	6	2023-10-05	On Hold	Partial
	7	P007	Project Eta	7	2023-04-22	Active	Paid
	8	P008	Project Theta	8	2023-11-17	Completed	Unpaid
	9	P009	Project Iota	9	2023-08-30	On Hold	Paid
	10	P010	Project Kap...	10	2023-02-14	Active	Partial

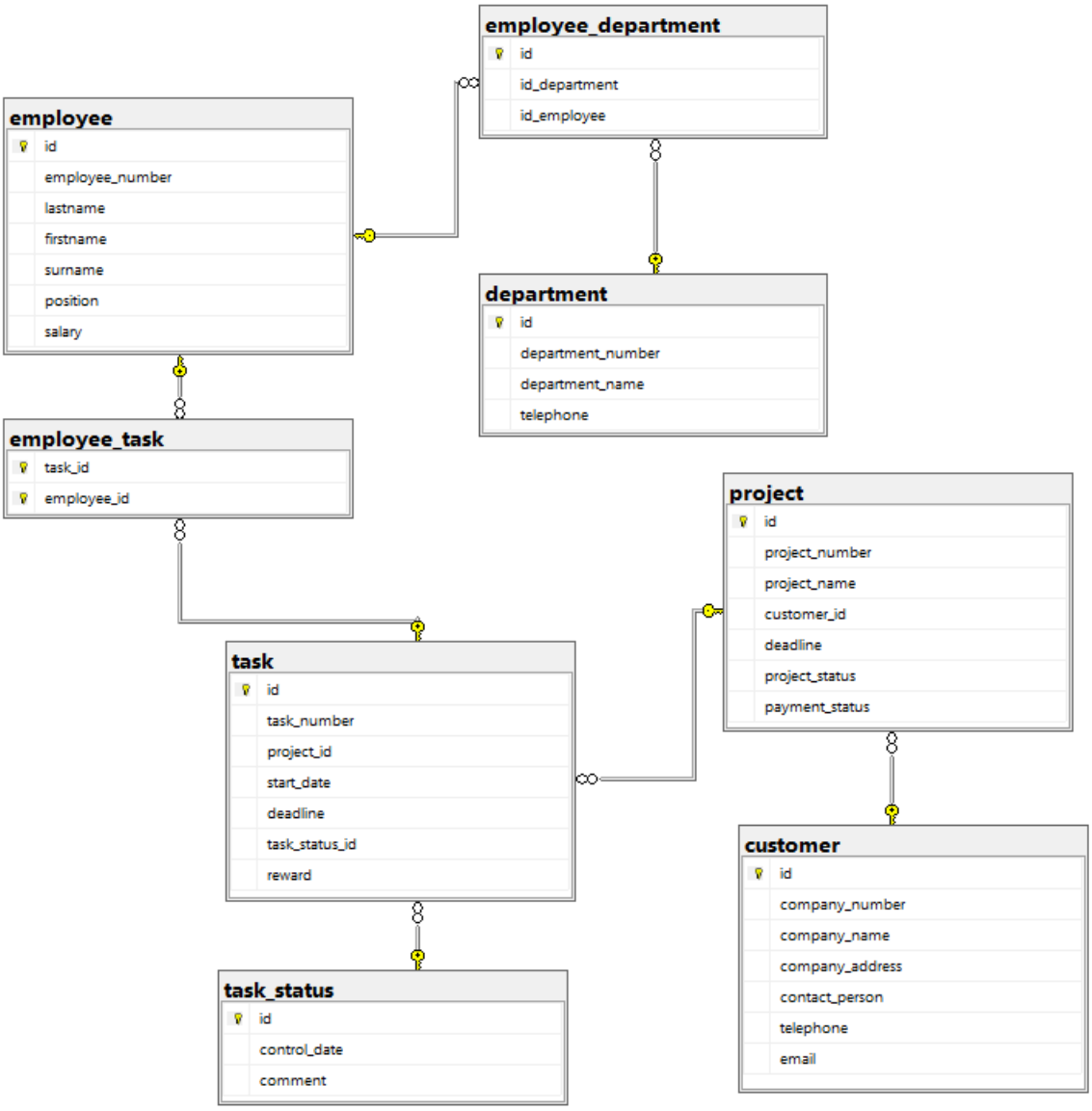
DESKTOP-TJA2076\...panyDB - dbo.task | DESKTOP-TJA2076\S...yDB - dbo.project | DESKTOP-TJ

	id	task_number	project_id	start_date	deadline	task_status_...	reward
▶	1	T001	4	2023-01-01	2023-03-31	1	1000.00
	2	T002	4	2023-02-01	2023-04-30	2	1500.00
	3	T003	5	2023-01-15	2023-05-01	3	1200.00
	4	T004	5	2023-03-10	2023-06-10	4	1100.00
	5	T005	6	2023-02-05	2023-04-20	5	1300.00
	6	T006	6	2023-04-01	2023-05-20	6	1400.00
	7	T007	7	2023-05-10	2023-07-05	7	900.00
	8	T008	7	2023-06-01	2023-07-15	8	800.00
	9	T009	8	2023-07-20	2023-09-01	9	850.00
	10	T010	8	2023-08-01	2023-09-15	10	950.00

DESKTOP-TJA2076\SQ...- dbo.task_status

id	control_date	comment
1	2023-01-01	Started
2	2023-02-01	In Progress
3	2023-03-01	Completed
4	2023-04-01	On Hold
5	2023-05-01	Pending Rev...
6	2023-06-01	Needs Revis...
7	2023-07-01	Approved
8	2023-08-01	Rejected
9	2023-09-01	Scheduled
10	2023-10-01	Delayed

Задание 5. Диаграмма БД.



Задание 6. Вывести все данные из таблиц.

SQLQuery1.sql - D...TJA2076\Home (57))

```sql
use CompanyDB;
-- Вывод всех данных из таблицы сотрудников
SELECT * FROM employee;
-- Вывод всех данных из таблицы отделов
SELECT * FROM department;
-- Вывод всех данных из таблицы связи сотрудников и отделов
SELECT * FROM employee_department;
-- Вывод всех данных из таблицы статусов задач
SELECT * FROM task_status;
-- Вывод всех данных из таблицы клиентов
SELECT * FROM customer;
-- Вывод всех данных из таблицы проектов
SELECT * FROM project;
-- Вывод всех данных из таблицы задач
SELECT * FROM task;
-- Вывод всех данных из таблицы связи сотрудников и задач
SELECT * FROM employee_task;
```

Results | Messages

	id	employee_number	lastname	firstname	surname	position	salary
1	1	1001	Ivanov	Ivan	Ivanovich	Manager	50000.00
2	2	1002	Petrov	Petr	Petrovich	Engineer	40000.00

	id	department_number	department_name	telephone
1	1	1	HR	+1234567890
2	2	2	Engineering	+1234567891

	id	id_department	id_employee
1	1	1	1
2	2	1	2

	id	control_date	comment
1	1	2023-01-01	Started
2	2	2023-02-01	In Progress

	id	company_number	company_name	company_address	contact_person	telephone	email
1	1	C001	TechCorp	123 Tech St.	John Doe	+1234567892	johndoe@techcorp.com
2	2	C002	BuildInc	456 Build Ave.	Jane Smith	+1234567893	janesmith@buildinc.com

	id	project_number	project_name	customer_id	deadline	project_status	payment_status
1	1	P001	Project Alpha	1	2023-12-31	Active	Paid
2	2	P002	Project Beta	2	2023-06-30	On Hold	Unpaid

	id	task_number	project_id	start_date	deadline	task_status_id	reward
3	3	T003	5	2023-01-15	2023-05-01	3	1200.00
4	4	T004	5	2023-03-10	2023-06-10	4	1100.00

	task_id	employee_id
1	1	7
2	2	8

Задание 7. вывести определенные данные, отсортированные в алфавитном порядке по определенному полю.

```sql
use CompanyDB;
-- Вывод списка сотрудников, отсортированный по фамилии
SELECT firstname, lastname FROM employee ORDER BY lastname;
```

99 %

Results | Messages

	firstname	lastname
1	Ivan	Ivanov
2	Konstantin	Kuznetsov
3	Maxim	Morozov
4	Pavel	Pavlov
5	Petr	Petrov
6	Dmitry	Popov
7	Sergey	Sidorov
8	Alexey	Smirnov
9	Vasily	Vasiliev
10	Zahar	Zaitsev

Задание 8. Вывести определенные данные, отсортированные в обратном алфавитном порядке по двум полям.

```sql
use CompanyDB;
-- Вывод списка проектов, отсортированный в обратном алфавитном порядке по названию проекта и статусу платежа
SELECT project_name, payment_status FROM project ORDER BY project_name DESC, payment_status DESC;
```

99 %

Results | Messages

	project_name	payment_status
1	Project Zeta	Partial
2	Project Theta	Unpaid
3	Project Kappa	Partial
4	Project Iota	Paid
5	Project Gamma	Partial
6	Project Eta	Paid
7	Project Epsilon	Unpaid
8	Project Delta	Paid
9	Project Beta	Unpaid
10	Project Alpha	Paid

Задание 9. Создать вычисляемое поле, и вывести данные, убрав дубликаты.

```sql
use CompanyDB;
-- Создать вычисляемое поле, и вывести данные, убрав дубликаты.
-- Пример: выводим список уникальных фамилий сотрудников в формате "Фамилия (Должность)"

SELECT DISTINCT
    lastname + ' (' + position + ')' AS 'Фамилия (Должность)'
FROM
    employee;

```

99 %

Results | Messages

	Фамилия (Должность)
1	Ivanov (Manager)
2	Kuznetsov (R&D Engineer)
3	Morozov (Customer Support)
4	Pavlov (Operations Manager)
5	Petrov (Engineer)
6	Popov (Finance Analyst)
7	Sidorov (Sales Rep)
8	Smirnov (Marketing Specialist)
9	Vasiliev (IT Support)
10	Zaitsev (Product Manager)

Задание 10. Вывести 30% строк.

```sql
use CompanyDB;
-- Вывод 30% строк из таблицы сотрудников
SELECT TOP 30 PERCENT * FROM employee;
```

99 %

Results | Messages

	id	employee_number	lastname	firstname	surname	position	salary
1	1	1001	Ivanov	Ivan	Ivanovich	Manager	50000.00
2	2	1002	Petrov	Petr	Petrovich	Engineer	40000.00
3	3	1003	Sidorov	Sergey	Sergeevich	Sales Rep	45000.00

Задание 11. Вывести первые 5 строк.

```sql
use CompanyDB;
-- Вывод первых 5 записей из таблицы клиентов
SELECT TOP 5 * FROM customer;
```

99 %

Results | Messages

	id	company_number	company_name	company_address	contact_person	telephone	email
1	1	C001	TechCorp	123 Tech St.	John Doe	+1234567892	johndoe@techcorp.com
2	2	C002	BuildInc	456 Build Ave.	Jane Smith	+1234567893	janesmith@buildinc.com
3	3	C003	SoftSolutions	789 Software Rd.	Bill Gates	+1234567894	bill@softsolutions.com
4	4	C004	NetTech	101 Network Ln.	Elon Musk	+1234567895	elon@nettech.com
5	5	C005	EcoPower	202 Energy Pl.	Greta Thunberg	+1234567896	greta@ecopower.com

Задание 12. Вывести строки начиная с 4 строки по 7 строку, отсортированные в обратном порядке.

```sql
use CompanyDB;
-- Вывод строк с 4 по 7 из таблицы задач, отсортированных в обратном порядке по дате начала
SELECT * FROM (
  SELECT *, ROW_NUMBER() OVER (ORDER BY start_date DESC) as row_num
  FROM task
) as tasks_with_row_num
WHERE row_num BETWEEN 4 AND 7;
```

99 %

Results | Messages

	id	task_number	project_id	start_date	deadline	task_status_id	reward	row_num
1	7	T007	7	2023-05-10	2023-07-05	7	900.00	4
2	6	T006	6	2023-04-01	2023-05-20	6	1400.00	5
3	4	T004	5	2023-03-10	2023-06-10	4	1100.00	6
4	5	T005	6	2023-02-05	2023-04-20	5	1300.00	7

Задание 13. Отсортировать данные, используя в качестве критерия сортировки сложно выражение на основе столбцов.

```sql
use CompanyDB;
-- Сортировка сотрудников по отделам и заработной плате
SELECT e.id, e.lastname, e.firstname, e.salary, d.department_name
FROM employee e
JOIN employee_department ed ON e.id = ed.id_employee
JOIN department d ON d.id = ed.id_department
ORDER BY d.department_name, e.salary DESC, e.lastname, e.firstname;
```

99 %

Results | Messages

	id	lastname	firstname	salary	department_name
1	4	Smirnov	Alexey	47000.00	Engineering
2	3	Sidorov	Sergey	45000.00	Engineering
3	10	Pavlov	Pavel	52000.00	Finance
4	9	Morozov	Maxim	41000.00	Finance
5	1	Ivanov	Ivan	50000.00	HR
6	2	Petrov	Petr	40000.00	HR
7	7	Zaitsev	Zahar	53000.00	Marketing
8	8	Kuznetsov	Konstantin	51000.00	Marketing
9	5	Popov	Dmitry	48000.00	Sales
10	6	Vasiliev	Vasily	42000.00	Sales

Задание 14. Запрос на выборку с добавлением SELECT INTO.

```sql
use CompanyDB;
-- Создание отчета о задачах с добавлением в новую таблицу task_report
SELECT t.id AS task_id, t.task_number, ts.control_date, ts.comment INTO task_report
FROM task t
JOIN task_status ts ON t.task_status_id = ts.id;
```

99 %

Messages

```
(10 rows affected)

Completion time: 2023-11-03T23:25:10.0284991+03:00
```

```sql
use CompanyDB;
SELECT t.id AS task_id, t.task_number, ts.control_date, ts.comment
FROM task t
JOIN task_status ts ON t.task_status_id = ts.id;
Select * from task_report
```

99 %

Results | Messages

	task_id	task_number	control_date	comment
1	1	T001	2023-01-01	Started
2	2	T002	2023-02-01	In Progress
3	3	T003	2023-03-01	Completed
4	4	T004	2023-04-01	On Hold
5	5	T005	2023-05-01	Pending Review
6	6	T006	2023-06-01	Needs Revision
7	7	T007	2023-07-01	Approved
8	8	T008	2023-08-01	Rejected
9	9	T009	2023-09-01	Scheduled
10	10	T010	2023-10-01	Delayed

	task_id	task_number	control_date	comment
1	1	T001	2023-01-01	Started
2	2	T002	2023-02-01	In Progress
3	3	T003	2023-03-01	Completed
4	4	T004	2023-04-01	On Hold
5	5	T005	2023-05-01	Pending Review
6	6	T006	2023-06-01	Needs Revision
7	7	T007	2023-07-01	Approved
8	8	T008	2023-08-01	Rejected
9	9	T009	2023-09-01	Scheduled
10	10	T010	2023-10-01	Delayed

Задание 15. 7 запросов на фильтрацию данных с использованием в запросах операторов сравнения, логических операторов.

```sql
-- 1. Найти всех сотрудников с зарплатой выше 45000
SELECT * FROM employee WHERE salary > 45000;
```

109 %

Results | Messages

	id	employee_number	lastname	firstname	surname	position	salary
1	1	1001	Ivanov	Ivan	Ivanovich	Manager	50000.00
2	4	1004	Smirnov	Alexey	Alexeevich	Marketing Specialist	47000.00
3	5	1005	Popov	Dmitry	Dmitrievich	Finance Analyst	48000.00
4	7	1007	Zaitsev	Zahar	Zakharevich	Product Manager	53000.00
5	8	1008	Kuznetsov	Konstantin	Konstantinovich	R&D Engineer	51000.00
6	10	1010	Pavlov	Pavel	Pavlovich	Operations Manager	52000.00

```sql
-- 2. Найти проекты, которые активны и еще не оплачены
SELECT * FROM project WHERE project_status = 'Active' AND payment_status = 'Unpaid';
```

109 %

Results | Messages

	id	project_number	project_name	customer_id	deadline	project_status	payment_status
1	5	P005	Project Epsilon	5	2023-09-10	Active	Unpaid

```sql
-- 3. Получить список задач, которые начались после '2023-01-15' и должны завершиться до '2023-06-30'
SELECT * FROM task WHERE start_date > '2023-01-15' AND deadline < '2023-06-30';
```

109 %

Results | Messages

	id	task_number	project_id	start_date	deadline	task_status_id	reward
1	2	T002	4	2023-02-01	2023-04-30	2	1500.00
2	4	T004	5	2023-03-10	2023-06-10	4	1100.00
3	5	T005	6	2023-02-05	2023-04-20	5	1300.00
4	6	T006	6	2023-04-01	2023-05-20	6	1400.00

```sql
-- 4. Выбрать всех сотрудников, чья фамилия начинается на 'S' или 'P'
SELECT * FROM employee WHERE lastname LIKE 'S%' OR lastname LIKE 'P%';
```

109 %

Results | Messages

	id	employee_number	lastname	firstname	surname	position	salary
1	2	1002	Petrov	Petr	Petrovich	Engineer	40000.00
2	3	1003	Sidorov	Sergey	Sergeevich	Sales Rep	45000.00
3	4	1004	Smirnov	Alexey	Alexeevich	Marketin...	47000.00
4	5	1005	Popov	Dmitry	Dmitrievich	Finance ...	48000.00
5	1...	1010	Pavlov	Pavel	Pavlovich	Operatio...	52000.00

```sql
-- 5. Найти задачи с наградой между 800 и 1500 включительно
SELECT * FROM task WHERE reward BETWEEN 800 AND 1500;
```

132 %

Results | Messages

	id	task_number	project_id	start_date	deadline	task_status_id	reward
1	1	T001	4	2023-01-01	2023-03-31	1	1000.00
2	2	T002	4	2023-02-01	2023-04-30	2	1500.00
3	3	T003	5	2023-01-15	2023-05-01	3	1200.00
4	4	T004	5	2023-03-10	2023-06-10	4	1100.00
5	5	T005	6	2023-02-05	2023-04-20	5	1300.00
6	6	T006	6	2023-04-01	2023-05-20	6	1400.00
7	7	T007	7	2023-05-10	2023-07-05	7	900.00
8	8	T008	7	2023-06-01	2023-07-15	8	800.00

```sql
-- 6. Выбрать задачи, которые были либо отложены, либо нуждаются в пересмотре
SELECT * FROM task_status WHERE comment IN ('Delayed', 'Needs Revision');
```

159 %

Results | Messages

	id	control_date	comment
1	6	2023-06-01	Needs Revision
2	10	2023-10-01	Delayed

```sql
-- 7. Найти сотрудников, которые работают в отделе 'IT Support' или 'Finance' и имеют зарплату выше 45000
SELECT e.* FROM employee e
JOIN employee_department ed ON e.id = ed.id_employee
JOIN department d ON d.id = ed.id_department
WHERE d.department_name IN ('IT Support', 'Finance') AND e.salary > 45000;
```

132 %

Results | Messages

	id	employee_number	lastname	firstname	surname	position	salary
1	10	1010	Pavlov	Pavel	Pavlovich	Operations Manager	52000.00

Задание 16. Переименуйте одну из таблиц вашей БД.

- dbo.task
- dbo.task_status

```sql
use CompanyDB;
-- Переименовываем таблицу "task_status" в "task_progress"
EXEC sp_rename 'task_status', 'task_progress';
```

132 %

Messages

```
Caution: Changing any part of an object name could break scripts and stored procedures.

Completion time: 2023-11-03T23:37:25.4093402+03:00
```

- dbo.task
- dbo.task_progress
- dbo.task_report

Задание 17. Добавьте по две записи в каждую из таблиц вашей БД.

```sql
use CompanyDB;
-- Добавление записей в таблицу "department"
INSERT INTO department (department_number, department_name, telephone) VALUES
(11, 'Legal', '+1234567802'),
(12, 'Quality Assurance', '+1234567803');

-- Добавление записей в таблицу "employee"
INSERT INTO employee (employee_number, lastname, firstname, surname, position, salary) VALUES
(1011, 'Lobov', 'Artem', 'Artemovich', 'Lawyer', 55000),
(1012, 'Sokolov', 'Igor', 'Igorevich', 'QA Engineer', 50000);

-- Так как мы переименовали таблицу "task_status" в "task_progress", добавляем записи в "task_progress"
INSERT INTO task_status (control_date, comment) VALUES
('2023-11-01', 'Under Review'),
('2023-12-01', 'Finalizing');

-- Добавление записей в таблицу "customer"
INSERT INTO customer (company_number, company_name, company_address, contact_person, telephone, email) VALUES
('C011', 'FashionTech', '808 Design St.', 'Donatella Versace', '+1234567804', 'donatella@fashiontech.com'),
('C012', 'SpaceXplore', '909 Space Rd.', 'Neil Armstrong', '+1234567805', 'neil@spacexplore.com');

-- Добавление записей в таблицу "project"
INSERT INTO project (project_number, project_name, customer_id, deadline, project_status, payment_status) VALUES
('P011', 'Project Lambda', 11, '2023-12-15', 'Active', 'Unpaid'),
('P012', 'Project Mu', 12, '2023-08-20', 'Active', 'Partial');

-- Добавление записей в таблицу "task"
INSERT INTO task (task_number, project_id, start_date, deadline, task_status_id, reward) VALUES
('T011', 9, '2023-02-15', '2023-04-15', 1, 2000),
('T012', 10, '2023-03-05', '2023-05-20', 2, 2500);
```

```sql
-- Добавление записей в таблицу "employee_department"
INSERT INTO employee_department (id_department, id_employee) VALUES
(11, 11),
(12, 12);

-- Добавление записей в таблицу "employee_task"
INSERT INTO employee_task (task_id, employee_id) VALUES
(11, 9),
(12, 10);
```

Messages

```
(2 rows affected)

(2 rows affected)

(2 rows affected)

(2 rows affected)

(2 rows affected)

(2 rows affected)

(2 rows affected)

(2 rows affected)
```

Задание 18. Добавьте по смыслу один столбец в любую таблицу.

```sql
use CompanyDB;
-- Добавление столбца "email" в таблицу "employee"
ALTER TABLE employee ADD email NVARCHAR(100) NULL;
```

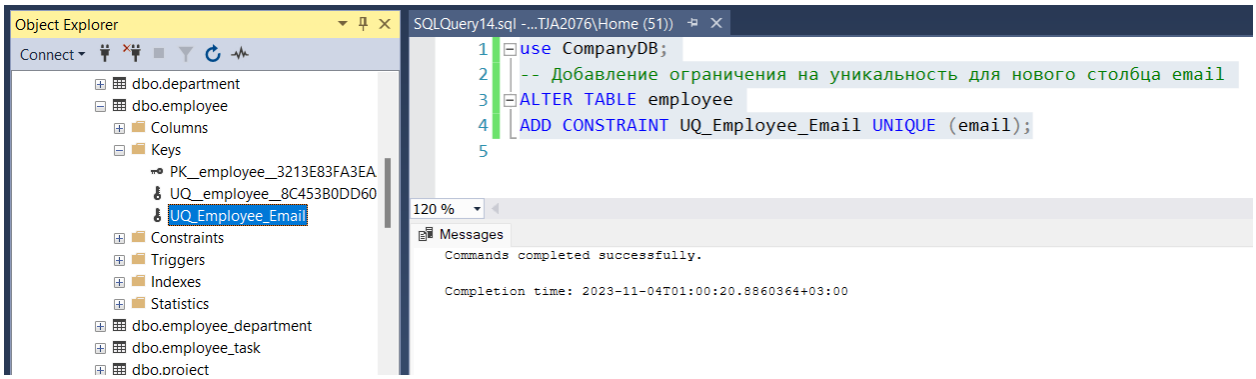
120 %

Messages

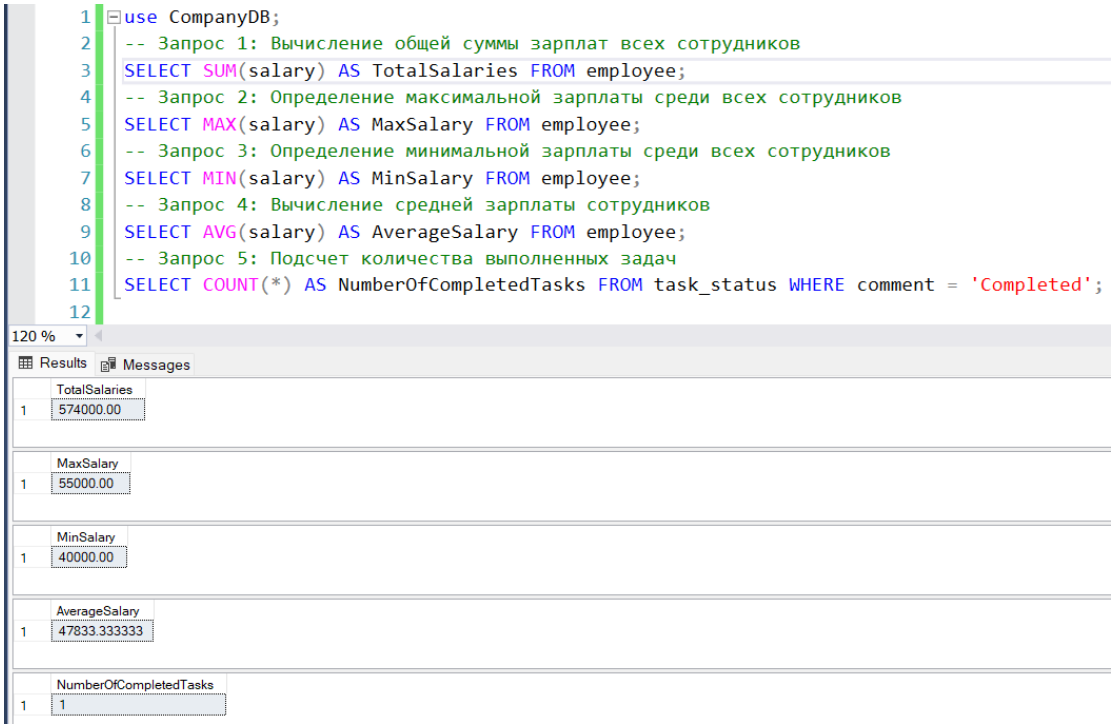
```
Commands completed successfully.

Completion time: 2023-11-03T23:42:21.4502378+03:00
```

Column Name	Data Type	Allow Nulls
id	int	☐
employee_number	int	☐
lastname	nvarchar(50)	☐
firstname	nvarchar(50)	☐
surname	nvarchar(50)	☑
position	nvarchar(50)	☐
salary	decimal(15, 2)	☐
email	nvarchar(100)	☑

Задание 19. Добавьте ограничение с именем на новый введенный вами столбец.



Задание 20. Напишите 5 запросов, используя агрегатные функции (SUM, MAX, MIN, AVG, COUNT).



```
use CompanyDB;
-- Запрос 1: Вычисление общей суммы зарплат всех сотрудников
SELECT SUM(salary) AS TotalSalaries FROM employee;
-- Запрос 2: Определение максимальной зарплаты среди всех сотрудников
SELECT MAX(salary) AS MaxSalary FROM employee;
-- Запрос 3: Определение минимальной зарплаты среди всех сотрудников
SELECT MIN(salary) AS MinSalary FROM employee;
-- Запрос 4: Вычисление средней зарплаты сотрудников
SELECT AVG(salary) AS AverageSalary FROM employee;
-- Запрос 5: Подсчет количества выполненных задач
SELECT COUNT(*) AS NumberOfCompletedTasks FROM task_status WHERE comment = 'Completed';
```

120 %

Results | Messages

	TotalSalaries
1	574000.00

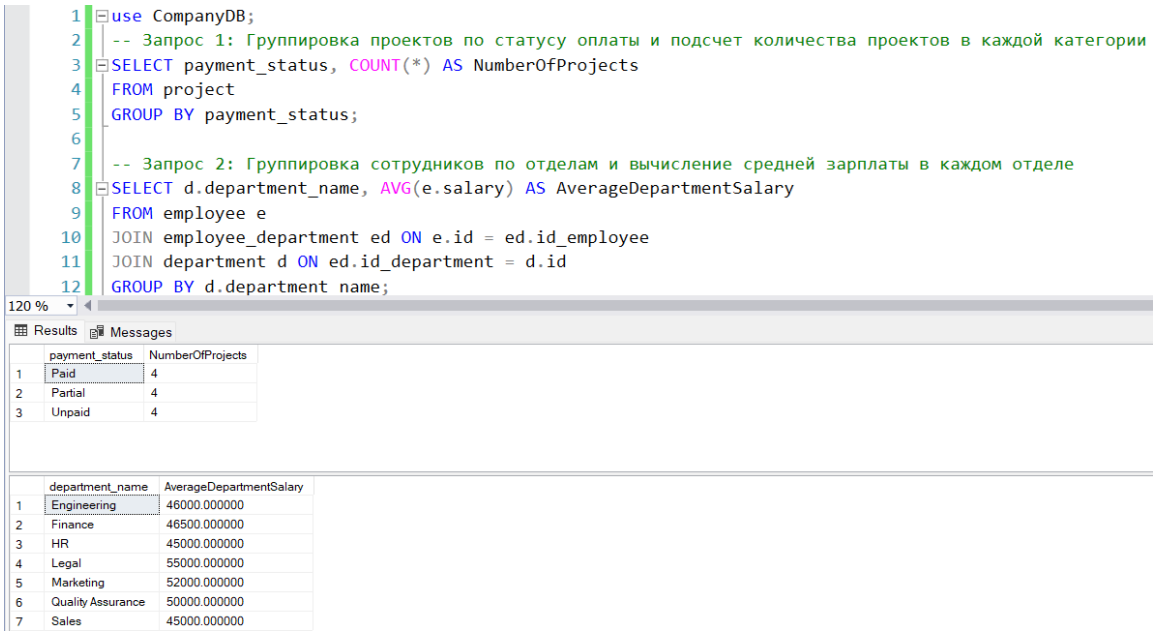
	MaxSalary
1	55000.00

	MinSalary
1	40000.00

	AverageSalary
1	47833.333333

	NumberOfCompletedTasks
1	1

Задание 21. Написать 2 запроса на группировку данных, используя оператор GROUP BY.



```
use CompanyDB;
-- Запрос 1: Группировка проектов по статусу оплаты и подсчет количества проектов в каждой категории
SELECT payment_status, COUNT(*) AS NumberOfProjects
FROM project
GROUP BY payment_status;

-- Запрос 2: Группировка сотрудников по отделам и вычисление средней зарплаты в каждом отделе
SELECT d.department_name, AVG(e.salary) AS AverageDepartmentSalary
FROM employee e
JOIN employee_department ed ON e.id = ed.id_employee
JOIN department d ON ed.id_department = d.id
GROUP BY d.department name;
```

120 %

Results | Messages

	payment_status	NumberOfProjects
1	Paid	4
2	Partial	4
3	Unpaid	4

	department_name	AverageDepartmentSalary
1	Engineering	46000.000000
2	Finance	46500.000000
3	HR	45000.000000
4	Legal	55000.000000
5	Marketing	52000.000000
6	Quality Assurance	50000.000000
7	Sales	45000.000000

Задание 22. Написать 2 запроса на фильтрацию групп, используя HAVING.

```sql
use CompanyDB;
-- 1. Запрос для поиска отделов с средней зарплатой выше 45000
SELECT d.department_name, AVG(e.salary) as avg_salary
FROM employee e
JOIN employee_department ed ON e.id = ed.id_employee
JOIN department d ON ed.id_department = d.id
GROUP BY d.department_name
HAVING AVG(e.salary) > 45000;
-- 2. Запрос для поиска проектов с общей суммой выплат сотрудникам более 2000
SELECT p.project_name, SUM(t.reward) as total_reward
FROM project p
JOIN task t ON p.id = t.project_id
JOIN employee_task et ON t.id = et.task_id
GROUP BY p.project_name
HAVING SUM(t.reward) > 2000;
```

120 %

Results | Messages

	department_name	avg_salary
1	Engineering	46000.000000
2	Finance	46500.000000
3	Legal	55000.000000
4	Marketing	52000.000000
5	Quality Assurance	50000.000000

	project_name	total_reward
1	Project Delta	2500.00
2	Project Epsilon	2300.00
3	Project Kappa	2500.00
4	Project Zeta	2700.00

Задание 23. Написать запросы на расширения SQL Server для группировки (ROLLUP, CUBE, GROUPING SETS, OVER).

```sql
-- Использование ROLLUP для получения субтоталов и общих итогов по отделам и их зарплатам
SELECT d.department_name, e.position, SUM(e.salary) as total_salary
FROM employee e
JOIN employee_department ed ON e.id = ed.id_employee
JOIN department d ON ed.id_department = d.id
GROUP BY ROLLUP(d.department_name, e.position);
```

159 %

Results | Messages

	department_name	position	total_salary
1	Engineering	Marketing Specialist	47000.00
2	Engineering	Sales Rep	45000.00
3	Engineering	NULL	92000.00
4	Finance	Customer Support	41000.00
5	Finance	Operations Manager	52000.00
6	Finance	NULL	93000.00
7	HR	Engineer	40000.00
8	HR	Manager	50000.00

```sql
-- Использование CUBE для получения всех возможных агрегаций по отделам и позициям
SELECT d.department_name, e.position, SUM(e.salary) as total_salary
FROM employee e
JOIN employee_department ed ON e.id = ed.id_employee
JOIN department d ON ed.id_department = d.id
GROUP BY CUBE(d.department_name, e.position);
```

159 %

Results | Messages

	department_name	position	total_salary
1	Finance	Customer Support	41000.00
2	NULL	Customer Support	41000.00
3	HR	Engineer	40000.00
4	NULL	Engineer	40000.00
5	Sales	Finance Analyst	48000.00
6	NULL	Finance Analyst	48000.00
7	Sales	IT Support	42000.00
8	NULL	IT Support	42000.00

```sql
-- Использование GROUPING SETS для получения агрегаций по отделам, позициям и их комбинациям
SELECT d.department_name, e.position, SUM(e.salary) as total_salary
FROM employee e
JOIN employee_department ed ON e.id = ed.id_employee
JOIN department d ON ed.id_department = d.id
GROUP BY GROUPING SETS((d.department_name), (e.position), (d.department_name, e.position));
```

159 %

Results | Messages

	department_name	position	total_salary
24	NULL	Sales Rep	45000.00
25	Engineering	NULL	92000.00
26	Finance	NULL	93000.00
27	HR	NULL	90000.00
28	Legal	NULL	55000.00
29	Marketing	NULL	104000.00
30	Quality Assurance	NULL	50000.00
31	Sales	NULL	90000.00

```sql
-- Использование OVER для расчета средней зарплаты по всем сотрудникам в каждой строке
SELECT e.firstname, e.lastname, e.salary, AVG(e.salary) OVER() as avg_salary_across_all
FROM employee e;
```

159 %

Results | Messages

	firstname	lastname	salary	avg_salary_across_all
1	Ivan	Ivanov	50000.00	47833.333333
2	Petr	Petrov	40000.00	47833.333333
3	Sergey	Sidorov	45000.00	47833.333333
4	Alexey	Smirnov	47000.00	47833.333333
5	Dmitry	Popov	48000.00	47833.333333
6	Vasily	Vasiliev	42000.00	47833.333333
7	Zahar	Zaitsev	53000.00	47833.333333
8	Konsta...	Kuznet...	51000.00	47833.333333

Задание 24. Написать запрос на разворачивание данных (PIVOT), отмена разворачивания (UNPIVOT).

```sql
-- Пример использования PIVOT для просмотра суммарной зарплаты по каждой должности
SELECT *
FROM (
    SELECT position, salary
    FROM employee
) AS SourceTable
PIVOT (
    SUM(salary)
    FOR position IN ([Manager], [Engineer]
                    , [Sales Rep], [Marketing Specialist]
                    , [Finance Analyst], [IT Support]
                    , [Product Manager], [R&D Engineer]
                    , [Customer Support], [Operations Manager])
) AS PivotTable;
```

120 %

Results | Messages

	Manager	Engineer	Sales Rep	Marketing Specialist	Finance Analyst	IT Support	Product Manager	R&D Engineer	Customer Support	Operations Manager
1	50000.00	40000.00	45000.00	47000.00	48000.00	42000.00	53000.00	51000.00	41000.00	52000.00

```sql
-- Пример использования UNPIVOT для преобразования данных о статусе проектов в формат ключ-значение
SELECT id, StatusType, StatusValue
FROM
(
    SELECT id, project_status, payment_status
    FROM project
) AS SourceTable
UNPIVOT
(
    StatusValue FOR StatusType IN (project_status, payment_status)
) AS UnpivotTable;
```

120 %

Results | Messages

	id	StatusType	StatusValue
1	1	project_status	Active
2	1	payment_status	Paid
3	2	project_status	On Hold
4	2	payment_status	Unpaid
5	3	project_status	Active
6	3	payment_status	Partial
7	4	project_status	Completed
8	4	payment_status	Paid
9	5	project_status	Active
10	5	payment_status	Unpaid
11	6	project_status	On Hold
12	6	payment_status	Partial
13	7	project_status	Active
14	7	payment_status	Paid
15	8	project_status	Completed

Задание 25. Запрос на внутреннее соединение таблиц.

```sql
use CompanyDB;
-- Запрос 1: Получение информации о сотрудниках и их отделах
SELECT e.firstname, e.lastname, d.department_name
FROM employee e
INNER JOIN employee_department ed ON e.id = ed.id_employee
INNER JOIN department d ON ed.id_department = d.id;

-- Запрос 2: Получение информации о проектах и статусах их оплаты
SELECT p.project_name, p.project_status, p.payment_status
FROM project p
INNER JOIN customer c ON p.customer_id = c.id
WHERE p.payment_status = 'Unpaid';
```

120 %

Results | Messages

	firstname	lastname	department_name
1	Ivan	Ivanov	HR
2	Petr	Petrov	HR
3	Sergey	Sidorov	Engineering
4	Alexey	Smirnov	Engineering
5	Dmitry	Popov	Sales
6	Vasily	Vasiliev	Sales
7	Zahar	Zaitsev	Marketing
8	Konstantin	Kuznetsov	Marketing
9	Maxim	Morozov	Finance
10	Pavel	Pavlov	Finance

	project_name	project_status	payment_status
1	Project Beta	On Hold	Unpaid
2	Project Epsilon	Active	Unpaid
3	Project Theta	Completed	Unpaid
4	Project Lambda	Active	Unpaid

Задание 26. Запрос на внешнее левое соединение таблиц.

```sql
use CompanyDB;
-- Запрос 1: Получение списка всех сотрудников и задач, которые им назначены (если есть)
SELECT e.firstname, e.lastname, t.task_number
FROM employee e
LEFT JOIN employee_task et ON e.id = et.employee_id
LEFT JOIN task t ON et.task_id = t.id;

-- Запрос 2: Получение списка всех проектов, включая информацию о клиентах, даже если проекты не имеют клиентов
SELECT p.project_name, c.company_name
FROM project p
LEFT JOIN customer c ON p.customer_id = c.id;
```

120 %

Results | Messages

	firstname	lastname	task_number
1	Ivan	Ivanov	T004
2	Petr	Petrov	T005
3	Petr	Petrov	T006
4	Sergey	Sidorov	T007
5	Alexey	Smirnov	T008
6	Dmitry	Popov	T009
7	Vasily	Vasiliev	T010
8	Zahar	Zaitsev	T001

	project_name	company_name
1	Project Alpha	TechCorp
2	Project Beta	BuildInc
3	Project Gamma	SoftSolutions
4	Project Delta	NetTech
5	Project Epsilon	EcoPower
6	Project Zeta	HealthFirst
7	Project Eta	FoodDelight
8	Project Theta	AutoDrive
9	Project Iota	MobiTech
10	Project Kappa	EduLearn
11	Project Lambda	FashionTech
12	Project Mu	SpaceXplore

Задание 27. Запрос на внешнее правое соединение таблиц.

```sql
use CompanyDB;
-- Запрос 1: Получение списка всех отделов и сотрудников, которые в них работают (включая отделы без сотрудников)
SELECT d.department_name, e.firstname, e.lastname
FROM department d
RIGHT JOIN employee_department ed ON d.id = ed.id_department
RIGHT JOIN employee e ON ed.id_employee = e.id;

-- Запрос 2: Получение списка всех задач и соответствующих им статусов, включая задачи без статуса
SELECT t.task_number, ts.comment
FROM task t
RIGHT JOIN task_status ts ON t.task_status_id = ts.id;
```

120 %

Results | Messages

	department_name	firstname	lastname
1	HR	Ivan	Ivanov
2	HR	Petr	Petrov
3	Engineering	Sergey	Sidorov
4	Engineering	Alexey	Smirnov
5	Sales	Dmitry	Popov
6	Sales	Vasily	Vasiliev
7	Marketing	Zahar	Zaitsev
8	Marketing	Konstantin	Kuznetsov

	task_number	comment
1	T001	Started
2	T011	Started
3	T002	In Progress
4	T012	In Progress
5	T003	Completed
6	T004	On Hold
7	T005	Pending ...
8	T006	Needs R...
9	T007	Approved
10	T008	Rejected
11	T009	Scheduled
12	T010	Delayed

Задание 28. Запрос на полное внешнее соединение таблиц.

```sql
use CompanyDB;
-- Запрос 1: Получение списка всех сотрудников и проектов, независимо от того, назначены ли они на проекты
SELECT e.firstname, e.lastname, p.project_name
FROM employee e
FULL OUTER JOIN employee_task et ON e.id = et.employee_id
FULL OUTER JOIN task t ON et.task_id = t.id
FULL OUTER JOIN project p ON t.project_id = p.id;

-- Запрос 2: Получение списка всех клиентов и их проектов, включая клиентов без проектов и проекты без клиентов
SELECT c.company_name, p.project_name
FROM customer c
FULL OUTER JOIN project p ON c.id = p.customer_id;
```

120 %

Results | Messages

	firstname	lastname	project_name
10	Maxim	Morozov	Project Iota
11	Pavel	Pavlov	Project Epsilon
12	Pavel	Pavlov	Project Kappa
13	Artem	Lobov	NULL
14	Igor	Sokolov	NULL
15	NULL	NULL	Project Gamma
16	NULL	NULL	Project Mu
17	NULL	NULL	Project Alpha

	company_name	project_name
1	TechCorp	Project Alpha
2	BuildInc	Project Beta
3	SoftSolutions	Project Gamma
4	NetTech	Project Delta
5	EcoPower	Project Epsilon
6	HealthFirst	Project Zeta
7	FoodDelight	Project Eta
8	AutoDrive	Project Theta
9	MobiTech	Project Iota
10	EduLearn	Project Kappa
11	FashionTech	Project Lambda

Задание 29. Запрос на перекрестное соединение таблиц.

```sql
use CompanyDB;
-- Запрос 1: Получение декартова произведения сотрудников и отделов (каждый с каждым)
SELECT e.firstname, e.lastname, d.department_name
FROM employee e
CROSS JOIN department d;

-- Запрос 2: Получение декартова произведения статусов задач и статусов оплаты проектов
SELECT ts.comment AS TaskStatus, ps.payment_status AS PaymentStatus
FROM task_status ts
CROSS JOIN (SELECT DISTINCT payment_status FROM project) ps;
```

120 %

Results | Messages

	firstname	lastname	department_name
1	Ivan	Ivanov	HR
2	Petr	Petrov	HR
3	Sergey	Sidorov	HR
4	Alexey	Smirnov	HR
5	Dmitry	Popov	HR
6	Vasily	Vasiliev	HR
7	Zahar	Zaitsev	HR
8	Konstantin	Kuznetsov	HR
9	Maxim	Morozov	HR
10	Pavel	Pavlov	HR
11	Artem	Lobov	HR
12	Igor	Sokolov	HR

	TaskStatus	PaymentStatus
1	Started	Paid
2	In Progress	Paid
3	Completed	Paid
4	On Hold	Paid
5	Pending Review	Paid
6	Needs Revision	Paid
7	Approved	Paid
8	Rejected	Paid
9	Scheduled	Paid
10	Delayed	Paid

Задание 30. Запрос на пересечение.

```sql
use CompanyDB;
-- Запрос 1: на пересечение для нахождения проектов,
-- которые активны и по которым были начислены вознаграждения сотрудникам

SELECT p.id, p.project_name
FROM project p
WHERE p.project_status = 'Active'

INTERSECT

SELECT p.id, p.project_name
FROM project p
INNER JOIN task t ON p.id = t.project_id
WHERE t.reward > 0;
```

120 %

Results | Messages

	id	project_name
1	5	Project Epsilon
2	7	Project Eta
3	10	Project Kappa

```sql
use CompanyDB;
-- Запрос 2: на пересечение для нахождения сотрудников,
-- которые работают в отделе HR и участвуют в задачах проекта 'Project Alpha'

SELECT e.id, e.firstname, e.lastname
FROM employee e
INNER JOIN employee_department ed ON e.id = ed.id_employee
INNER JOIN department d ON ed.id_department = d.id
WHERE d.department_name = 'Engineering'

INTERSECT

SELECT e.id, e.firstname, e.lastname
FROM employee e
INNER JOIN employee_task et ON e.id = et.employee_id
INNER JOIN task t ON et.task_id = t.id
INNER JOIN project p ON t.project_id = p.id
WHERE p.project_name = 'Project Alpha';
```

99 %

Results | Messages

id	firstname	lastname

Задание 31. Запрос на разность.

```sql
use CompanyDB;
-- Запрос 1: Получение списка сотрудников,
-- которые не участвуют ни в одном проекте
SELECT e.id, e.firstname, e.lastname
FROM employee e
EXCEPT
SELECT et.employee_id, e.firstname, e.lastname
FROM employee_task et
JOIN employee e ON e.id = et.employee_id;

-- Запрос 2: Получение списка проектов, у которых нет задач,
-- начавшихся до сегодняшнего дня (или на сегодняшний день)
SELECT p.id, p.project_name
FROM project p
EXCEPT
SELECT t.project_id, p.project_name
FROM task t
JOIN project p ON p.id = t.project_id
WHERE t.start_date <= CAST(GETDATE() AS DATE);
```

99 %

Results | Messages

	id	firstname	lastname
1	11	Artem	Lobov
2	12	Igor	Sokolov

	id	project_name
1	1	Project Alpha
2	2	Project Beta
3	3	Project Gamma
4	11	Project Lambda
5	12	Project Mu

Задание 32. Запрос на объединение таблиц.

SQLQuery6.sql - D...TJA2076\Home (62)) | SQLQuery8.sql - D...TJA2076\Home (52)) | SQLQuery7.sql - D...TJA2076\Home (74)) | SQLQuery9.sq

```sql
use CompanyDB;
-- Запрос 1: Запрос на объединение таблиц employee и department с использованием CROSS APPLY

SELECT
    e.id AS EmployeeID,
    e.firstname,
    e.lastname,
    d.department_name
FROM
    employee e
CROSS APPLY
    department d;
```

118 %

Results | Messages

	EmployeeID	firstname	lastname	department_name
1	1	Ivan	Ivanov	HR
2	2	Petr	Petrov	HR
3	3	Sergey	Sidorov	HR
4	4	Alexey	Smirnov	HR
5	5	Dmitry	Popov	HR
6	6	Vasily	Vasiliev	HR
7	7	Zahar	Zaitsev	HR
8	8	Konstantin	Kuznetsov	HR
9	9	Maxim	Morozov	HR
10	10	Pavel	Pavlov	HR
11	11	Artem	Lobov	HR
12	12	Igor	Sokolov	HR
13	1	Ivan	Ivanov	Engineering
14	2	Petr	Petrov	Engineering
15	3	Sergey	Sidorov	Engineering
16	4	Alexey	Smirnov	Engineering
17	5	Dmitry	Popov	Engineering
18	6	Vasily	Vasiliev	Engineering
19	7	Zahar	Zaitsev	Engineering
20	8	Konstantin	Kuznetsov	Engineering
21	9	Maxim	Morozov	Engineering

Query executed successfully. | DESKTOP-TJA2076\SQLEXPRESS ... | DESKTO

```sql
-- Запрос 2: Запрос на объединение таблиц customer и project с использованием OUTER APPLY

SELECT
    c.id AS CustomerID,
    c.company_name,
    p.project_name
FROM
    customer c
OUTER APPLY
    project p;
```

118 %

Results | Messages

	CustomerID	company_name	project_name
1	1	TechCorp	Project Alpha
2	1	TechCorp	Project Beta
3	1	TechCorp	Project Gamma
4	1	TechCorp	Project Delta
5	1	TechCorp	Project Epsilon
6	1	TechCorp	Project Zeta
7	1	TechCorp	Project Eta
8	1	TechCorp	Project Theta
9	1	TechCorp	Project Iota
10	1	TechCorp	Project Kappa
11	1	TechCorp	Project Lambda
12	1	TechCorp	Project Mu
13	2	BuildInc	Project Alpha
14	2	BuildInc	Project Beta
15	2	BuildInc	Project Gamma
16	2	BuildInc	Project Delta
17	2	BuildInc	Project Epsilon
18	2	BuildInc	Project Zeta
19	2	BuildInc	Project Eta
20	2	BuildInc	Project Theta
21	2	BuildInc	Project Iota
22	2	BuildInc	Project Kappa

Задание 33. Написать 4 запроса с использованием подзапросов, используя операторы сравнения, операторы IN, ANY|SOME и ALL, предикат EXISTS.

```sql
use CompanyDB;
-- Запрос 1: Выбираем сотрудников, чья зарплата выше средней зарплаты в компании
SELECT *
FROM employee
WHERE salary > (SELECT AVG(salary) FROM employee);
-- Запрос 2: Выбираем проекты, где контактное лицо клиента является одним из указанных
SELECT p.*
FROM project p
JOIN customer c ON p.customer_id = c.id
WHERE c.contact_person IN ('John Doe', 'Jane Smith', 'Bill Gates');
```

90 %

Results | Messages

	id	employee_number	lastname	firstname	surname	position	salary
1	1	1001	Ivanov	Ivan	Ivanovich	Manager	50000.00
2	5	1005	Popov	Dmitry	Dmitrievich	Finance Analyst	48000.00
3	7	1007	Zaitsev	Zahar	Zakharevich	Product Manager	53000.00
4	8	1008	Kuznetsov	Konstantin	Konstantinovich	R&D Engineer	51000.00
5	10	1010	Pavlov	Pavel	Pavlovich	Operations Manager	52000.00
6	11	1011	Lobov	Artem	Artemovich	Lawyer	55000.00
7	12	1012	Sokolov	Igor	Igorevich	QA Engineer	50000.00

	id	project_number	project_name	customer_id	deadline	project_status	payment_status
1	1	P001	Project Alpha	1	2023-12-31	Active	Paid
2	2	P002	Project Beta	2	2023-06-30	On Hold	Unpaid
3	3	P003	Project Gamma	3	2023-05-15	Active	Partial

```sql
use CompanyDB;
-- Запрос 3: Находим задачи с наградой выше, чем у любой задачи в проекте 'Project Delta'
SELECT t.*
FROM task t
WHERE t.reward > ANY (
    SELECT t2.reward
    FROM task t2
    JOIN project p ON t2.project_id = p.id
    WHERE p.project_name = 'Project Delta'
);
-- Запрос 4: Выбираем отделы, в которых существует хотя бы один сотрудник с зарплатой выше 50000
SELECT d.*
FROM department d
WHERE EXISTS (
    SELECT 1
    FROM employee e
    JOIN employee_department ed ON e.id = ed.id_employee
    WHERE ed.id_department = d.id AND e.salary > 50000
);
```

109 %

Results | Messages

	id	task_number	project_id	start_date	deadline	task_status_id	reward
1	2	T002	4	2023-02-01	2023-04-30	2	1500.00
2	3	T003	5	2023-01-15	2023-05-01	3	1200.00
3	4	T004	5	2023-03-10	2023-06-10	4	1100.00
4	5	T005	6	2023-02-05	2023-04-20	5	1300.00
5	6	T006	6	2023-04-01	2023-05-20	6	1400.00
6	11	T011	9	2023-02-15	2023-04-15	1	2000.00
7	12	T012	10	2023-03-05	2023-05-20	2	2500.00

	id	department_number	department_name	telephone
1	4	4	Marketing	+1234567893
2	5	5	Finance	+1234567894
3	11	11	Legal	+1234567802

Задание 34. Написать 4 запроса по строковым функциям

SQLQuery1.sql - D...TJA2076\Home (79)) | SQLQuery2.sql - D...TJA2076\Home (78)) | SQLQuery5.sql - D...TJA2076\Home (66)) | SQ

```sql
USE CompanyDB;
-- Запросы с использованием строковых функций
-- Выбор сотрудников с конкатенацией имени и фамилии
SELECT id, CONCAT(firstname, ' ', surname, ' ', lastname) AS Full_Name, position
FROM employee;
-- Поиск компаний, названия которых начинаются на определённую букву
SELECT company_number, company_name
FROM customer
WHERE company_name LIKE 'T%';
-- Запрос для выбора названий проектов, где все буквы преобразованы в верхний регистр
SELECT UPPER(project_name) AS project_name_uppercase FROM project;
-- Подсчёт количества символов в комментариях к статусу задачи
SELECT id, comment, LEN(comment) AS Comment_Length
FROM task_status;
```

118 %

Results | Messages

	id	Full_Name	position
1	1	Ivan Ivanovich Ivanov	Manager
2	2	Petr Petrovich Petrov	Engineer
3	3	Sergey Sergeevich Sidorov	Sales Rep
4	4	Alexey Alexeevich Smirnov	Marketing Specialist
5	5	Dmitry Dmitrievich Popov	Finance Analyst

	company_number	company_name
1	C001	TechCorp

	project_name_uppercase
1	PROJECT ALPHA
2	PROJECT BETA
3	PROJECT GAMMA
4	PROJECT DELTA

	id	comment	Comment_Length
1	1	Started	7
2	2	In Progress	11
3	3	Completed	9
4	4	On Hold	7

Query executed successfully. | DESKTOP-TJA2076\SQLEXPRES

Задание 35. Написать 4 запроса по числовым функциям

SQLQuery1.sql - D...TJA2076\Home (79)) | SQLQuery2.sql - D...TJA2076\Home (78)) | SQLQuery5.sql - D...TJA2076\Home (66)) | SQLQuery8.sql - D...TJA2076\Home (65))

```sql
USE CompanyDB;
-- Запросы с использованием числовых функций
-- Запрос для расчета общего количества сотрудников в каждом отделе
SELECT id_department, COUNT(*) AS total_employees FROM employee_department GROUP BY id_department;
-- Запрос для округления зарплаты до ближайшего целого числа.
SELECT id, firstname, lastname, salary, ROUND(salary, 0) AS Rounded_Salary
FROM employee;
-- Запрос для расчета средней зарплаты сотрудников
SELECT AVG(salary) AS average_salary FROM employee;
-- Запрос для выбора заданий, где срок выполнения (количество дней между началом и концом задания) превышает 30 дней
SELECT *, DATEDIFF(day, start_date, deadline) AS duration FROM task WHERE DATEDIFF(day, start_date, deadline) > 30;
```

118 %

Results | Messages

	id_department	total_employees
1	1	2
2	2	2
3	3	2
4	4	2
5	5	2
6	11	1
7	12	1

	id	firstname	lastname	salary	Rounded_Salary
1	1	Ivan	Ivanov	55000.00	55000.00
2	2	Petr	Petrov	40000.00	40000.00
3	3	Sergey	Sidorov	45000.00	45000.00
4	4	Alexey	Smirnov	47000.00	47000.00
5	5	Dmitry	Popov	48000.00	48000.00
6	6	Vasily	Vasiliev	42000.00	42000.00

	average_salary
1	48250.000000

	id	task_number	project_id	start_date	deadline	task_status_id	reward	duration
1	1	T001	4	2023-01-01	2023-03-31	1	1000.00	89
2	2	T002	4	2023-02-01	2023-04-30	2	1500.00	88
3	3	T003	5	2023-01-15	2023-05-01	3	1200.00	106
4	4	T004	5	2023-03-10	2023-06-10	4	1100.00	92

Query executed successfully. | DESKTOP-TJA2076\SQLEXPRESS ... | DESKTOP-TJA2076\Home (78) | CompanyDB

Задание 36. Создайте 5 представлений по своей БД

SQLQuery3.sql - D...TJA2076\Home (72)) | SQLQuery2.sql - D...TJA2076\Home (67)) | SQLQuery1.sql - D...TJA2076\Home (70))

```sql
USE CompanyDB;
GO
-- Представления:
-- Представление для отображения полного имени сотрудника и его отдела:
CREATE VIEW EmployeeDepartmentView AS
SELECT e.firstname + ' ' + e.surname + ' ' + e.lastname AS FullName, d.department_name
FROM employee e
JOIN employee_department ed ON e.id = ed.id_employee
JOIN department d ON ed.id_department = d.id;
GO
SELECT * FROM EmployeeDepartmentView;
GO
```

118 %

Results | Messages

	FullName	department_name
1	Ivan Ivanovich Ivanov	HR
2	Petr Petrovich Petrov	HR
3	Sergey Sergeevich Sidorov	Engineering
4	Alexey Alexeevich Smirnov	Engineering
5	Dmitry Dmitrievich Popov	Sales
6	Vasily Vasilievich Vasiliev	Sales
7	Zahar Zakharevich Zaitsev	Marketing
8	Konstantin Konstantinovich Kuznetsov	Marketing
9	Maxim Maximovich Morozov	Finance
10	Pavel Pavlovich Pavlov	Finance
11	Artem Artemovich Lobov	Legal
12	Igor Igorevich Sokolov	Quality Assurance

SQLQuery3.sql - D...TJA2076\Home (72)) | SQLQuery2.sql - D...TJA2076\Home (67)) | SQLQuery1.sql

```sql

-- Представление для отображения проектов и статуса их оплаты:
CREATE VIEW ProjectPaymentStatusView AS
SELECT p.project_name, p.project_status, p.payment_status
FROM project p;
GO
SELECT * FROM ProjectPaymentStatusView;
GO
```

118 %

Results | Messages

	project_name	project_status	payment_status
1	Project Alpha	Active	Paid
2	Project Beta	On Hold	Unpaid
3	Project Gamma	Active	Partial
4	Project Delta	Completed	Paid
5	Project Epsilon	Active	Unpaid
6	Project Zeta	On Hold	Partial
7	Project Eta	Active	Paid
8	Project Theta	Completed	Unpaid
9	Project Iota	On Hold	Paid
10	Project Kappa	Active	Partial
11	Project Lambda	Active	Unpaid
12	Project Mu	Active	Partial

SQLQuery3.sql - D...TJA2076\Home (72)) | SQLQuery2.sql - D...TJA2076\Home (67)) | SQLQuery1.sql - D...TJA2076\H

```sql
-- Представление для списка задач с контрольными датами и комментариями:
CREATE VIEW TaskStatusView AS
SELECT t.task_number, ts.control_date, ts.comment
FROM task t
JOIN task_status ts ON t.task_status_id = ts.id;
GO
SELECT * FROM TaskStatusView;
GO
```

118 %

Results | Messages

	task_number	control_date	comment
1	T001	2023-01-01	Started
2	T002	2023-02-01	In Progress
3	T003	2023-03-01	Completed
4	T004	2023-04-01	On Hold
5	T005	2023-05-01	Pending Review
6	T006	2023-06-01	Needs Revision
7	T007	2023-07-01	Approved
8	T008	2023-08-01	Rejected
9	T009	2023-09-01	Scheduled
10	T010	2023-10-01	Delayed
11	T011	2023-01-01	Started
12	T012	2023-02-01	In Progress

SQLQuery3.sql - D...TJA2076\Home (72)) | SQLQuery2.sql - D...TJA2076\Home (67)) | SQLQuery1.sql - D...

```sql

-- Представление для информации о клиентах с контактными данными:
CREATE VIEW CustomerContactInfoView AS
SELECT company_name, contact_person, telephone, email
FROM customer;
GO
SELECT * FROM CustomerContactInfoView;
GO
```

118 %

Results | Messages

	company_name	contact_person	telephone	email
1	TechCorp	John Doe	+1234567892	johndoe@techcorp.com
2	BuildInc	Jane Smith	+1234567893	janesmith@buildinc.com
3	SoftSolutions	Bill Gates	+1234567894	bill@softsolutions.com
4	NetTech	Elon Musk	+1234567895	elon@nettech.com
5	EcoPower	Greta Thunberg	+1234567896	greta@ecopower.com
6	HealthFirst	Ben Carson	+1234567897	ben@healthfirst.com
7	FoodDelight	Gordon Ramsay	+1234567898	gordon@fooddelight.com
8	AutoDrive	Jeremy Clarkson	+1234567899	jeremy@autodrive.com
9	MobiTech	Steve Jobs	+1234567800	steve@mobitech.com
10	EduLearn	Albert Einstein	+1234567801	albert@edulearn.com
11	FashionTech	Donatella Versace	+1234567804	donatella@fashiontech.com
12	SpaceXplore	Neil Armstrong	+1234567805	neil@spacexplore.com

SQLQuery3.sql - D...TJA2076\Home (72)) | SQLQuery2.sql - D...TJA2076\Home (67)) | SQLQuery1.sc

```sql
-- Представление для отображения средней зарплаты по отделам:
CREATE VIEW AverageSalaryByDepartmentView AS
SELECT d.department_name, AVG(e.salary) AS AverageSalary
FROM employee e
JOIN employee_department ed ON e.id = ed.id_employee
JOIN department d ON ed.id_department = d.id
GROUP BY d.department_name;
GO
SELECT * FROM AverageSalaryByDepartmentView;
```

118 %

Results | Messages

	department_name	AverageSalary
1	Engineering	46000.000000
2	Finance	46500.000000
3	HR	45000.000000
4	Legal	55000.000000
5	Marketing	52000.000000
6	Quality Assurance	50000.000000
7	Sales	45000.000000

Задание 37. Покажите применение табличных переменных, временные локальных и глобальных таблиц, а также обобщенных табличных выражений.

SQLQuery4.sql - D...TJA2076\Home (56)) | SQLQuery3.sql - D...TJA2076\Home (72)) | SQLQuery2.sql - D...TJA2076\Home (67)) | SQLQuery1.sql - D

```sql
USE CompanyDB;
-- Применение табличных переменных
DECLARE @EmployeeTable TABLE (EmployeeID INT, EmployeeName NVARCHAR(100), Salary DECIMAL(15,2));
INSERT INTO @EmployeeTable (EmployeeID, EmployeeName, Salary)
SELECT id, firstname + ' ' + lastname, salary FROM employee;

SELECT * FROM @EmployeeTable WHERE Salary > 40000;
```

118 %

Results | Messages

	EmployeeID	EmployeeName	Salary
1	1	Ivan Ivanov	50000.00
2	3	Sergey Sidorov	45000.00
3	4	Alexey Smirnov	47000.00
4	5	Dmitry Popov	48000.00
5	6	Vasily Vasiliev	42000.00
6	7	Zahar Zaitsev	53000.00
7	8	Konstantin Kuznetsov	51000.00
8	9	Maxim Morozov	41000.00
9	10	Pavel Pavlov	52000.00
10	11	Artem Lobov	55000.00
11	12	Igor Sokolov	50000.00

SQLQuery5.sql - D...TJA2076\Home (79)) | SQLQuery4.sql - D...TJA2076\Home (56)) | SQLQuery3.sql - D...TJA207

```sql
USE CompanyDB;
-- Применение временных локальных таблиц
CREATE TABLE #LocalTempEmployee
(
    EmployeeID INT,
    EmployeeName NVARCHAR(100),
    Department NVARCHAR(50)
);

INSERT INTO #LocalTempEmployee (EmployeeID, EmployeeName, Department)
SELECT e.id, e.firstname + ' ' + e.lastname, d.department_name
FROM employee e
JOIN employee_department ed ON e.id = ed.id_employee
JOIN department d ON ed.id_department = d.id;

SELECT * FROM #LocalTempEmployee;

DROP TABLE #LocalTempEmployee;
```

118 %

Results | Messages

	EmployeeID	EmployeeName	Department
1	1	Ivan Ivanov	HR
2	2	Petr Petrov	HR
3	3	Sergey Sidorov	Engineering
4	4	Alexey Smirnov	Engineering
5	5	Dmitry Popov	Sales
6	6	Vasily Vasiliev	Sales
7	7	Zahar Zaitsev	Marketing
8	8	Konstantin Kuznetsov	Marketing
9	9	Maxim Morozov	Finance
10	10	Pavel Pavlov	Finance
11	11	Artem Lobov	Legal
12	12	Igor Sokolov	Quality Assurance

SQLQuery6.sql - D...TJA2076\Home (80)) | SQLQuery5.sql - D...TJA2076\Home (79)) | SQLQuery4.sql - D...TJA2076\H

```sql
USE CompanyDB;
-- Применение временных глобальных таблиц
CREATE TABLE ##GlobalTempEmployee
(
    EmployeeID INT,
    EmployeeName NVARCHAR(100),
    Department NVARCHAR(50)
);

INSERT INTO ##GlobalTempEmployee (EmployeeID, EmployeeName, Department)
SELECT e.id, e.firstname + ' ' + e.lastname, d.department_name
FROM employee e
JOIN employee_department ed ON e.id = ed.id_employee
JOIN department d ON ed.id_department = d.id;

SELECT * FROM ##GlobalTempEmployee;
-- Эту таблицу можно использовать в других сессиях SQL.
DROP TABLE ##GlobalTempEmployee;
```

118 %

Results | Messages

	EmployeeID	EmployeeName	Department
1	1	Ivan Ivanov	HR
2	2	Petr Petrov	HR
3	3	Sergey Sidorov	Engineering
4	4	Alexey Smirnov	Engineering
5	5	Dmitry Popov	Sales
6	6	Vasily Vasiliev	Sales
7	7	Zahar Zaitsev	Marketing
8	8	Konstantin Kuznetsov	Marketing
9	9	Maxim Morozov	Finance
10	10	Pavel Pavlov	Finance
11	11	Artem Lobov	Legal
12	12	Igor Sokolov	Quality Assurance

SQLQuery7.sql - D...TJA2076\Home (60)) | SQLQuery6.sql - D...TJA2076\Home (80)) | SQLQuery5.sql

```sql
USE CompanyDB;
-- Применение обобщенных табличных выражений (CTE)
WITH EmployeeCTE AS
(
    SELECT id, firstname + ' ' + lastname AS FullName, salary
    FROM employee
    WHERE salary > 45000
)
SELECT FullName, salary FROM EmployeeCTE;
```

118 %

Results | Messages

	FullName	salary
1	Ivan Ivanov	50000.00
2	Alexey Smirnov	47000.00
3	Dmitry Popov	48000.00
4	Zahar Zaitsev	53000.00
5	Konstantin Kuznetsov	51000.00
6	Pavel Pavlov	52000.00
7	Artem Lobov	55000.00
8	Igor Sokolov	50000.00

Задание 38. Создать 2 хранимые процедуры, одна из которых будет иметь выходные параметры и показать их применение.

SQLQuery9.sql - D...TJA2076\Home (68)) | SQLQuery8.sql - D...TJA2076\Home (51)) | SQLQuery7.sql - D...TJA2076\Hom

```sql
USE CompanyDB;
GO
-- Хранимая процедура 1: Информация о сотрудниках в отделе
-- Эта процедура принимает номер отдела в качестве
-- входного параметра и возвращает информацию о сотрудниках этого отдела.
CREATE PROCEDURE GetEmployeesByDepartment
    @DepartmentID INT
AS
BEGIN
    SELECT e.id, e.firstname, e.lastname, e.position, d.department_name
    FROM employee e
    INNER JOIN employee_department ed ON e.id = ed.id_employee
    INNER JOIN department d ON ed.id_department = d.id
    WHERE d.id = @DepartmentID;
END;
GO
EXEC GetEmployeesByDepartment @DepartmentID = 3;
```

118 %

Results | Messages

	id	firstname	lastname	position	department_name
1	5	Dmitry	Popov	Finance Analyst	Sales
2	6	Vasily	Vasiliev	IT Support	Sales

SQLQuery9.sql - D...TJA2076\Home (68)) | SQLQuery8.sql - D...TJA2076\Home (51)) | SQLQuery7.sql - D...TJA2076\Home (60)) | SQLQuery6.sql - D...TJA2076\H

```sql
USE CompanyDB;
GO
-- Хранимая процедура 2: Расчет средней зарплаты сотрудников
-- Эта процедура вычисляет среднюю зарплату сотрудников и возвращает результат через выходной параметр.
CREATE PROCEDURE GetAverageSalary
    @AverageSalary DECIMAL(15,2) OUTPUT
AS
BEGIN
    SELECT @AverageSalary = AVG(salary)
    FROM employee;
END;
GO
DECLARE @AvgSalary DECIMAL(15,2);
EXEC GetAverageSalary @AverageSalary = @AvgSalary OUTPUT;
SELECT @AvgSalary AS AverageSalary;
```

118 %

Results | Messages

	AverageSalary
1	47833.33

Задание 39. Создать 2 определяемые пользователем функции, одна из которых скалярная функция, другая возвращая табличное значение и показать их применение.

SQLQuery10.sql -...TJA2076\Home (77)) | SQLQuery9.sql - D...TJA2076\Home (68)) | SQLQuery8.sql - D...TJA2076\Home (51))

```sql
USE CompanyDB;
GO
-- Скалярная функция: Получение полного имени сотрудника
-- Эта функция возвращаеет полное имя сотрудника (имя, отчество, фамилия),
-- основываясь на его идентификаторе.
CREATE FUNCTION GetFullName (@EmployeeID INT)
RETURNS NVARCHAR(150)
AS
BEGIN
    DECLARE @FullName NVARCHAR(150);

    SELECT @FullName = firstname + ' ' + ISNULL(surname + ' ', '') + lastname
    FROM employee
    WHERE id = @EmployeeID;

    RETURN @FullName;
END;
GO
SELECT dbo.GetFullName(1) AS FullName;
```

118 %

Results | Messages

	FullName
1	Ivan Ivanovich Ivanov

SQLQuery11.sql -...TJA2076\Home (85)) | SQLQuery10.sql -...TJA2076\Home (77)) | SQLQuery9.sql - D...TJA2076\Home (68))

```sql
USE CompanyDB;
GO
-- Функция, возвращающая табличное значение: Список задач сотрудника
-- Эта функция возвращает список задач, назначенных на определенного сотрудника.
CREATE FUNCTION GetTasksForEmployee (@EmployeeID INT)
RETURNS TABLE
AS
RETURN
    SELECT t.task_number, t.start_date, t.deadline, t.reward
    FROM task t
    INNER JOIN employee_task et ON t.id = et.task_id
    WHERE et.employee_id = @EmployeeID;
GO
SELECT * FROM dbo.GetTasksForEmployee(2);
```

118 %

Results | Messages

	task_number	start_date	deadline	reward
1	T005	2023-02-05	2023-04-20	1300.00
2	T006	2023-04-01	2023-05-20	1400.00

Задание 40. Создайте два триггера: триггер AFTER и триггер INSTEAD OF и показать их применение

SQLQuery12.sql -...TJA2076\Home (86)) | SQLQuery11.sql -...TJA2076\Home (85)) | SQLQuery10.sql -...TJA2076\Home (77

```sql
USE CompanyDB;
GO
-- Триггер AFTER: Логирование изменений зарплаты сотрудников
-- Этот триггер активируется после каждого обновления зарплаты сотрудника.
-- Он записывает информацию об изменении в специальную таблицу логов.
CREATE TABLE SalaryAudit (
    AuditID INT PRIMARY KEY IDENTITY(1,1),
    EmployeeID INT,
    OldSalary DECIMAL(15,2),
    NewSalary DECIMAL(15,2),
    UpdateDate DATETIME
);
GO
```

SQLQuery12.sql -...TJA2076\Home (86)) | SQLQuery11.sql -...TJA2076\Home (85)) | SQLQuery10.sql -...TJA2076\Home (77))

```sql
CREATE TRIGGER trg_AfterSalaryUpdate
ON employee
AFTER UPDATE
AS
BEGIN
    IF (UPDATE(salary))
    BEGIN
        INSERT INTO SalaryAudit (EmployeeID, OldSalary, NewSalary, UpdateDate)
        SELECT i.id, d.salary, i.salary, GETDATE()
        FROM inserted i
        INNER JOIN deleted d ON i.id = d.id;
    END
END;
GO
```

```sql
SELECT * FROM SalaryAudit;
UPDATE employee
SET salary = 55000
WHERE id = 1;
SELECT * FROM SalaryAudit;
GO
```

118 %

Results | Messages

AuditID	EmployeeID	OldSalary	NewSalary	UpdateDate

	AuditID	EmployeeID	OldSalary	NewSalary	UpdateDate
1	1	1	50000.00	55000.00	2023-11-17 01:02:40.520

SQLQuery13.sql -...TJA2076\Home (65)) | SQLQuery12.sql -...TJA2076\Home (86)) | SQLQuery11.sql -...TJA2076\Home (85)) | SQLQuery10.sql -...TJA2076\Home (77))

```sql
USE CompanyDB;
GO
-- Триггер INSTEAD OF: Проверка условий перед удалением проекта
-- Этот триггер будет активироваться вместо удаления записи из таблицы project.
-- Он проверит, есть ли невыполненные задачи по проекту, и если они есть, удаление не произойдет.
CREATE TRIGGER trg_InsteadOfDeleteProject
ON project
INSTEAD OF DELETE
AS
BEGIN
    DECLARE @ProjectID INT;

    SELECT @ProjectID = id FROM deleted;

    IF NOT EXISTS (SELECT * FROM task WHERE project_id = @ProjectID AND task_status_id <> 3) -- 3 - статус "Завершено"
    BEGIN
        DELETE FROM project WHERE id = @ProjectID;
    END
    ELSE
    BEGIN
        RAISERROR ('Невозможно удалить проект, так как есть незавершенные задачи', 16, 1);
    END
END;
GO
DELETE FROM project WHERE id = 4;
```

118 %

Messages

```
Msg 50000, Level 16, State 1, Procedure trg_InsteadOfDeleteProject, Line 19 [Batch Start Line 24]
Невозможно удалить проект, так как есть незавершенные задачи

(1 row affected)
```