<!--Схема БД "Проектная организация"

Таблицы:

Отделы (Departments): id\_department, name

Сотрудники (Employees): id\_employee, first\_name, last\_name, date\_of\_birth, salary, gender, id\_position, id\_department

Должности (Positions): id\_position, name, salary

Проекты (Projects): id\_project, name, id\_manager (FK to Employees)

Участие в проектах (Employee\_Project): id\_employee (FK to Employees), id\_project (FK to Projects)

Дети сотрудников (Employees\_Children): id\_employee (FK to Employees), child\_name

SQL-запросы для БД "Проектная организация"

Сотрудники 2,6,7 отделов, родившиеся в 80-е годы прошлого века.

sql

SELECT first\_name, last\_name, date\_of\_birth

FROM Employees

WHERE id\_department IN (2, 6, 7)

AND YEAR(date\_of\_birth) BETWEEN 1980 AND 1989;

Программисты, у которых оклад после уплаты подоходного налога не менее 40000 рублей.

sql

SELECT first\_name, last\_name, salary

FROM Employees

WHERE id\_position = (SELECT id\_position FROM Positions WHERE name = 'Программист')

AND salary \* 0.85 >= 40000;

Фамилия всех начальников отделов.

sql

SELECT last\_name

FROM Employees

WHERE id\_employee IN (SELECT id\_manager FROM Departments);

Отделы, в которых работают экономисты.

sql

SELECT name

FROM Departments

WHERE id\_department IN (SELECT id\_department FROM Employees WHERE id\_position = (SELECT id\_position FROM Positions WHERE name = 'Экономист'));

Перечень должностей с зарплатами.

sql

SELECT name, salary

FROM Positions;

Количество сотрудниц организации.

sql

SELECT COUNT(\*)

FROM Employees

WHERE gender = 'женщина';

Минимальный и максимальный оклад в отделах 1 и 2.

sql

SELECT id\_department, MIN(salary) AS min\_salary, MAX(salary) AS max\_salary

FROM Employees

WHERE id\_department IN (1, 2)

GROUP BY id\_department;

Количество сотрудников 6-го отдела.

sql

SELECT COUNT(\*)

FROM Employees

WHERE id\_department = 6;

Минимальный и максимальный оклады по отделам и разница между ними.

sql

SELECT id\_department, MIN(salary) AS min\_salary, MAX(salary) AS max\_salary, MAX(salary) - MIN(salary) AS salary\_diff

FROM Employees

GROUP BY id\_department;

Минимальный и максимальный оклады по каждой должности.

sql

SELECT id\_position, MIN(salary) AS min\_salary, MAX(salary) AS max\_salary

FROM Employees

GROUP BY id\_position;

Отдел, должность, средний оклад.

sql

SELECT d.name AS department, p.name AS position, AVG(e.salary) AS avg\_salary

FROM Employees e

JOIN Departments d ON e.id\_department = d.id\_department

JOIN Positions p ON e.id\_position = p.id\_position

GROUP BY d.name, p.name;

Отделы, в которых количество сотрудников меньше 4-х.

sql

SELECT name

FROM Departments

WHERE id\_department IN (SELECT id\_department FROM Employees GROUP BY id\_department HAVING COUNT(\*) < 4);

Отделы, в которых минимальный возраст сотрудников меньше 20 лет.

sql

SELECT name

FROM Departments

WHERE id\_department IN (SELECT id\_department FROM Employees WHERE TIMESTAMPDIFF(YEAR, date\_of\_birth, CURDATE()) < 20 GROUP BY id\_department);

Отделы, в которых средний возраст сотрудников больше сорока лет.

sql

SELECT name

FROM Departments

WHERE id\_department IN (SELECT id\_department FROM Employees GROUP BY id\_department HAVING AVG(TIMESTAMPDIFF(YEAR, date\_of\_birth, CURDATE())) > 40);

Участники проектов с указанием названий проектов.

sql

SELECT e.first\_name, e.last\_name, p.name AS project\_name

FROM Employees e

JOIN Employee\_Project ep ON e.id\_employee = ep.id\_employee

JOIN Projects p ON ep.id\_project = p.id\_project;

Сотрудники с перечнем детей.

sql

SELECT e.first\_name, e.last\_name, ec.child\_name

FROM Employees e

LEFT JOIN Employees\_Children ec ON e.id\_employee = ec.id\_employee;

Название отдела, ФИО и должность сотрудника, имя ребенка.

sql

SELECT d.name AS department, e.first\_name, e.last\_name, p.name AS position, ec.child\_name

FROM Employees e

JOIN Departments d ON e.id\_department = d.id\_department

JOIN Positions p ON e.id\_position = p.id\_position

LEFT JOIN Employees\_Children ec ON e.id\_employee = ec.id\_employee;

Сотрудники, у которых есть дети.

sql

SELECT DISTINCT e.first\_name, e.last\_name

FROM Employees e

JOIN Employees\_Children ec ON e.id\_employee = ec.id\_employee;

Сотрудники 1-го и 5-го отделов, у которых есть дети.

sql

SELECT e.first\_name, e.last\_name

FROM Employees e

WHERE e.id\_department IN (1, 5)

AND e.id\_employee IN (SELECT id\_employee FROM Employees\_Children);

Сотрудники с указанием количества детей.

sql

SELECT e.first\_name, e.last\_name, COUNT(ec.child\_name) AS num\_children

FROM Employees e

LEFT JOIN Employees\_Children ec ON e.id\_employee = ec.id\_employee

GROUP BY e.id\_employee;

Проекты с указанием руководителей.

sql

SELECT p.name AS project\_name, e.first\_name, e.last\_name

FROM Projects p

JOIN Employees e ON p.id\_manager = e.id\_employee;

Сотрудники с указанием количества проектов, в которых они участвуют.

sql

SELECT e.first\_name, e.last\_name, COUNT(ep.id\_project) AS num\_projects

FROM Employees e

LEFT JOIN Employee\_Project ep ON e.id\_employee = ep.id\_employee

GROUP BY e.id\_employee;

Сотрудники, участвующие в нескольких проектах.

sql

SELECT e.first\_name, e.last\_name

FROM Employees e

JOIN Employee\_Project ep ON e.id\_employee = ep.id\_employee

GROUP BY e.id\_employee

HAVING COUNT(ep.id\_project) > 1;

Сотрудники, которые являются руководителями проектов.

sql

SELECT e.first\_name, e.last\_name

FROM Employees e

WHERE e.id\_employee IN (SELECT id\_manager FROM Projects);

Название проекта и количество участников.

sql

SELECT p.name AS project\_name, COUNT(ep.id\_employee) AS num\_participants

FROM Projects p

LEFT JOIN Employee\_Project ep ON p.id\_project = ep.id\_project

GROUP BY p.id\_project;

Проекты, в которых участвуют более 10 сотрудников.

sql

SELECT p.name AS project\_name

FROM Projects p

JOIN Employee\_Project ep ON p.id\_project = ep.id\_project

GROUP BY p.id\_project

HAVING COUNT(ep.id\_employee) > 10;

Сотрудники, участвующие в одном проекте в разных ролях.

sql

SELECT e.first\_name, e.last\_name, p.name AS project\_name

FROM Employees e

JOIN Employee\_Project ep ON e.id\_employee = ep.id\_employee

JOIN Projects p ON ep.id\_project = p.id\_project

GROUP BY e.id\_employee, p.id\_project

HAVING COUNT(DISTINCT ep.id\_project) > 1;

Отделы, в которых работают только мужчины или только женщины.

sql

SELECT d.name

FROM Departments d

JOIN Employees e ON e.id\_department = d.id\_department

GROUP BY d.id\_department

HAVING COUNT(DISTINCT e.gender) = 1;

Братьев, у которых есть старшие сестры.

sql

SELECT e1.first\_name, e1.last\_name

FROM Employees e1

JOIN Employees\_Children ec1 ON e1.id\_employee = ec1.id\_employee

JOIN Employees\_Children ec2 ON e1.id\_employee = ec2.id\_employee

WHERE e1.gender = 'мужчина'

AND ec1.child\_name IN (SELECT child\_name FROM Employees\_Children WHERE gender = 'женщина')

AND ec2.child\_name != ec1.child\_name;

Сотрудники, у которых есть разнополые дети.

sql

SELECT DISTINCT e.first\_name, e.last\_name

FROM Employees e

JOIN Employees\_Children ec ON e.id\_employee = ec.id\_employee

JOIN Employees\_Children ec2 ON e.id\_employee = ec2.id\_employee

WHERE ec.gender != ec2.gender;

Бездетные сотрудники.

sql

SELECT e.first\_name, e.last\_name

FROM Employees e

LEFT JOIN Employees\_Children ec ON e.id\_employee = ec.id\_employee

WHERE ec.id\_employee IS NULL;

Сотрудники, у которых оклад меньше, чем средний по каждому отделу.

sql

SELECT e.first\_name, e.last\_name, e.salary, e.id\_department

FROM Employees e

WHERE e.salary < (SELECT AVG(salary) FROM Employees WHERE id\_department = e.id\_department);

Проекты, у которых нет руководителя.

sql

SELECT name

FROM Projects

WHERE id\_manager IS NULL;

Сотрудники, получающие максимальный оклад в своем отделе.

sql

SELECT e.first\_name, e.last\_name, e.salary

FROM Employees e

WHERE e.salary = (SELECT MAX(salary) FROM Employees WHERE id\_department = e.id\_department);

Название проекта, ФИО руководителя, количество исполнителей, количество консультантов.

sql

SELECT p.name AS project\_name,

e.first\_name AS manager\_first\_name, e.last\_name AS manager\_last\_name,

COUNT(ep.id\_employee) AS num\_employees,

COUNT(DISTINCT ec.id\_employee) AS num\_consultants

FROM Projects p

LEFT JOIN Employees e ON p.id\_manager = e.id\_employee

LEFT JOIN Employee\_Project ep ON p.id\_project = ep.id\_project

LEFT JOIN Employee\_Project ec ON ep.id\_employee = ec.id\_employee

GROUP BY p.id\_project;

Схема БД "Каталог книг"

Таблицы:

Авторы (Authors): id\_author, first\_name, last\_name

Произведения (Works): id\_work, title

Авторства (Works\_Authors): id\_author, id\_work

SQL-запросы для БД "Каталог книг"

Произведения, у которых нет авторов.

sql

SELECT title

FROM Works

WHERE id\_work NOT IN (SELECT id\_work FROM Works\_Authors);

Авторы, у которых есть соавторы.

sql

SELECT DISTINCT a.first\_name, a.last\_name

FROM Authors a

JOIN Works\_Authors wa ON a.id\_author = wa.id\_author

JOIN Works\_Authors wa2 ON wa.id\_work = wa2.id\_work AND wa2.id\_author != a.id\_author;

Произведения, у которых более одного автора.

sql

SELECT w.title

FROM Works w

JOIN Works\_Authors wa ON w.id\_work = wa.id\_work

GROUP BY w.id\_work

HAVING COUNT(wa.id\_author) > 1;

Авторы, которые хотя бы одно произведение написали без соавторов.

sql

SELECT DISTINCT a.first\_name, a.last\_name

FROM Authors a

JOIN Works\_Authors wa ON a.id\_author = wa.id\_author

GROUP BY a.id\_author, wa.id\_work

HAVING COUNT(wa.id\_author) = 1;

Авторы, которые все произведения писали без соавторов.

sql

SELECT a.first\_name, a.last\_name

FROM Authors a

JOIN Works\_Authors wa ON a.id\_author = wa.id\_author

GROUP BY a.id\_author

HAVING COUNT(DISTINCT wa.id\_work) = COUNT(wa.id\_author);

Схема БД "Больница"

Таблицы:

Пациенты (Patients): id\_patient, name, gender

Лечение (Treatment): id\_patient, id\_doctor, diagnosis

Палаты (Wards): id\_ward, id\_department, capacity

Врачи (Doctors): id\_doctor, name, specialization, id\_department

Отделения (Departments): id\_department, name

SQL-запросы для БД "Больница"

Список всех пациентов, которые в настоящее время лежат в больнице.

sql

SELECT p.name

FROM Patients p

JOIN Treatment t ON p.id\_patient = t.id\_patient

WHERE t.diagnosis IS NOT NULL;

Список диагнозов пациентов, лечащими врачами которых являются хирурги.

sql

SELECT p.name AS patient\_name, t.diagnosis

FROM Patients p

JOIN Treatment t ON p.id\_patient = t.id\_patient

JOIN Doctors d ON t.id\_doctor = d.id\_doctor

WHERE d.specialization = 'Хирург';

Список всех врачей больницы с указанием отделения и специализации:

sql

SELECT d.name AS doctor\_name, d.specialization, dep.name AS department

FROM Doctors d

JOIN Departments dep ON d.id\_department = dep.id\_department;

Список пациентов по отделениям.

sql

SELECT dep.name AS department\_name, p.name AS patient\_name

FROM Patients p

JOIN Treatment t ON p.id\_patient = t.id\_patient

JOIN Doctors d ON t.id\_doctor = d.id\_doctor

JOIN Departments dep ON d.id\_department = dep.id\_department;

Количество пациентов по палатам.

sql

SELECT w.id\_ward, COUNT(p.id\_patient) AS num\_patients

FROM Wards w

LEFT JOIN Treatment t ON w.id\_ward = t.id\_ward

LEFT JOIN Patients p ON t.id\_patient = p.id\_patient

GROUP BY w.id\_ward;-->