

Правительство Российской Федерации
Федеральное государственное автономное образовательное учреждение
высшего профессионального образования
**"Национальный исследовательский университет
Высшая школа экономики"**
Департамент прикладной математики, бакалавр

ЛАБОРАТОРНАЯ РАБОТА:

По теме: ПРОЕКТИРОВАНИЕ БАЗЫ ДАННЫХ.

Выполнил:

Колодин Матвей Алексеевич

Преподаватель:

Шульгин Михаил Алексеевич

Москва, 2024

Содержание

1	Описание таблиц БД	3
2	Реализация запросов на языке реляционной алгебры	4

1 Описание таблиц БД

Таблица 1: persons

Имя атрибута	Тип атрибута	Ограничение	Описание атрибута
id	serial	NOT NULL, Primary key	id персоны
name	character varying	NOT NULL, Length = 128	Имя персоны
surname	character varying	NOT NULL, Length = 128	Фамилия персоны
patronymic	character varying	NOT NULL, Length = 128	Отчество персоны

Таблица 2: plots

Имя атрибута	Тип атрибута	Ограничение	Описание атрибута
id	integer	NOT NULL, Primary key	id участка
status_id	smallint	NOT NULL	id статуса участка

Таблица 3: plots_status

Имя атрибута	Тип атрибута	Ограничение	Описание атрибута
id	smallint	NOT NULL, Primary key	id статуса участка
status	character varying	NOT NULL, Length = 16	Наименование статуса

Таблица 4: ownership

Имя атрибута	Тип атрибута	Ограничение	Описание атрибута
person_id	integer	NOT NULL	id персоны
plot_id	integer	NOT NULL	id участка
ownership_id	integer	NOT NULL	id типа права собственности
start_period_id	integer	NOT NULL	id периода начала владения правами на собственность
end_period_id	integer	NOT NULL	id периода окончания владения правами на собственность

Таблица 5: ownership_type

Имя атрибута	Тип атрибута	Ограничение	Описание атрибута
id	serial	NOT NULL, Primary key	id типа права собственности
type	integer	NOT NULL	Наименование типа права собственности

Таблица 6: expenses

Имя атрибута	Тип атрибута	Ограничение	Описание атрибута
id	serial	NOT NULL, Primary key	id статьи расхода
name	character varying	NOT NULL, Length = 256	Наименование статьи расхода

Таблица 7: incomes

Имя атрибута	Тип атрибута	Ограничение	Описание атрибута
id	serial	NOT NULL, Primary key	id статьи дохода
name	character varying	NOT NULL, Length = 256	Наименование статьи дохода

Таблица 8: periods

Имя атрибута	Тип атрибута	Ограничение	Описание атрибута
id	serial	NOT NULL, Primary key	id периода
month	smallint	NOT NULL	Месяц периода
year	integer	NOT NULL	Год периода

Таблица 9: plot_expenses

Имя атрибута	Тип атрибута	Ограничение	Описание атрибута
id	uuid	NOT NULL, Primary key	id транзакции расхода в товариществе
period_id	integer	NOT NULL	id периода
plot_id	integer	NOT NULL	id участка
expense_id	integer	NOT NULL	id статьи расхода
value	double precision	NOT NULL	Величина расхода

Таблица 10: partnership_income

Имя атрибута	Тип атрибута	Ограничение	Описание атрибута
id	uuid	NOT NULL, Primary key	id транзакции взноса в товарищество
persona_id	integer	NOT NULL	id персоны
income_id	integer	NOT NULL	id статьи дохода
period_id	integer	NOT NULL	id периода
plot_id	integer	-	id связанного участка (при наличии)
value	double precision	NOT NULL	Величина дохода

2 Реализация запросов на языке реляционной алгебры

Я немного уже по привычке к SQL, поэтому буду писать сначала код на нем, а потом уже формализовать языком реляционной алгебры. В самом же деле, просто боюсь допустить ошибку при такой формализации, и выписывая SQL код пытаюсь показать правильность рассуждений.

Для рассмотрим саму базу на ERD-схеме:

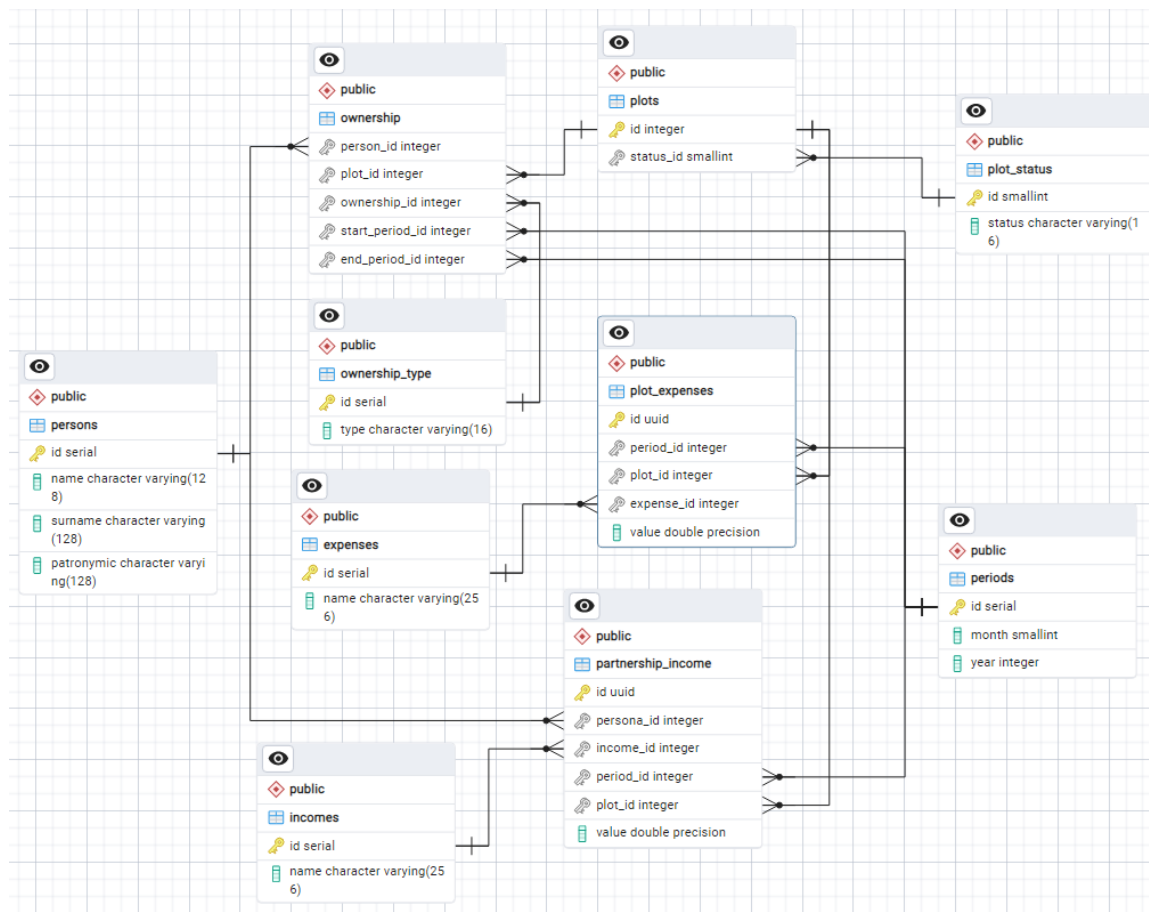


Рис. 1: ERD-схема базы данных

Теперь перейдем к составлению отчетов!

1. Найти всех владельцев, являющихся собственниками более чем одного участка.

- SQL:

```

SELECT
    o.person_id,
    p.name,
    p.surname,
    p.patronymic,
    COUNT(o.plot_id) as plot_count
FROM
    public.ownership o
JOIN
    public.persons p ON o.person_id = p.id
GROUP BY
    o.person_id, p.name, p.surname, p.patronymic
HAVING
    COUNT(o.plot_id) > 1;
  
```

- Реляционная алгебра:

$$\begin{aligned}
 &\pi_{\text{person_id}, \text{name}, \text{surname}, \text{patronymic}, \text{plot_count}} \\
 &\left(\sigma_{\text{plot_count} > 1} \right. \\
 &\left(\gamma_{\text{person_id}, \text{name}, \text{surname}, \text{patronymic}; \text{count}(\text{plot_id}) \rightarrow \text{plot_count}} \right. \\
 &\left(\pi_{\text{person_id}, \text{plot_id}, \text{name}, \text{surname}, \text{patronymic}} \right. \\
 &\left(\sigma_{\text{ownership.person_id} = \text{persons.id}} \right. \\
 &\left. (\text{ownership} \bowtie \text{persons}) \right)
 \end{aligned}$$

2. Найти все участки, приобретенные (владельцы с типом «покупатель») за заданный период.

- SQL:

```

SELECT
    o.plot_id,
    o.person_id,
    p.name,
    p.surname,
    p.patronymic,
    o.start_period_id,
    o.end_period_id
FROM
    public.ownership o
JOIN
    public.ownership_type ot ON o.ownership_id = ot.id
JOIN
    public.persons p ON o.person_id = p.id
WHERE
    ot.type = 'Покупатель'
    AND o.start_period_id >= 1    -- Ноябрь 2023
    AND o.end_period_id <= 5;    -- Март 2024

```

- Реляционная алгебра:

$$\begin{aligned}
 &\pi_{\text{plot_id}, \text{person_id}, \text{name}, \text{surname}, \text{patronymic}, \text{start_period_id}, \text{end_period_id}} \\
 &\left(\sigma_{\text{type} = \text{'Покупатель'} \wedge \text{start_period_id} \geq 1 \wedge \text{end_period_id} \leq 5} \right. \\
 &\left(\pi_{\text{plot_id}, \text{person_id}, \text{start_period_id}, \text{end_period_id}, \text{ownership_id}, \text{name}, \text{surname}, \text{patronymic}} \right. \\
 &\left(\sigma_{\text{ownership.ownership_id} = \text{ownership_type.id}} \right. \\
 &\left(\sigma_{\text{ownership.person_id} = \text{persons.id}} \right. \\
 &\left. (\text{ownership} \bowtie \text{ownership_type} \bowtie \text{persons}) \right)
 \end{aligned}$$

3. Найти все участки, принадлежащие владельцам с указанным типом на заданную дату.

- SQL:

```

SELECT
    o.plot_id,
    o.person_id,
ot.type AS ownership_type,
    p.name,
    p.surname,
    p.patronymic,
    o.start_period_id,
    o.end_period_id
FROM
    public.ownership o
JOIN
    public.ownership_type ot ON o.ownership_id = ot.id
JOIN
    public.persons p ON o.person_id = p.id
WHERE
    ot.type = 'Покупатель'    -- Можно заменить на требуемый тип персоны
    AND o.start_period_id >= 1    -- Можно заменить на требуемый начальный
    AND o.end_period_id <= 5;    -- Можно заменить на требуемый конечный пе

```

- Реляционная алгебра:

$$\begin{aligned} & \pi_{\text{plot_id}, \text{person_id}, \text{ownership_type}, \text{name}, \text{surname}, \text{patronymic}, \text{start_period_id}, \text{end_period_id}} \\ & \left(\sigma_{\text{type} = \text{'Покупатель'} \wedge \text{start_period_id} \geq 1 \wedge \text{end_period_id} \leq 5} \right. \\ & \left(\pi_{\text{plot_id}, \text{person_id}, \text{ownership_type}, \text{name}, \text{surname}, \text{patronymic}, \text{start_period_id}, \text{end_period_id}} \right. \\ & \left(\sigma_{\text{ownership.ownership_id} = \text{ownership_type.id}} \right. \\ & \left(\sigma_{\text{ownership.person_id} = \text{persons.id}} \right. \\ & \left. \left(\text{ownership} \bowtie \text{ownership_type} \bowtie \text{persons} \right) \right) \right) \end{aligned}$$

4. Определить размер взноса с заданного участка на погашение расходов

- SQL:

```
SELECT
    p.id AS period_id,
    p.month,
    p.year,
    COALESCE(SUM(pi.value), 0) AS total_income,
    COALESCE(SUM(pe1.value), 0) AS total_expense_electricity,
    COALESCE(SUM(pe2.value), 0) AS total_expense_trash_removal,
    COALESCE(SUM(pe3.value), 0) AS total_expense_security,
    COALESCE(SUM(pe4.value), 0) AS total_expense_bank_service,
    COALESCE(SUM(pe5.value), 0) AS total_expense_office_supplies,
    COALESCE(SUM(pe6.value), 0) AS total_expense_internet,
    COALESCE(SUM(pe7.value), 0) AS total_expense_equipment_maintenance,
    COALESCE(SUM(pe8.value), 0) AS total_expense_salaries,
    COALESCE(SUM(pe9.value), 0) AS total_expense_rent,
    COALESCE(SUM(pe10.value), 0) AS total_expense_software_purchase
FROM
    public.periods p
LEFT JOIN
    public.partnership_income pi ON p.id = pi.period_id AND pi.plot_id = 1
LEFT JOIN
    public.plot_expenses pe1 ON p.id = pe1.period_id AND pe1.plot_id = 1
LEFT JOIN
    public.plot_expenses pe2 ON p.id = pe2.period_id AND pe2.plot_id = 1
LEFT JOIN
    public.plot_expenses pe3 ON p.id = pe3.period_id AND pe3.plot_id = 1
LEFT JOIN
    public.plot_expenses pe4 ON p.id = pe4.period_id AND pe4.plot_id = 1
LEFT JOIN
    public.plot_expenses pe5 ON p.id = pe5.period_id AND pe5.plot_id = 1
LEFT JOIN
```



```

    public.plot_expenses pe6 ON p.id = pe6.period_id AND pe6.plot_id = 1 A
LEFT JOIN
    public.plot_expenses pe7 ON p.id = pe7.period_id AND pe7.plot_id = 1 A
LEFT JOIN
    public.plot_expenses pe8 ON p.id = pe8.period_id AND pe8.plot_id = 1 A
LEFT JOIN
    public.plot_expenses pe9 ON p.id = pe9.period_id AND pe9.plot_id = 1 A
LEFT JOIN
    public.plot_expenses pe10 ON p.id = pe10.period_id AND pe10.plot_id =
WHERE
    p.id BETWEEN 1 AND 5 -- Можно заменить на требуемый начальный и конечн
GROUP BY
    p.id, p.month, p.year
ORDER BY
    p.year, p.month;

```

- Реляционная алгебра:
Сам я такое не потяну..

5. Определить самый популярный период внесения взносов

- SQL:

```

SELECT
    p.id AS period_id,
    p.month,
    p.year,
    COUNT(pi.id) AS contributions_count
FROM
    public.partnership_income pi
JOIN
    public.periods p ON pi.period_id = p.id
GROUP BY
    p.id, p.month, p.year
ORDER BY
    contributions_count DESC
LIMIT 1;

```

- Реляционная алгебра:

$$\begin{aligned} & \pi_{\text{period_id,month,year,contributions_count}} \\ & \left(\tau_{\text{contributions_count DESC}} \right. \\ & \left(\sigma_{\text{rank} \leq 1} \right. \\ & \left(\gamma_{\text{id,month,year;count(pi.id)} \rightarrow \text{contributions_count}} \right. \\ & \left(\pi_{\text{id,month,year,pi.id}} \right. \\ & \left(\sigma_{\text{pi.period_id=p.id}} \right. \\ & \left. \left. \left(\text{public.partnership_income} \bowtie \text{public.periods} \right) \right) \right) \end{aligned}$$

6. Вывести список всех статусов участка с указанием процента от общего количества участков в СНТ

- SQL:

```
SELECT
    ps.status,
    COUNT(pl.id) AS count,
    ROUND((COUNT(pl.id) * 100.0 / (SELECT COUNT(*) FROM public.plots)), 2)
FROM
    public.plot_status ps
LEFT JOIN
    public.plots pl ON ps.id = pl.status_id
GROUP BY
    ps.status;
```

- Реляционная алгебра:

```

 $\pi_{\text{status,count,percentage}}$ 
 $(\gamma_{\text{status;count(pl.id)} \rightarrow \text{count}, \text{ROUND}((\text{count(pl.id)} \times 100.0) / (\text{count}(\pi_*(\text{public.plots}))), 2) \rightarrow \text{percentage}}$ 
 $(\pi_{\text{status,pl.id}}$ 
 $(\sigma_{\text{ps.id=pl.status\_id}}$ 
 $(\text{public.plot\_status LEFT JOIN public.plots})$ 

```

7. Сформировать приходную ведомость, имеющую следующую структуру: “№ участка”, “ФИО владельца”, “Статья прихода”, “Сумма”

- SQL:

```

SELECT
    pl.id AS "№ участка",
    CONCAT(p.surname, ' ', p.name, ' ', p.patronymic) AS "ФИО владельца",
    COALESCE(SUM(CASE WHEN pi.income_id = 1 THEN pi.value ELSE 0 END), 0)
    COALESCE(SUM(CASE WHEN pi.income_id = 2 THEN pi.value ELSE 0 END), 0)
    COALESCE(SUM(CASE WHEN pi.income_id = 3 THEN pi.value ELSE 0 END), 0)
    COALESCE(SUM(CASE WHEN pi.income_id = 4 THEN pi.value ELSE 0 END), 0)
    COALESCE(SUM(CASE WHEN pi.income_id = 5 THEN pi.value ELSE 0 END), 0)
    COALESCE(SUM(CASE WHEN pi.income_id = 6 THEN pi.value ELSE 0 END), 0)
    COALESCE(SUM(CASE WHEN pi.income_id = 7 THEN pi.value ELSE 0 END), 0)
    COALESCE(SUM(CASE WHEN pi.income_id = 8 THEN pi.value ELSE 0 END), 0)
    COALESCE(SUM(CASE WHEN pi.income_id = 9 THEN pi.value ELSE 0 END), 0)
    COALESCE(SUM(CASE WHEN pi.income_id = 10 THEN pi.value ELSE 0 END), 0)
FROM
    public.plots pl
LEFT JOIN
    public.ownership o ON pl.id = o.plot_id
LEFT JOIN
    public.persons p ON o.person_id = p.id
LEFT JOIN
    public.partnership_income pi ON pl.id = pi.plot_id
GROUP BY
    pl.id, p.surname, p.name, p.patronymic
ORDER BY
    pl.id;

```

- Реляционная алгебра:

$$\begin{aligned}
 &\pi_{\text{№ участка, ФИО владельца, Членские взносы, Дополнительные взносы, Целевые взносы, Оплата за электроэнергию,}} \\
 &\left(\gamma_{\text{pl.id, p.surname, p.name, p.patronymic; } \sum(\sigma_{\text{pi.income_id=1}} \rightarrow \text{Членские взносы}), \sum(\sigma_{\text{pi.income_id=2}} \rightarrow \text{Дополнительные взносы}), \dots} \right. \\
 &\left(\pi_{\text{pl.id, p.surname, p.name, p.patronymic, pi.income_id, pi.value}} \right. \\
 &\left. \left(\sigma_{\text{pl.id=o.plot_id}} \left(\sigma_{\text{o.person_id=p.id}} \left(\text{public.plots} \bowtie \text{public.ownership} \bowtie \text{public.persons} \bowtie \text{public.partnership_income} \right) \right) \right) \right)
 \end{aligned}$$

8. Сформировать смету расходов в виде таблицы "Статья расхода" и "Сумма" для СНТ за заданный период

- SQL:

```

SELECT
    p.year,
    p.month,
    COALESCE(SUM(CASE WHEN pe.expense_id = 1 THEN pe.value ELSE 0 END), 0),
    COALESCE(SUM(CASE WHEN pe.expense_id = 2 THEN pe.value ELSE 0 END), 0),
    COALESCE(SUM(CASE WHEN pe.expense_id = 3 THEN pe.value ELSE 0 END), 0),
    COALESCE(SUM(CASE WHEN pe.expense_id = 4 THEN pe.value ELSE 0 END), 0),
    COALESCE(SUM(CASE WHEN pe.expense_id = 5 THEN pe.value ELSE 0 END), 0),
    COALESCE(SUM(CASE WHEN pe.expense_id = 6 THEN pe.value ELSE 0 END), 0),
    COALESCE(SUM(CASE WHEN pe.expense_id = 7 THEN pe.value ELSE 0 END), 0),
    COALESCE(SUM(CASE WHEN pe.expense_id = 8 THEN pe.value ELSE 0 END), 0),
    COALESCE(SUM(CASE WHEN pe.expense_id = 9 THEN pe.value ELSE 0 END), 0),
    COALESCE(SUM(CASE WHEN pe.expense_id = 10 THEN pe.value ELSE 0 END), 0),
    COALESCE(SUM(pe.value), 0) AS "Суммарные расходы"
FROM
    public.plot_expenses pe
JOIN
    public.periods p ON pe.period_id = p.id
WHERE
    p.id BETWEEN 1 AND 5 -- Можно заменить на требуемый начальный и конечный
GROUP BY
    p.year, p.month
ORDER BY
    p.year, p.month;

```

- Реляционная алгебра:

$\pi_{\text{year, month, Электроэнергия, Вывоз мусора, Охрана, Обслуживание банка, Канцелярские расходы, Интернет, Ремонт и ...}}$
 $\left(\gamma_{\text{year, month; } \sum(\sigma_{\text{pe.expense_id}=1} \rightarrow \text{Электроэнергия}), \sum(\sigma_{\text{pe.expense_id}=2} \rightarrow \text{Вывоз мусора}), \sum(\sigma_{\text{pe.expense_id}=3} \rightarrow \text{Охрана}), \dots} \right)$
 $\left(\pi_{\text{year, month, pe.expense_id, pe.value}} \right)$
 $\left(\sigma_{\text{p.id} \geq 1 \wedge \text{p.id} \leq 5} \right)$
 $(\text{public.plot_expenses} \bowtie \text{public.periods})$

$\sigma_{\text{p.year, p.month}}$