



МИНОБРНАУКИ РОССИИ

Федеральное государственное бюджетное образовательное учреждение
высшего образования
«МИРЭА – Российский технологический университет»

РТУ МИРЭА

**Институт информационных технологий (ИИТ)
Кафедра цифровой трансформации (ЦТ)**

ОТЧЕТ ПО ПРАКТИЧЕСКОЙ РАБОТЕ
по дисциплине «Разработка баз данных»

Практическое занятие №3

Студенты группы *ИКБО-20-23 Комисарик М.А.*

(подпись)

Ассистент *Брайловский А.В.*

(подпись)

Отчет представлен «___»____2025 г.

Москва 2025 г.

ПОСТАНОВКА ЗАДАЧИ

Цель работы: Работа направлена на формирование глубокого понимания и практического применения инструментов для реализации сложной бизнес логики непосредственно на уровне базы данных.

Постановка задачи:

Задание 1: использование оператора CASE.

Задание 2: использование подзапросов.

Задание 3: использование обобщенных табличных выражений (СТЕ).

ХОД РАБОТЫ

1 Начальные данные

На рисунке 1 представлены данные таблицы product.

	123 ↗ id_product	A-Z name	A-Z description	123 price
1	1	Margherita Pizza	Classic pizza with tomato and cheese	\$10.99
2	2	Pepperoni Pizza	Pizza with pepperoni slices	\$12.99
3	3	Cola	Refreshing soft drink	\$2.99
4	4	Veggie Supreme	Pizza with assorted vegetables	\$14.99
5	5	Chocolate Cake	Rich chocolate dessert	\$6.99
6	6	Caesar Salad	Fresh salad with Caesar dressing	\$8.99
7	7	Garlic Bread	Toasted bread with garlic butter	\$4.99

Рисунок 1 – Содержание таблицы product

На рисунке 2 представлены данные таблицы product_ingredient.

	123 ↗ id_product_ingredient	123 ↗ id_product	123 ↗ id_ingredient	123 ingredient_weight
1	1	1	1	500
2	2	1	2	200
3	3	1	3	150
4	4	2	1	500
5	5	2	2	200
6	6	2	3	150
7	7	2	4	100
8	8	4	1	500
9	9	4	2	200
10	10	4	3	150
11	11	4	5	80
12	12	4	6	70
13	13	4	7	60

Рисунок 2 – Содержание таблицы product_ingredient

На рисунке 3 представлены данные таблицы ingredient_supplier.

	123 ↗ id_ingredient_supplier	123 ↗ id_ingredient	123 ↗ id_supplier	
1	1	1	1	
2	2	2	2	
3	3	3	1	
4	4	4	4	
5	5	5	3	
6	6	6	3	
7	7	7	3	

Рисунок 3 – Содержание таблицы ingredient_supplier

На рисунке 4 представлены данные таблицы supplier.

	123 ↗ id_supplier	A-Z name	A-Z phone_number	123 ↗ id_address	
1	1	Fresh Ingredients Co.	555-1234	1	
2	2	Dairy Suppliers Ltd.	555-5678	2	
3	3	Produce Partners	555-9012	3	
4	4	Meat Masters	555-3456	4	
5	5	Bakery Basics	555-7890	5	

Рисунок 4 – Содержание таблицы supplier

На рисунке 5 представлены данные таблицы client.

	123 ↗ id_client	A-Z first_name	A-Z second_name	A-Z third_name	A-Z phone_number	123 ↗ id_delivery_address	A-Z account_password_id_hash	
1	1	John	Doe	[NULL]	555-1111	1	hashed_password_123	
2	2	Jane	Smith	Marie	555-2222	2	hashed_password_456	
3	3	Mike	Johnson	Robert	555-3333	3	hashed_password_789	
4	4	Sarah	Wilson	[NULL]	555-4444	4	hashed_password_101	
5	5	David	Brown	James	555-5555	5	hashed_password_112	
6	6	Emily	Davis	Anne	555-6666	6	hashed_password_131	
7	7	Chris	Miller	Thomas	555-7777	7	hashed_password_415	

Рисунок 5 – Содержание таблицы client

На рисунках 6-7 представлены данные таблицы employee.

	123 ↗ id_employee	123 ↗ id_job_position	A-Z first_name	A-Z second_name	A-Z third_name	A-Z app_account_password_hash	
1	1	1	Alice	Johnson	[NULL]	emp_hash_1	555
2	2	2	Bob	Williams	Lee	emp_hash_2	555
3	3	3	Carol	Martinez	[NULL]	emp_hash_3	555
4	4	4	Dave	Anderson	Paul	emp_hash_4	555
5	5	5	Eva	Garcia	Maria	emp_hash_5	555
6	6	2	Frank	Taylor	[NULL]	emp_hash_6	555
7	7	3	Grace	Thomas	Elizabeth	emp_hash_7	555

Рисунок 6 – Содержание таблицы employee, часть 1

A-Z phone_number	123 ↗ id_registration_address	⌚ employment_date	⌚ employment_contract_end_date	
555-3333	1	2023-01-15	2025-01-15	
555-4444	2	2023-02-20	2025-02-20	
555-8888	3	2023-03-10	2024-03-10	
555-9999	4	2023-04-05	2024-10-05	
555-0000	5	2023-05-12	2025-05-12	
555-1212	6	2023-06-18	2024-12-18	
555-1313	7	2023-07-22	2024-07-22	

Рисунок 7 – Содержание таблицы employee, часть 2

	123 ↗ category_id	A-Z name	123 ↘ parent_id
1	1	Электроника	[NULL]
2	2	Бытовая техника	[NULL]
3	3	Смартфоны	1
4	4	Ноутбуки	1
5	5	Холодильники	2
6	6	Аксессуары для смартфонов	3

Рисунок 8 – Содержание таблицы categories

2 Использование оператора CASE

2.1

The screenshot shows a database interface with a query editor and a results viewer.

```

    select
        p.id_product,
        p.ingredient_weight,
        case
            when p.ingredient_weight < 100 then 'Мало'
            when p.ingredient_weight < 300 then 'Средне'
            else 'Много'
        end as weight_grade
    from
        product_ingredient as p;
  
```

The results table is titled "product_ingredient 1" and contains the following data:

id_product	ingredient_weight	weight_grade
1	500	Много
1	200	Средне
1	150	Средне
2	500	Много
2	200	Средне
2	150	Средне
2	100	Средне
4	500	Много
4	200	Средне
4	150	Средне
4	80	Мало
4	70	Мало
4	60	Мало

Рисунок 9

2.2

```
with categorized_ingredients AS (
    select
        id_ingredient,
        case
            when ingredient_weight < 100 then 'Мало'
            when ingredient_weight < 300 then 'Средне'
            else 'Много'
        end as weight_grade
    from product_ingredient
)
select
    weight_grade,
    count(id_ingredient) as ingredient_count
from categorized_ingredients
group by weight_grade
order by MIN(
    case
        when weight_grade = 'Много' then 3
        when weight_grade = 'Средне' then 2
        else 1
    end
);
```

Результат 1

Таблица	weight_grade	ingredient_count
1	Мало	3
2	Средне	7
3	Много	3

Рисунок 10

3 Использование подзапросов

3.1 Скалярный подзапрос

```
select
    p.id_product,
    p.ingredient_weight
from
    product_ingredient as p
where
    p.ingredient_weight > (select AVG(p.ingredient_weight) from product_ingredient as p);
```

product_ingredient 1

Таблица	id_product	ingredient_weight
1	1	500
2	2	500
3	4	500

Рисунок 11

3.2 Многострочный подзапрос с IN

The screenshot shows a database interface with a code editor and a results viewer. The code editor contains the following SQL query:

```
select
    p.id_product,
    p.ingredient_weight
from
    product_ingredient as p
where
    p.ingredient_weight IN (
        select p.ingredient_weight
        from product_ingredient as p
        where p.ingredient_weight < 100
    );
```

The results viewer shows a table named 'product_ingredient 1' with two columns: 'id_product' and 'ingredient_weight'. The data is as follows:

	id_product	ingredient_weight
1	4	80
2	4	70
3	4	60

Рисунок 12

3.3 Коррелированный подзапрос с EXISTS

The screenshot shows a database interface with a code editor and a results viewer. The code editor contains the following SQL query:

```
select
    p.id_product
from
    product_ingredient as p
where exists (
    select 1
    from
        ingredient_supplier as i
    where
        i.id_ingredient_supplier = p.id_ingredient
);
```

The results viewer shows a table named 'product_ingredient 1' with one column: 'id_product'. The data is as follows:

id_product
1
1
1
2
2
2
2
4
4
4
4
4
4
4

Рисунок 13

3.4 Альтернативное решение с JOIN

The screenshot shows a MySQL Workbench interface. On the left, there's a sidebar with icons for navigating between databases, schemas, and tables. The main area has a query editor at the top containing a SQL query:

```
select p.id_product
from product_ingredient as p
join ingredient_supplier as i
on p.id_ingredient = i.id_ingredient
order by p.id_product;
```

Below the query editor is a results viewer titled "product_ingredient 1". It displays a table with two columns: "id" and "id_product". The data is as follows:

id	id_product
1	1
2	1
3	1
4	2
5	2
6	2
7	2
8	4
9	4
10	4
11	4
12	4
13	4

Рисунок 14

4 Применение теоретико-множественных операторов

4.1 Стандартное СТЕ

The screenshot shows a database management system interface with a dark theme. On the left, there is a vertical toolbar with icons for navigating between databases, tables, and queries. The main area contains a SQL editor window and a results grid.

SQL Editor:

```
with ingredients as (
    select distinct id_ingredient
    from ingredient_supplier
)
select
    p_i.id_product
from
    product_ingredient as p_i
join
    ingredients as i_s
on
    p_i.id_ingredient = i_s.id_ingredient
```

Results Grid:

id_product
1
1
1
2
2
2
2
4
4
4
4
4
4
4

Рисунок 15

4.2 Рекурсивное СТЕ

The screenshot shows a database development environment with a code editor and a results grid.

Code Editor:

```
with recursive hierarchy_ as (
    select
        category_id,
        name,
        parent_id,
        0 as lvl
    from categories
    where parent_id is null
union all
    select
        c.category_id,
        c.name,
        c.parent_id,
        h.lvl + 1
    from categories as c
    join hierarchy_ as h
    on c.parent_id = h.category_id
)
select * from hierarchy_
```

Results Grid:

	category_id	name	parent_id	lvl
1	1	Электроника	[NULL]	0
2	2	Бытовая техника	[NULL]	0
3	3	Смартфоны	1	1
4	4	Ноутбуки	1	1
5	5	Холодильники	2	1
6	6	Аксессуары для с	3	2

Рисунок 16