

# Lab\_1\_Nemkov\_KMBO\_05\_23

## Part 1

### Task 1

Представление для преподавателя Иванова Ольга Дмитриевна, которое демонстрирует ему список его студентов (ID *студента*, *ФИО*, *email*, *Наименование курса*). Столбец *email* должен выводить *email* в следующем формате: *первые три символа*\*\*\*@*имя* домена (пример [abv\\*\\*\\*@mail.ru](mailto:abv***@mail.ru)).

```
CREATE VIEW task1 AS
SELECT
    s.id_student AS student_id,
    s.full_name AS full_name,
    CONCAT(
        LEFT(s.email, 3),
        '***',
        SUBSTRING(s.email FROM POSITION('@' IN s.email))
    ) AS email,
    c.course_name AS course_name
FROM
    students s,
    courses c,
    teachers t
WHERE
    t.full_name = 'Иванова Ольга Дмитриевна'
    AND t.course_id = c.course_id
    AND c.course_id = ANY(s.course_ids);

SELECT * FROM task1;
```

**Result:**

	student_id integer	full_name character varying (100)	email text	course_name character varying (100)
1	1	Смирнов Сергей Васильевич	iva***@university.com	Алгоритмы и структуры данных
2	2	Петрова Мария Алексеевна	pet***@academy.edu	Алгоритмы и структуры данных
3	6	Смирнов Дмитрий Сергеевич	smi***@college.net	Алгоритмы и структуры данных
4	7	Лебедева Наталья Васильевна	leb***@university.ru	Алгоритмы и структуры данных
5	12	Павлов Александр Николаевич	pav***@university.edu	Алгоритмы и структуры данных
6	14	Соколова Елена Вячеславовна	sok***@college.edu	Алгоритмы и структуры данных
7	16	Тимофеева Алина Константиновна	tim***@university.org	Алгоритмы и структуры данных
8	19	Егорова Ксения Валерьевна	ego***@school.com	Алгоритмы и структуры данных
9	22	Семенова Олеся Вячеславовна	sem***@student.ru	Алгоритмы и структуры данных
10	25	Дмитриев Сергей Павлович	dmi***@university.com	Алгоритмы и структуры данных
11	27	Морозова Ольга Николаевна	mor***@university.org	Алгоритмы и структуры данных
12	28	Шмидт Екатерина Анатольевна	sch***@academy.edu	Алгоритмы и структуры данных
13	31	Потапова Валентина Александровна	pot***@student.org	Алгоритмы и структуры данных
14	34	Васильева Оксана Дмитриевна	vas***@school.com	Алгоритмы и структуры данных
15	36	Гончарова Олеся Валентиновна	gon***@college.org	Алгоритмы и структуры данных
16	38	Дорожкина Людмила Павловна	dor***@college.com	Алгоритмы и структуры данных
17	39	Захарова Александра Борисовна	zah***@academy.ru	Алгоритмы и структуры данных
18	42	Филиппова Екатерина Олеговна	fil***@student.org	Алгоритмы и структуры данных
19	48	Мишина Ольга Сергеевна	mis***@university.edu	Алгоритмы и структуры данных
20	50	Федорова Ксения Владимировна	fed***@academy.org	Алгоритмы и структуры данных
Total rows: 31 of 31    Query complete 00:00:00.118    Ln 20, Col 32				

## Task 2

Представление для студента с id\_студента = 8, которое демонстрирует ему список курсов, которые он посещает, вместе с фамилиями преподавателей.

```
CREATE VIEW task2 AS
SELECT
    c.course_name AS course_name,
    t.full_name AS teacher_name
FROM
    students s,
    courses c,
    teachers t
WHERE
    s.id_student = 8
    AND c.course_id = ANY(s.course_ids)
    AND t.course_id = c.course_id;
```

```
SELECT * FROM task2;
```

## Result:

	<b>course_name</b> character varying (100)	<b>teacher_name</b> character varying (100)
1	Системный анализ	Иванов Иван Иванович
2	Теория баз данных	Кузнецова Наталья Павловна

## Verify:

```
SELECT * FROM students WHERE id_student = 8;
```

	<b>id_student</b> [PK] integer	<b>full_name</b> character varying (100)	<b>phone</b> character varying (11)	<b>email</b> character varying (50)	<b>course_ids</b> integer[]
1	8	Кузнецова Татьяна Владимировна	89861234567	kuznetsova@school.org	{3,4}

```
SELECT * FROM courses WHERE course_id = 3 OR course_id = 4;
```

	<b>course_id</b> [PK] integer	<b>course_name</b> character varying (100)
1	3	Системный анализ
2	4	Теория баз данных

## Task 3

Создайте материализованное представление для курсов, которое содержит название курса, ФИО преподавателя, общее количество студентов на курсе.

```
CREATE MATERIALIZED VIEW task3 AS  
SELECT  
    c.course_name AS course_name,
```




```

t.full_name AS teacher_name,
COUNT(s.id_student) AS student_count
FROM
    students s,
    courses c,
    teachers t
WHERE
    c.course_id = ANY(s.course_ids)
    AND t.course_id = c.course_id
GROUP BY
    c.course_name, t.full_name;

SELECT * FROM task3;

```

## Result

	course_name character varying (100) 	teacher_name character varying (100) 	student_count bigint 
1	Программирование на Python	Петров Владимир Александрович	32
2	Алгоритмы и структуры данных	Иванова Ольга Дмитриевна	31
3	Теория баз данных	Кузнецова Наталья Павловна	32
4	Системный анализ	Иванов Иван Иванович	28

## Task 4

Представление с информацией о студентах, которое будет показывать id\_студента, его ФИО в формате Фамилия И. О., количество курсов и уровень нагрузки. Будем считать, что если студент изучает 1 курс, то уровень нагрузки – «Низкий», если 2-3 – «Средний», если > 3 – «Высокий».

```

CREATE VIEW task4 AS
SELECT
    s.id_student AS student_id,
    CONCAT(
        SPLIT_PART(s.full_name, ' ', 1), ' ',
        LEFT(SPLIT_PART(s.full_name, ' ', 2), 1), '. ',
        LEFT(SPLIT_PART(s.full_name, ' ', 3), 1), '.'
    ) AS formatted_full_name,
    ARRAY_LENGTH(s.course_ids, 1) AS course_count,
    CASE

```

```

        WHEN ARRAY_LENGTH(s.course_ids, 1) = 1 THEN 'Низкий'
        WHEN ARRAY_LENGTH(s.course_ids, 1) BETWEEN 2 AND 3 THEN
'Sредний'
        ELSE 'Высокий'
    END AS workload_level
FROM
    students s;

SELECT * FROM task4;

```

## Result

	student_id 	formatted_full_name 	course_count 	workload_level 
1	1	Смирнов С. В.	2	Средний
2	2	Петрова М. А.	2	Средний
3	3	Сидоров А. В.	1	Низкий
4	4	Козлова Е. М.	2	Средний
5	5	Новиков О. П.	2	Средний
6	6	Смирнов Д. С.	1	Низкий
7	7	Лебедева Н. В.	3	Средний
8	8	Кузнецова Т. В.	2	Средний
9	9	Попова И. П.	1	Низкий
10	10	Михайлов А. О.	1	Низкий
Total rows: 81 of 81      Query complete 00:00:00.201      Ln 1, Col 1				

## Part 2

### Task 1

Найти все игры связанные с игрой “Resident Evil.”

```

WITH RECURSIVE task1 AS (
    SELECT
        game_id,
        title,

```

```

        parent_game_id,
        relation_type
FROM games
WHERE title = 'Resident Evil'
UNION
SELECT
    g.game_id,
    g.title,
    g.parent_game_id,
    g.relation_type
FROM games g
INNER JOIN task1 t1
    ON (g.parent_game_id = t1.game_id OR g.game_id =
t1.parent_game_id)
)
SELECT * FROM task1

```

## Result

	game_id integer	title character varying (100)	parent_game_id integer	relation_type character varying (20)	visited integer[]
1	23	Resident Evil	[null]	[null]	{23}
2	24	Resident Evil 2	23	сиквел	{23,24}
3	22	Resident Evil 0	24	приквел	{23,24,22}
4	25	Resident Evil 2 (Remake)	24	ремейк	{23,24,25}
5	26	Resident Evil 3: Nemesis	24	сиквел	{23,24,26}
6	33	Resident Evil: Operation Raccoon City	24	спин-офф	{23,24,33}
7	36	Resident Evil: The Darkside Chronicles	24	спин-офф	{23,24,36}
8	37	Resident Evil: The Umbrella Chronicles	22	спин-офф	{23,24,22,37}
9	27	Resident Evil 3 (Remake)	26	ремейк	{23,24,26,27}
10	28	Resident Evil 4	26	сиквел	{23,24,26,28}
11	29	Resident Evil 4 (Remake)	28	ремейк	{23,24,26,28,29}
12	30	Resident Evil 5	28	сиквел	{23,24,26,28,30}
13	31	Resident Evil 6	30	сиквел	{23,24,26,28,30,31}
14	34	Resident Evil Revelations	30	спин-офф	{23,24,26,28,30,34}
15	35	Resident Evil Revelations 2	30	спин-офф	{23,24,26,28,30,35}
16	32	Resident Evil 7: Biohazard	31	сиквел	{23,24,26,28,30,31,32}
17	38	Resident Evil Village	32	сиквел	{23,24,26,28,30,31,32,38}

## Task 2





Для игры «The Witcher 3: Wild Hunt» вывести всех предков.

```

WITH RECURSIVE task2 AS (
  -- Базовая часть
  SELECT game_id, title, parent_game_id, relation_type
  FROM games
  WHERE title = 'The Witcher 3: Wild Hunt'
  -- Рекурсивная часть
  UNION
  SELECT g.game_id, g.title, g.parent_game_id, g.relation_type
  FROM games g
  INNER JOIN task2 a ON g.game_id = a.parent_game_id
)
SELECT * FROM task2;

```

## Result

	game_id 	title 	parent_game_id 	relation_type 
1	41	The Witcher 3: Wild Hunt	40	сиквел
2	40	The Witcher 2: Assassins of Kings	39	сиквел
3	39	The Witcher	[null]	[null]