

ГУАП

КАФЕДРА № 42

ОТЧЕТ
ЗАЩИЩЕН С ОЦЕНКОЙ _____

ПРЕПОДАВАТЕЛЬ

Доцент				Бржезовский А. В.
должность, уч. степень, звание		подпись, дата		инициалы, фамилия

ОТЧЕТ О ЛАБОРАТОРНОЙ РАБОТЕ №4

ЗАПРОСЫ НА ЯЗЫКЕ SQL: ВЫБОРКА ДАННЫХ

Вариант 5

по курсу: Методы и средства проектирования информационных систем
и технологий

РАБОТУ ВЫПОЛНИЛ

СТУДЕНТ ГР. №	4128			Воробьев В.А.
			подпись, дата	инициалы, фамилия

Санкт-Петербург 2024

СОДЕРЖАНИЕ

1	Постановка задачи	3
1.1	Цель работы	3
1.2	Задание	3
1.3	Содержание отчета	3
2	Выполнение работы	5
2.1	Задание 4	5
2.2	Задание 5	6
2.3	Задание 6	7
2.4	Запросы для демонстрации	7
3	Вывод	11
	ПРИЛОЖЕНИЕ	12

1 Постановка задачи

1.1 Цель работы

Получить навыки работы с агрегатными функциями. Освоить агрегатные директивы `count`, `sum`, `avg`, `min`, `max`, `group by`, `distinct`, `having`, а также `union`, `intersect`, `except`.

1.2 Задание

По аналогии с примерами, приведенными выше:

- реализовать запросы г) .. е), указанные в варианте задания;
- самостоятельно предложить и реализовать запросы с агрегатными функциями, не использованными при выполнении варианта задания;
- самостоятельно предложить и реализовать запросы, демонстрирующие применение каждой из директив, приведенных в п. 4.2., показать различия в получаемых результатах при выполнении теоретико-множественных операций и операций над мультимножествами.

5 Вариант:

Создайте базу данных для хранения следующих сведений: ВУЗ, студент, группа, факультет, конференция, тема доклада, программа конференции. Составьте запросы, позволяющие выбрать:

- а) студентов первого факультета, выступавших на конференции Информатика;
- б) темы докладов студентов для заданной группы;
- в) выступления, подготовленные двумя студентами различных факультетов;
- г) количество докладов для каждой конференции;
- д) среднее количество докладов, сделанных студентами третьего факультета на конференциях;
- е) студентов, выступивших на трех или большем числе конференций;
- ж) студентов четвертого факультета, не выступавших на конференциях;
- з) студентов, выступивших на всех конференциях;
- и) пары студентов, всегда выступающие вместе.

1.3 Содержание отчета

- текст запросов на SQL (с пояснениями/комментариями);

- наборы данных, возвращаемые запросами.

2 Выполнение работы

Исходные данные взяты из лабораторной работы №2, отчет для которой есть на GitHub (URI - https://github.com/vladcto/suai-labs/blob/d8c7a508971967641d8638ebcd107539c8fd618e/6_semester/%D0%9C%D0%A1%D0%9F%D0%98%D0%A1%D0%A2/%D0%BC%D1%81%D0%B8%D0%BF%D0%B8%D1%81%D1%82_2.pdf).

Текст запросов представлен в Приложении а также в репозитории GitHub (URI - https://github.com/vladcto/suai-labs/tree/1feac804866a924523979b3a271d293076b96bdf/6_semester/%D0%9C%D0%A1%D0%9F%D0%98%D0%A1%D0%A2/4).

2.1 Задание 4

Ниже представлен запрос для выполнения задания 4. Мы используем оператор JOIN для объединения таблиц по соответствующим полям. Результаты группируются по идентификатору с помощью оператора GROUP BY. Затем применяется функция агрегации COUNT для подсчета количества докладов по каждой конференции.

Листинг 4 задания:

```
1  -- количество докладов для каждой конференции;
2  USE conference_db_lab1;
3
4  SELECT c.id                AS conference_id ,
5         COUNT(a.topic_id) AS report_count
6  FROM conference c
7
8         JOIN
9         conference_session cs ON c.id = cs.conference_id
10
11        JOIN
12        topic t ON cs.id = t.session_id
13
14        JOIN
15        authorship a ON t.id = a.topic_id
16
17  GROUP BY c.id;
```

	conference_id	report_count
1	1	3
2	2	3
3	3	3
4	4	4

Рисунок 2.1 - Результат 4 задания

2.2 Задание 5

Теперь разберем запрос для задания 5. Мы используем подзапрос для подсчета количества докладов по каждой конференции, представленных студентами из факультета с номером 3. Для этого применяются операторы JOIN для связывания таблиц student, uni_group, faculty, authorship, topic, и conference_session. Затем, во внешнем запросе, вычисляется среднее количество докладов по каждой конференции с использованием функции AVG, сгруппированные по id конференции.

Листинг 5 задания:

```

1  -- среднее количество докладов , сделанных студентами третьего
    факультета на конференциях;
2  USE conference_db_lab1;
3
4  SELECT subquery.conference_id ,
5         AVG(report_count) AS average_reports
6  FROM (SELECT cs.conference_id ,
7              COUNT(a.topic_id) AS report_count
8        FROM student s
9              JOIN
10             uni_group ug ON s.group_id = ug.id
11             JOIN
12             faculty f ON ug.faculty_id = f.id
13             JOIN
14             authorship a ON s.id = a.author_id
15             JOIN
16             topic t ON a.topic_id = t.id
17             JOIN
18             conference_session cs ON t.session_id
19                                = cs.id
20        WHERE f.number = 3
21        GROUP BY cs.conference_id) AS subquery
22 GROUP BY conference_id;
```

	conference_id	average_reports
1	3	1.0000

Рисунок 2.2 - Результат 5 задания

2.3 Задание 6

В задании 6 мы используем операторы JOIN для связывания таблиц student, authorship, topic, и conference_session, чтобы отследить участие каждого студента в конференциях. Результаты группируются по идентификатору и имени студента. Затем применяется функция COUNT для подсчета уникальных конференций, в которых участвовал каждый студент. Отбор студентов, участвовавших в трех или более конференциях, осуществляется с помощью условия HAVING.

Листинг 6 задания:

```

1  -- студентов , выступивших на трех или большем числе
   конференций
2  USE conference_db_lab1;
3
4  SELECT s.id                               AS student_id ,
5         s.name                             AS student_name ,
6         COUNT(DISTINCT cs.conference_id) AS conference_count
7  FROM student s
8       JOIN
9       authorship a ON s.id = a.author_id
10      JOIN
11      topic t ON a.topic_id = t.id
12      JOIN
13      conference_session cs ON t.session_id = cs.id
14  GROUP BY s.id , s.name
15  HAVING conference_count >= 3;

```

	student_id	student_name	conference_count
1	1	Клон 1	4

Рисунок 2.3 - Результат 6 задания

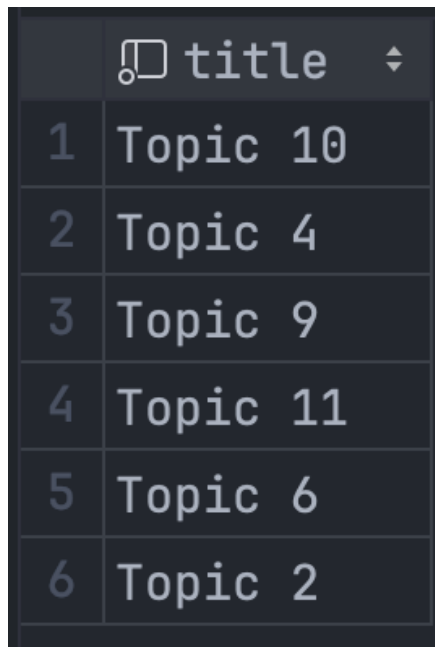
2.4 Запросы для демонстрации

В представленном SQL-скрипте мы используем операторы EXCEPT и INTERSECT для сравнения тем, представленных двумя разными авторами.

Далее применяется функция `DATEDIFF` для вычисления продолжительности каждой конференции. Наконец, оператор `UNION` используется для объединения тем и названий из двух разных таблиц в один набор данных.

Листинг except.sql:

```
1  -- Темы, над которыми работал студент студент 1, но не
   студент 2
2  USE conference_db_lab1;
3
4  SELECT t.title
5         FROM topic t
6             JOIN authorship a ON t.id = a.topic_id
7             WHERE a.author_id = 1
8
9  EXCEPT
10
11 SELECT t.title
12        FROM topic t
13            JOIN authorship a ON t.id = a.topic_id
14            WHERE a.author_id = 2;
```



	title
1	Topic 10
2	Topic 4
3	Topic 9
4	Topic 11
5	Topic 6
6	Topic 2

Рисунок 2.4 - Результат except.sql

Листинг intersect.sql:

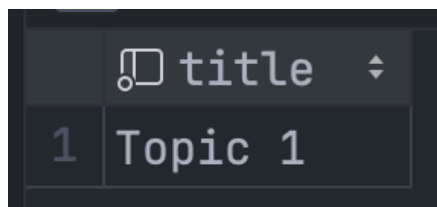
```
1  -- Темы, над которыми работал студент студент 1 и студент 2
2  USE conference_db_lab1;
3
```



```

4 SELECT t.title
5     FROM topic t
6         JOIN authorship a ON t.id = a.topic_id
7     WHERE a.author_id = 1
8
9 INTERSECT
10
11 SELECT t.title
12     FROM topic t
13         JOIN authorship a ON t.id = a.topic_id
14     WHERE a.author_id = 2;

```



	title
1	Topic 1

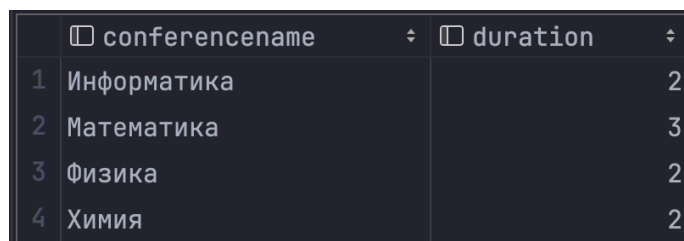
Рисунок 2.5 - Результат intersect.sql

Листинг min_max.sql:

```

1 -- Длительность конференции
2 USE conference_db_lab1;
3
4 SELECT c.name AS conferencename, DATEDIFF(MAX(cs.date), MIN(
5     cs.date)) + 1 AS duration
6     FROM conference_session cs
7         JOIN conference c ON cs.conference_id = c.id
8     GROUP BY c.name;

```



	conferencename	duration
1	Информатика	2
2	Математика	3
3	Физика	2
4	Химия	2

Рисунок 2.6 - Результат min_max.sql

Листинг union.sql:

```

1 USE conference_db_lab1;
2

```

```
3 SELECT theme AS name FROM conference
4 UNION
5 SELECT title AS name FROM topic;
```

	name
1	Тема 1
2	Тема 2
3	Тема 3
4	Тема 4
5	Topic 1
6	Topic 2
7	Topic 3
8	Topic 4
9	Topic 5
10	Topic 6
11	Topic 7
12	Topic 8
13	Topic 9
14	Topic 10
15	Topic 11

Рисунок 2.7 - Результат union.sql

3 Вывод

В результате выполнения лабораторной работы мы приобрели навыки работы с агрегатными функциями и директивами SQL. Это включает в себя использование функций `count`, `sum`, `avg`, `min`, `max`, а также директив `group by`, `distinct`, `having`, `union`, `intersect`, `except`. Эти инструменты позволяют нам обрабатывать большие объемы данных, выполнять сложные запросы и получать ценную информацию из баз данных.

В процессе работы мы также научились реализовывать запросы используя агрегатные функции и директивы, которые не были использованы в исходном задании. Это помогло нам углубить понимание возможностей SQL и развить навыки решения сложных задач.

ПРИЛОЖЕНИЕ

```
1  except.sql
2  -- Темы, над которыми работал студент студент 1, но не
   студент 2
3  USE conference_db_lab1;
4
5  SELECT t.title
6         FROM topic t
7         JOIN authorship a ON t.id = a.topic_id
8         WHERE a.author_id = 1
9
10 EXCEPT
11
12 SELECT t.title
13        FROM topic t
14        JOIN authorship a ON t.id = a.topic_id
15        WHERE a.author_id = 2;
16 min_max.sql
17 -- Длительность конференции
18 USE conference_db_lab1;
19
20 SELECT c.name AS conferencename, DATEDIFF(MAX(cs.date), MIN(
   cs.date)) + 1 AS duration
21        FROM conference_session cs
22        JOIN conference c ON cs.conference_id = c.id
23        GROUP BY c.name;
24 4.sql
25 -- количество докладов для каждой конференции;
26 USE conference_db_lab1;
27
28 SELECT c.id AS conference_id,
29        COUNT(a.topic_id) AS report_count
30        FROM conference c
31        JOIN
32        conference_session cs ON c.id = cs.conference_id
33        JOIN
34        topic t ON cs.id = t.session_id
35        JOIN
36        authorship a ON t.id = a.topic_id
37        GROUP BY c.id;
38 5.sql
```

```

39  -- среднее количество докладов, сделанных студентами третьего
    факультета на конференциях;
40  USE conference_db_lab1;
41
42  SELECT subquery.conference_id ,
43         AVG(report_count) AS average_reports
44  FROM (SELECT cs.conference_id ,
45             COUNT(a.topic_id) AS report_count
46        FROM student s
47             JOIN
48             uni_group ug ON s.group_id = ug.id
49             JOIN
50             faculty f ON ug.faculty_id = f.id
51             JOIN
52             authorship a ON s.id = a.author_id
53             JOIN
54             topic t ON a.topic_id = t.id
55             JOIN
56             conference_session cs ON t.session_id
                    = cs.id
57        WHERE f.number = 3
58        GROUP BY cs.conference_id) AS subquery
59  GROUP BY conference_id;
60
61  intersect.sql
62  -- Темы, над которыми работал студент студент 1 и студент 2
63  USE conference_db_lab1;
64
65  SELECT t.title
66        FROM topic t
67             JOIN authorship a ON t.id = a.topic_id
68        WHERE a.author_id = 1
69
70  INTERSECT
71
72  SELECT t.title
73        FROM topic t
74             JOIN authorship a ON t.id = a.topic_id
75        WHERE a.author_id = 2;
76  6.sql
77  -- студентов, выступивших на трех или большем числе

```

```

конференций
78 USE conference_db_lab1;
79
80 SELECT s.id AS student_id ,
81        s.name AS student_name ,
82        COUNT(DISTINCT cs.conference_id) AS conference_count
83 FROM student s
84 JOIN
85     authorship a ON s.id = a.author_id
86 JOIN
87     topic t ON a.topic_id = t.id
88 JOIN
89     conference_session cs ON t.session_id = cs.id
90 GROUP BY s.id , s.name
91 HAVING conference_count >= 3;
92 union.sql
93 USE conference_db_lab1;
94
95 SELECT theme AS name FROM conference
96 UNION
97 SELECT title AS name FROM topic;

```