

Тестовое задание (Задание №2) – Разработчик бизнес решений

Выполнил Ганиев Ибраим

Задание: Решите и направьте решение двух наиболее сложных для вас задач. Направляйте составленные SQL запросы с кратким разъяснением.

Решение для задачи №47:

Ссылка на задачу: https://www.sql-ex.ru/learn_exercises.php?LN=47.

Текст задачи 47: Определить страны, которые потеряли в сражениях все свои корабли.

Шаги решения:

Задача заключается в том, чтобы определить страны, которые потеряли все свои корабли в сражениях. Задача осложняется тем, что в таблице Outcomes могут присутствовать корабли, которых нет в таблице Ships. В некоторых случаях информация о кораблях может быть найдена в таблице Classes, так как корабли могут быть построены по одинаковым проектам, и классы получают название либо от головного корабля (первого построенного по проекту), либо по имени проекта.

1. Сбор данных о кораблях из разных источников:

Используя объединение через UNION, мы собираем все корабли, которые либо упоминаются в таблице Outcomes напрямую, либо связаны через классы из таблиц Ships и Classes. Для каждой записи указываем страну, которой принадлежит корабль, взяв эту информацию из таблицы Classes.

2. Определение затонувших кораблей:

Применяем условие CASE, чтобы для каждого корабля указать, был ли он потоплен (значение 'sunk' в поле result).

3. Подсчёт кораблей по странам: Считаем для каждой страны общее количество кораблей и количество потопленных кораблей.

4. Фильтрация стран, потерявших все корабли: Выбираем только те страны, у которых количество потопленных кораблей равно всем флоту.

Листинг:

```
-- Основной запрос для определения стран, потерявших все
корабли
SELECT country FROM (
    -- Подзапрос для сбора всех кораблей и определения их
    принадлежности
    SELECT
        country, name, CASE WHEN result = 'sunk' THEN 1 ELSE 0
END AS sunk -- Определяем, был ли корабль потоплен
    FROM (
        -- Собираем корабли, которые есть в Outcomes, но могут
        отсутствовать в Ships
        SELECT ship AS name, c.country -- Имя и страна корабля
        FROM Outcomes o
        JOIN Classes c ON c.class = o.ship -- Корабли могут
        быть указаны через классы
        UNION
        -- Собираем корабли, которые есть в таблице Ships
        SELECT name, c.country -- Имя и страна корабля
        FROM Ships s
        JOIN Classes c ON c.class = s.class -- Связь через
        классы
    ) AS a_s
    -- Присоединяем информацию о результатах сражений, если
    корабль был потоплен
    LEFT JOIN Outcomes o ON o.ship = a_s.name
) AS g_c
GROUP BY country -- Группируем по стране
HAVING SUM(sunk) = COUNT(DISTINCT name); -- Фильтруем только те
страны, где все корабли были потоплены, сравнивая количество всего
флота и потопленные
```

Запрос, который выводит информацию о тех кораблях, которые есть в ships или classes.

Таблица 1.

country	name	sunk
Germany	Bismarck	1
Gt.Britain	Ramillies	0
Gt.Britain	Renown	0
Gt.Britain	Repulse	0
Gt.Britain	Resolution	0
Gt.Britain	Revenge	0
Gt.Britain	Royal Oak	0
Gt.Britain	Royal Sovereign	0
Japan	Haruna	0
Japan	Hiei	0
Japan	Kirishima	1
Japan	Kongo	0
Japan	Musashi	0
Japan	Yamato	0
USA	California	0
USA	California	0
USA	Iowa	0
USA	Missouri	0
USA	New Jersey	0
USA	North Carolina	0
USA	South Dakota	0
USA	Tennessee	0
USA	Washington	0
USA	Wisconsin	0

Таблица 2.

Итоговый результат

country
Germany

Решение для задачи №69:

Ссылка на задачу: https://www.sql-ex.ru/learn_exercises.php?LN=69.

Текст задачи 38: По таблицам Income и Outcome для каждого пункта приема найти остатки денежных средств на конец каждого дня, в который выполнялись операции по приходу и/или расходу на данном пункте. Учесть при этом, что деньги не изымаются, а остатки/задолженность переходят на следующий день.

Вывод: пункт приема, день в формате "dd/mm/yyyy", остатки/задолженность на конец этого дня.

Шаги решения:

1. Сбор данных о доходах и расходах: Мы используем два подзапроса для того, чтобы собрать данные из таблиц Income и Outcome. Для каждого дня и пункта:

- Из таблицы Outcome мы выбираем сумму расходов (out).
- Из таблицы Income — сумму доходов (inc).

2. Выборка по условиям: Перед объединением в двух таблиц, группируем по point, date и для каждой группы суммируем расходы и доходы.

3. Объединение данных в один результат: Объединяем с помощью UNION ALL.

4. Подведение итогов: Конвертируем дату CONVERT(VARCHAR, [date], 103). Используя оконную функцию SUM, мы рассчитываем остатки денежных средств на конец каждого дня, аккумулируя суммы доходов и расходов с начала времени до текущего дня. Это помогает учитывать переходы остатка с предыдущих дней.

Листинг:

-- Основной запрос для вычисления остатков денежных средств

```

SELECT
    point, -- Пункт приема
    CONVERT(VARCHAR, [date], 103) AS date, -- Дата в формате
    "dd/mm/yyyy"
    -- Вычисляем остаток как накопительную сумму разности приходов и
расходов
    SUM(SUM(sum_money) - SUM(diff_money))
        OVER (PARTITION BY point ORDER BY [date] ROWS BETWEEN
UNBOUNDED PRECEDING AND CURRENT ROW) AS rem
FROM (
    -- Подзапрос для расходов
    SELECT
        date, point,
        0 AS sum_money, -- В приходах ничего нет, но необходимо для
объединения
        SUM(out) AS diff_money -- Общая сумма расходов
    FROM outcome o
    GROUP BY date, point

    UNION ALL

    -- Подзапрос для доходов
    SELECT
        date, point,
        SUM(inc) AS sum_money, -- Общая сумма доходов
        0 AS diff_money -- В расходах ничего нет, но необходимо для
объединения
    FROM income i
    GROUP BY date, point
) AS all_operation
-- Группируем по дате и пункту приема
GROUP BY date, point

```

Таблица 1.

Объединение таблиц Income и Outcome

date	point	sum_money	diff_money
2001-03-14 00:00:00.000	1	.0000	153480 000
2001-03-22 00:00:00.000	1	300000 000	.0000
2001-03-22 00:00:00.000	2	.0000	28800 000
2001-03-22 00:00:00.000	2	100000 000	.0000
2001-03-23 00:00:00.000	1	150000 000	.0000
2001-03-24 00:00:00.000	1	.0000	71630 000
2001-03-24 00:00:00.000	1	70000 000	.0000
2001-03-24 00:00:00.000	2	30000 000	.0000
2001-03-26 00:00:00.000	1	.0000	12210 000
2001-03-28 00:00:00.000	1	.0000	20750 000
2001-03-29 00:00:00.000	1	.0000	40100 000
2001-03-29 00:00:00.000	2	.0000	78480 000
2001-04-02 00:00:00.000	2	.0000	20400 000
2001-04-11 00:00:00.000	1	.0000	31950 400
2001-04-13 00:00:00.000	1	.0000	44900 000
2001-04-13 00:00:00.000	1	100000 000	.0000
2001-04-27 00:00:00.000	1	.0000	31100 000
2001-05-11 00:00:00.000	1	.0000	25300 000
2001-05-11 00:00:00.000	1	45000 000	.0000
2001-09-13 00:00:00.000	3	.0000	27000 000
2001-09-13 00:00:00.000	3	31000 000	.0000
2001-09-14 00:00:00.000	3	.0000	11500 000

Таблица 2.

Итоговый результат

point	date	rem	point
1	14.03.2001	-153480 000	1
1	22.03.2001	146520 000	1
1	23.03.2001	296520 000	1
1	24.03.2001	294890 000	1
1	26.03.2001	282680 000	1
1	28.03.2001	261930 000	1
1	29.03.2001	221830 000	1
1	11.04.2001	189879 600	1
1	13.04.2001	244979 600	1
1	27.04.2001	213879 600	1
1	11.05.2001	233579 600	1
2	22.03.2001	71200 000	2
2	24.03.2001	101200 000	2
2	29.03.2001	22720 000	2
2	02.04.2001	2320 000	2
3	13.09.2001	4000 000	3
3	14.09.2001	-7500 000	3