**Лабораторная работа №10**

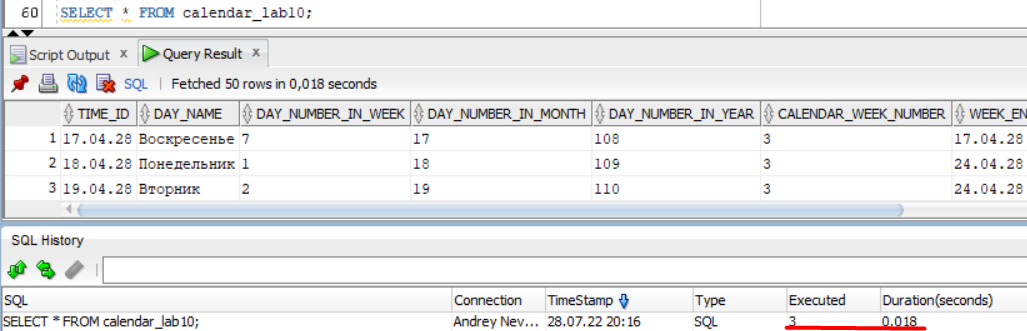
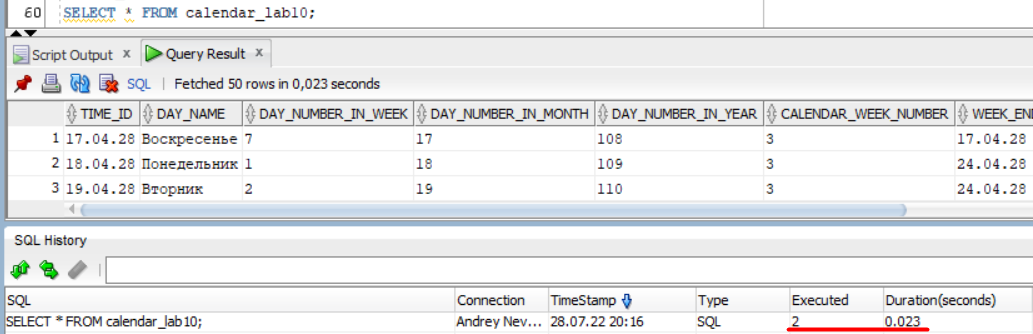
**Невейков Андрей, 2022**

**Методология:**

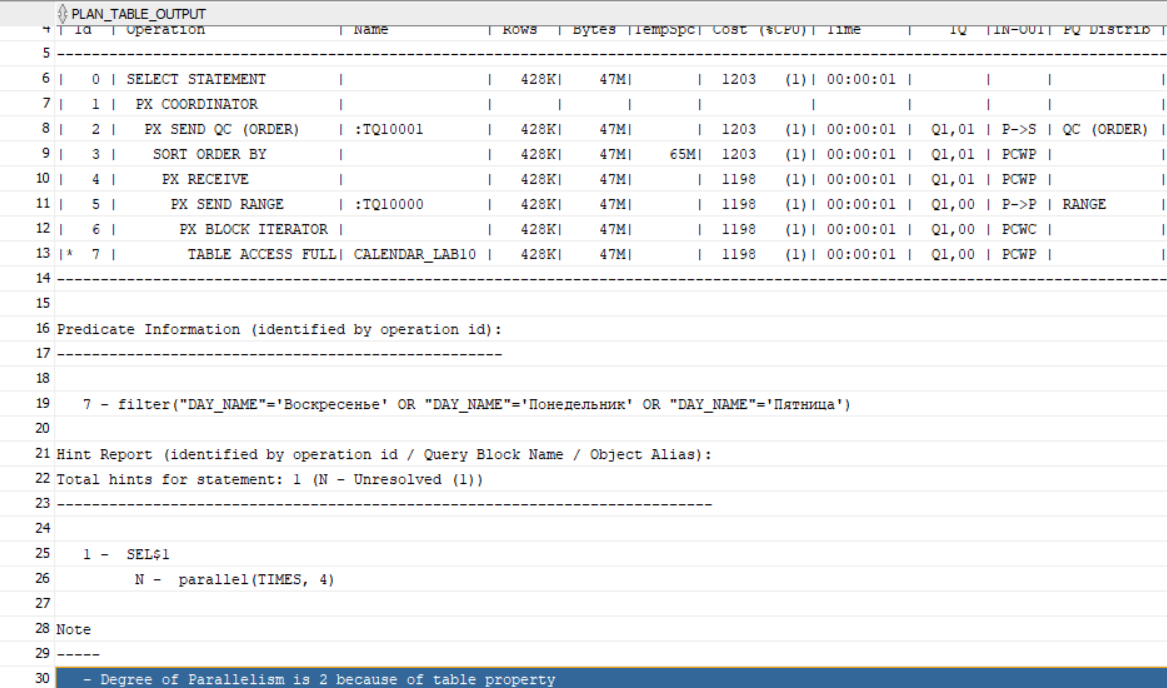
Для проверки эффективности ***Parallel execution*** был взят календарь из лабораторной работы №7. Т.к. parallel execution эффективен на больших объемах данных была создана выборка на 1.000.000 дней (один миллион). Также т.к. при выполнении команд, Oracle подгружает в кэш часто выполняемые операции, то после каждой операции таблица удалялась и пересоздавалась заново.

(В ходе выполнения лабораторной работы я проверил, как повлияет на время выполнения повтор операции без удаления таблицы с ней: с удаленение результат одной и той же операции отличался на одну-две тысячные секунды (0.001-0.002), без значение погрешности могло быть и на порядок больше (в среднем 0.017-0.023) секунды.

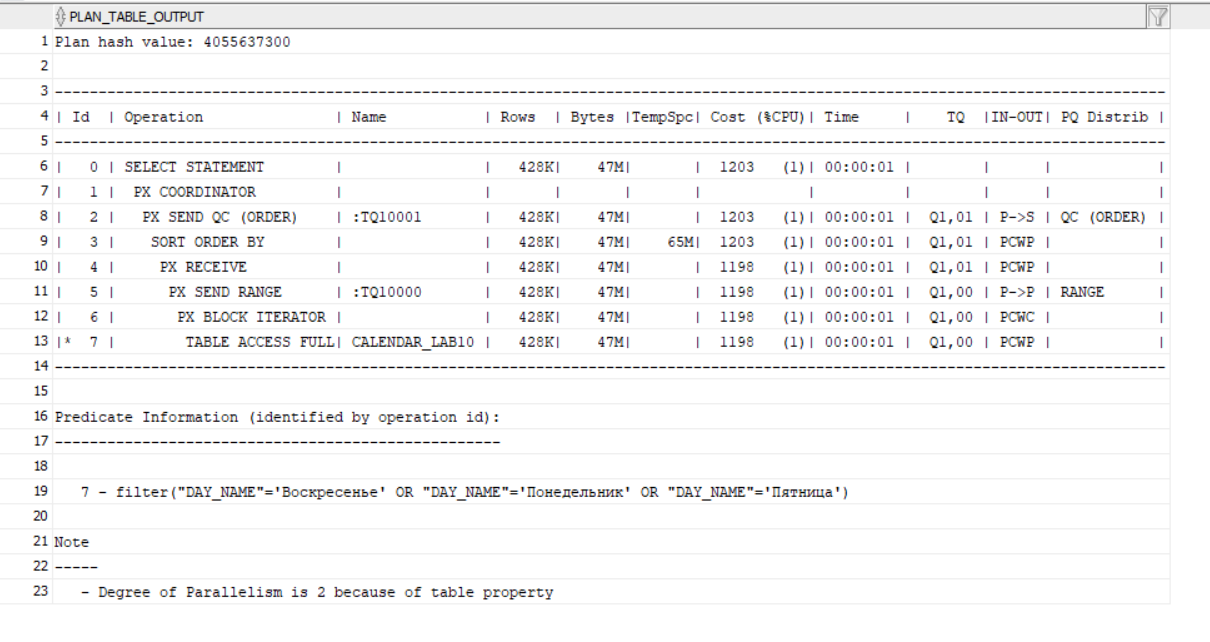
На скриншотах ниже показано как изменяется время выполнения, если выполнить запрос 2 и 3 раза подряд.



**TASK\_1** Select Parallel execution

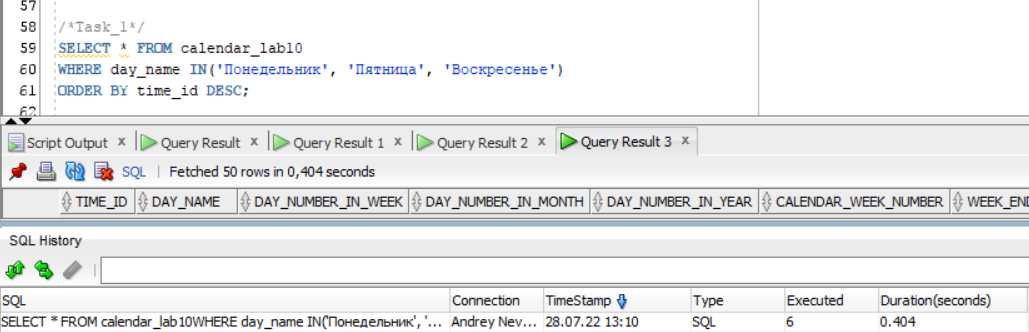


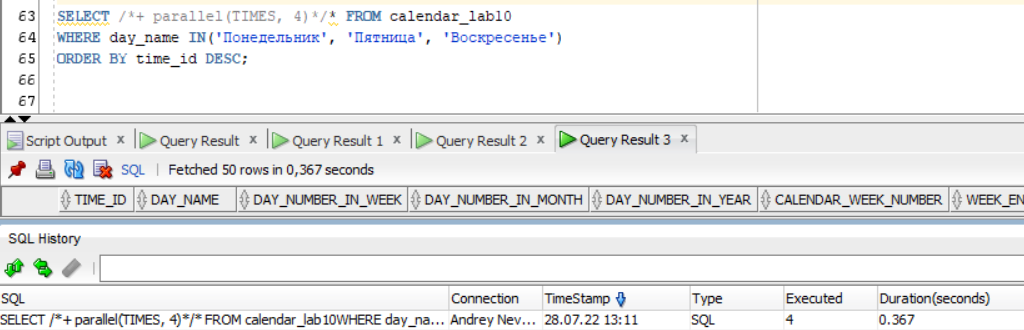
Execution Plan with parallel



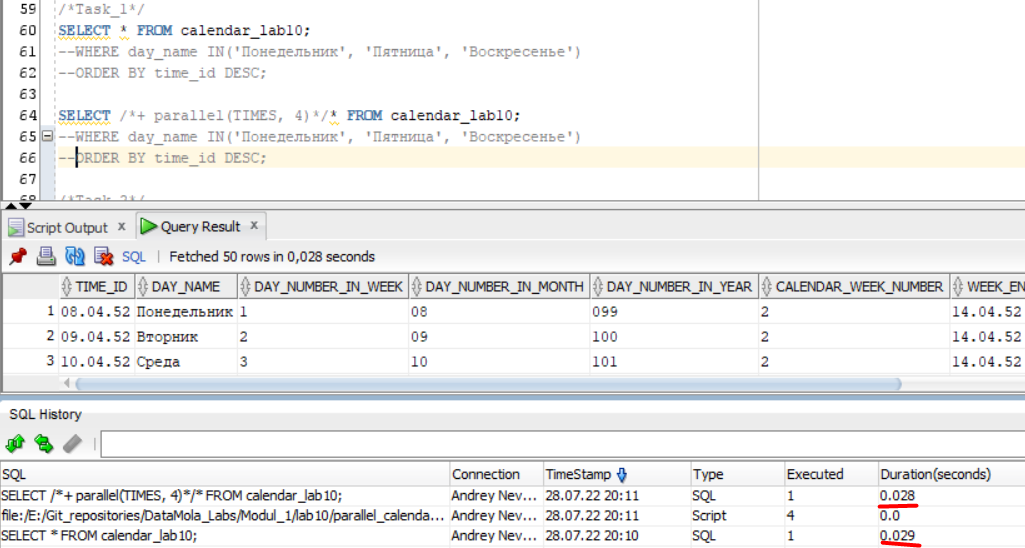
Execution Plan without parallel

Как следует из Execution Plan-ов, Oracle для больших выборок может сам использовать параллельные вычисления т.к. они почти идентичны.

****

****

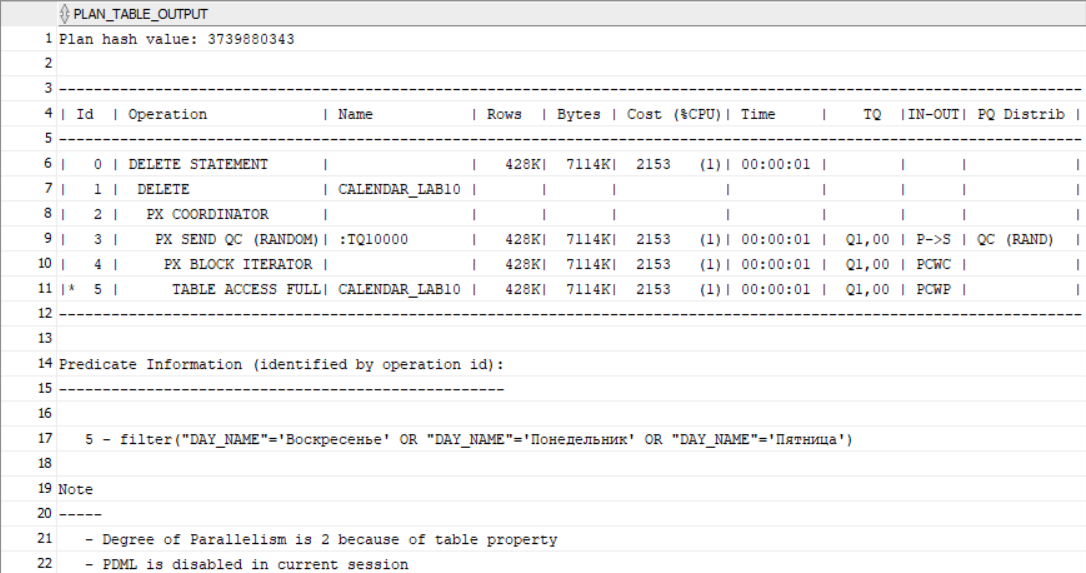
**Вывод:** parallel execution на выборке в 1 млн сток увеличил производительность на 9,2% при сложном запросе.

****

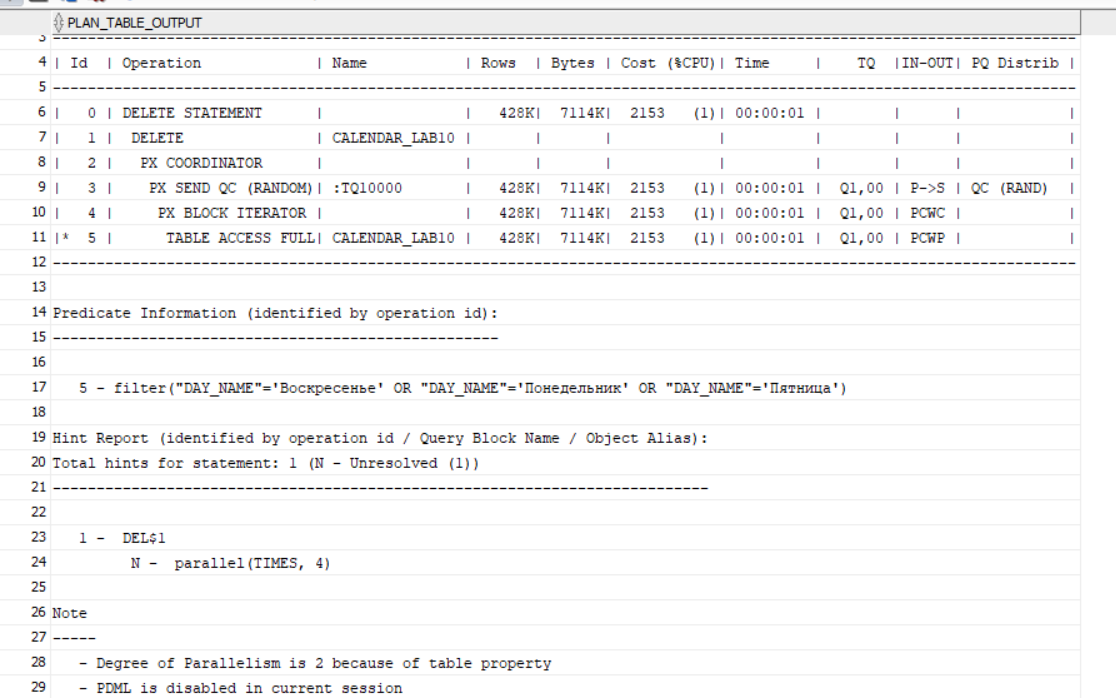
**Вывод:** parallel execution на выборке в 1 млн сток увеличил производительность на 3,5% при простом запросе.

Т.к. на сложном запросе разница видна ярче, а в работе часто будет необходимо делать запросы с условиями, то дальше будут использоваться только они.

**TASK\_2** Parallel DML

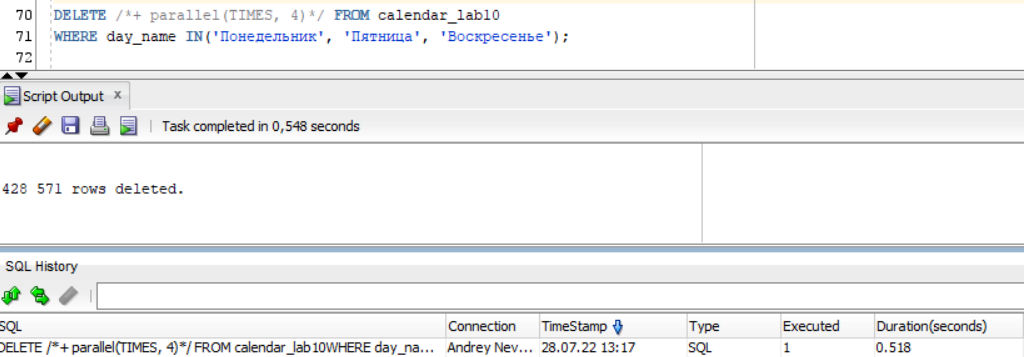
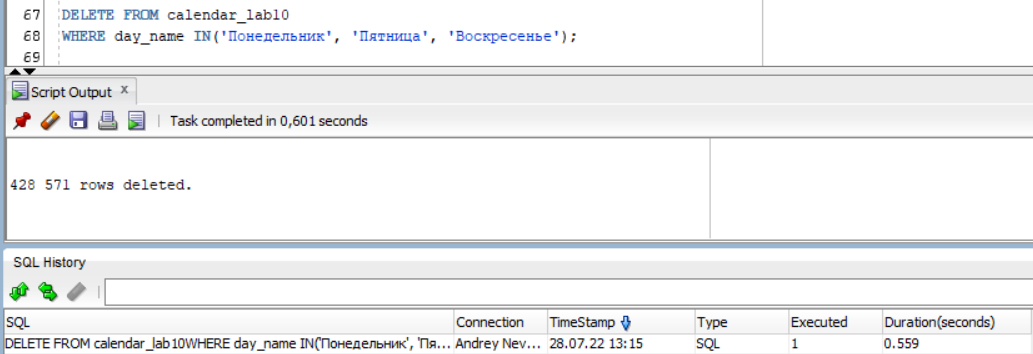


Execution Plan without parallel



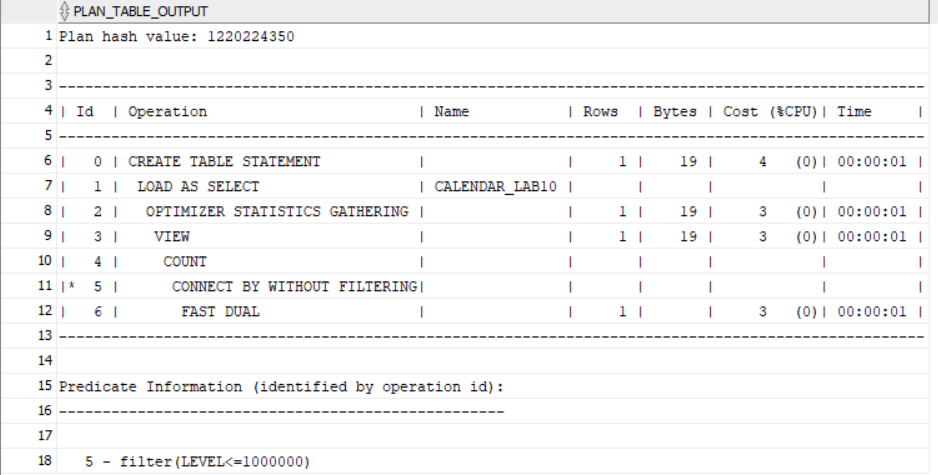
Execution Plan with parallel

Как следует из Execution Plan-ов, Oracle для больших выборок может сам использовать параллельные вычисления т.к. они почти идентичны.

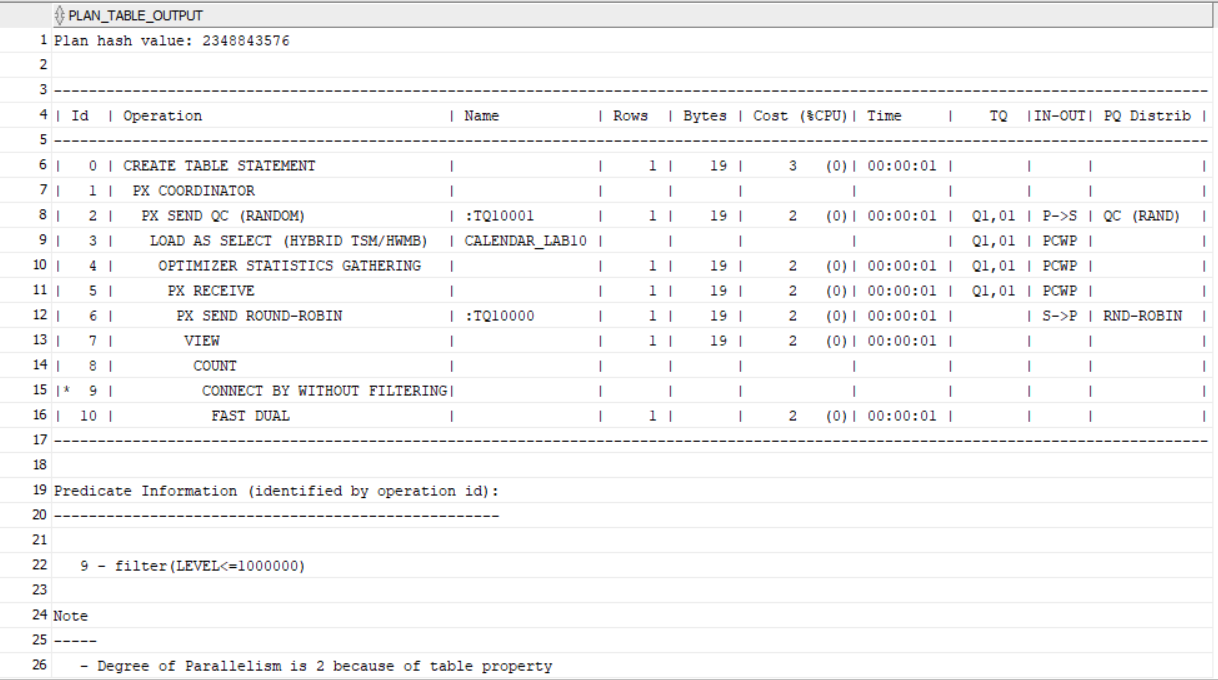
****

**Вывод:** parallel execution на выборке в 1 млн сток увеличил производительность на 7,4% при сложном запросе.

**TASK\_3** Parallel DDL

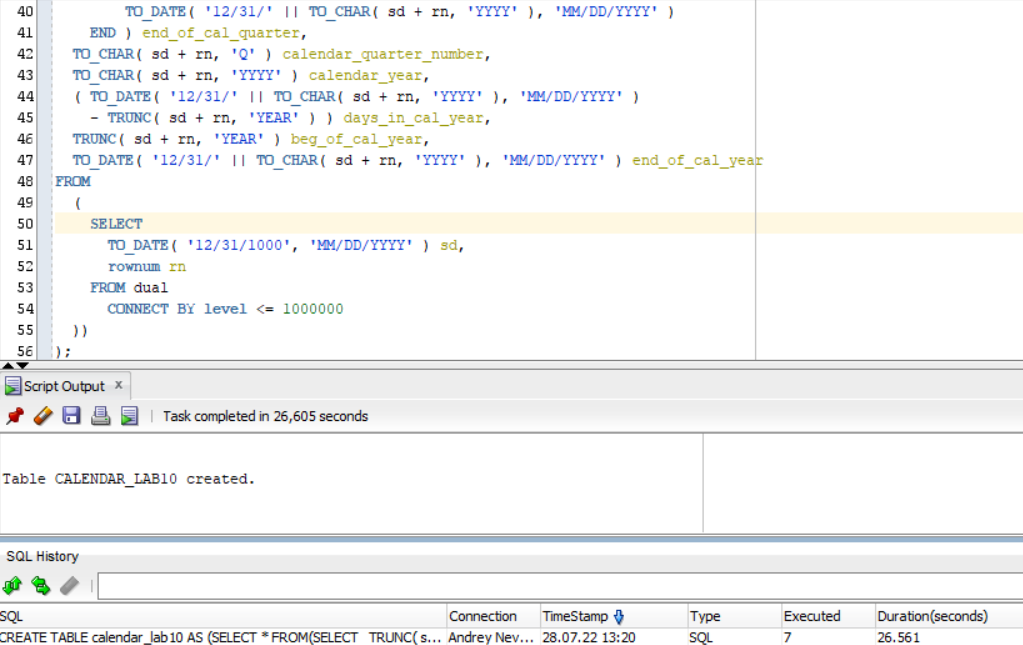


Execution Plan without parallel

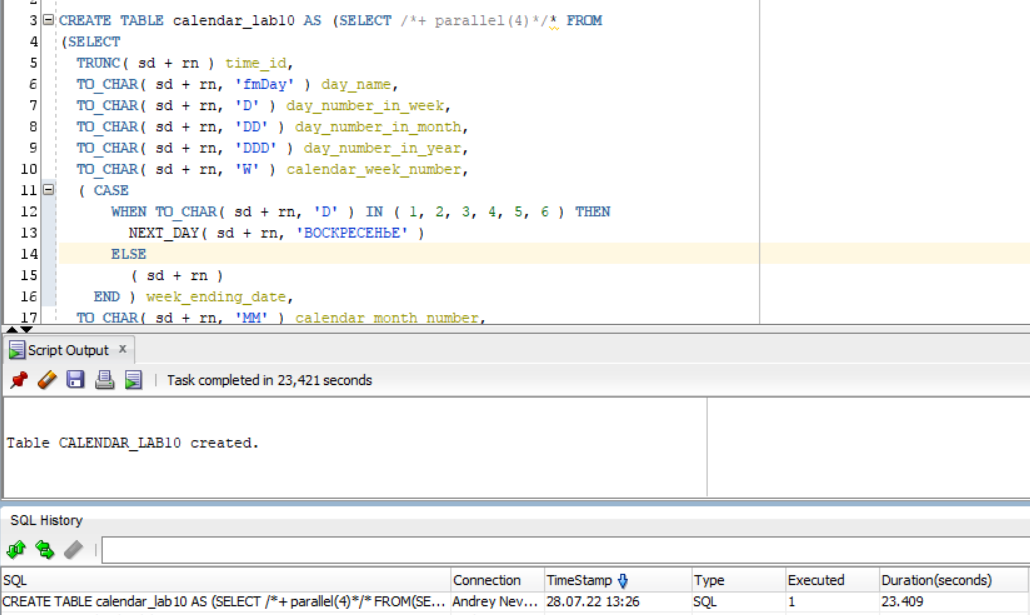
****

Execution Plan with parallel

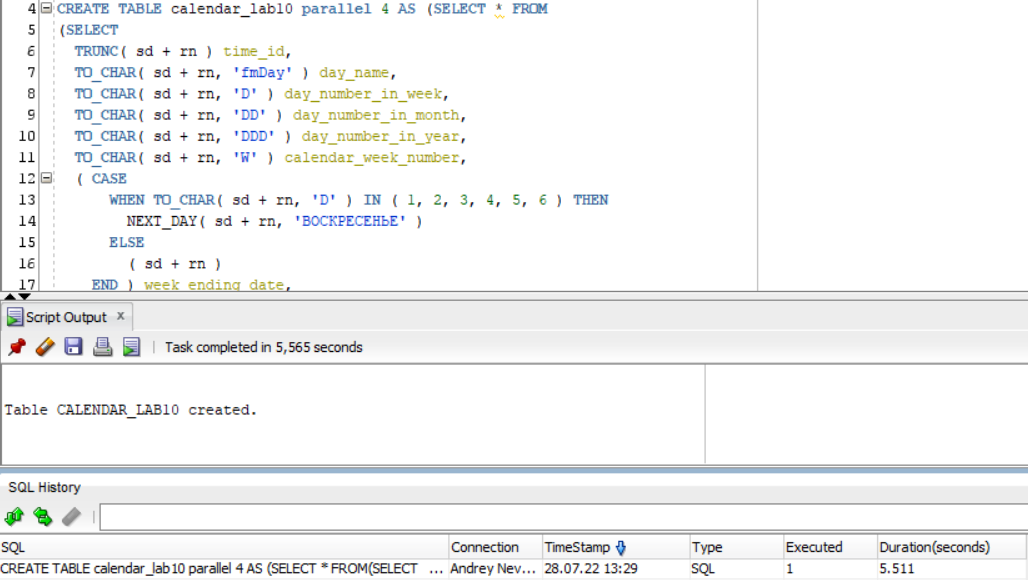
Как следует из Execution Plan-ов, Oracle для больших выборок может сам использовать параллельные вычисления т.к. они почти идентичны.

****

**Генерация таблицы без параллелизма**

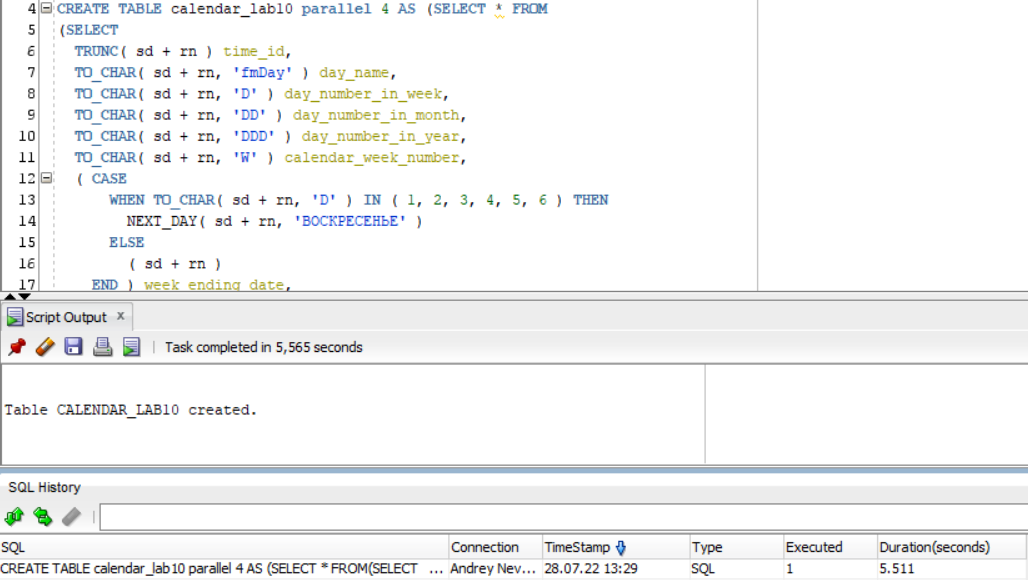
****

**Генерация таблицы используя хинт**

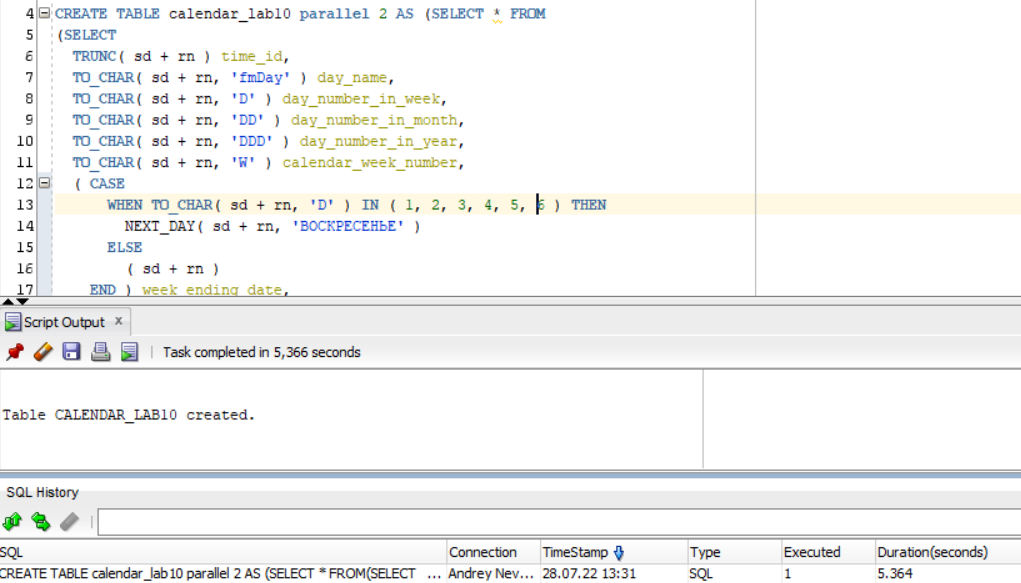
****

**Генерация таблицы используя строгое указание**

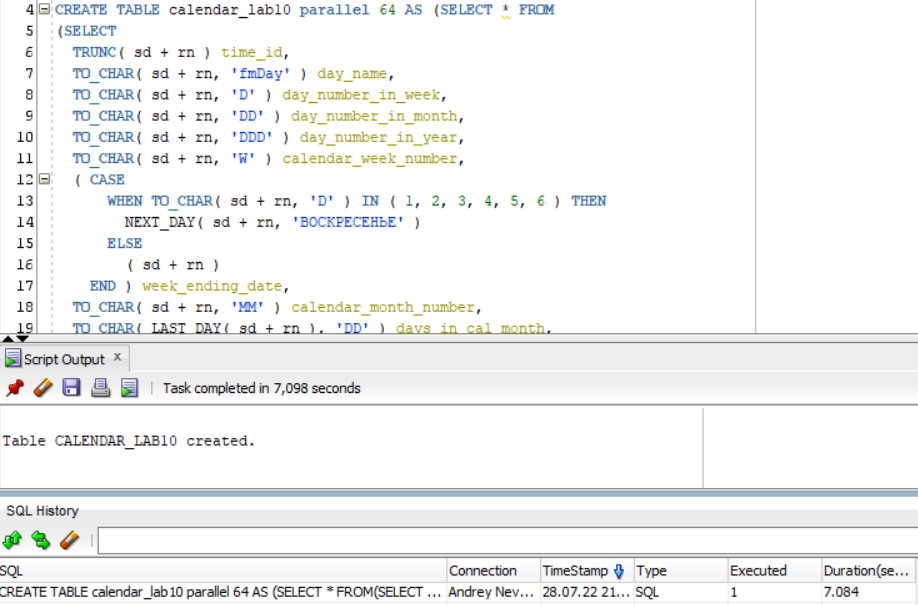
|  |  |  |
| --- | --- | --- |
| **No parallel** | **Parallel hint** | **Parallel comand** |
| **26.561 sec** | **23.409 sec** | **5.511 sec** |

****

**Генерация таблицы используя строгое указание ‘parallel 4'**

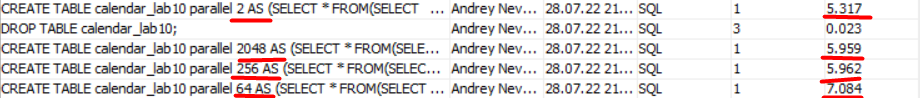
****

**Генерация таблицы используя строгое указание ‘parallel 2'**

****

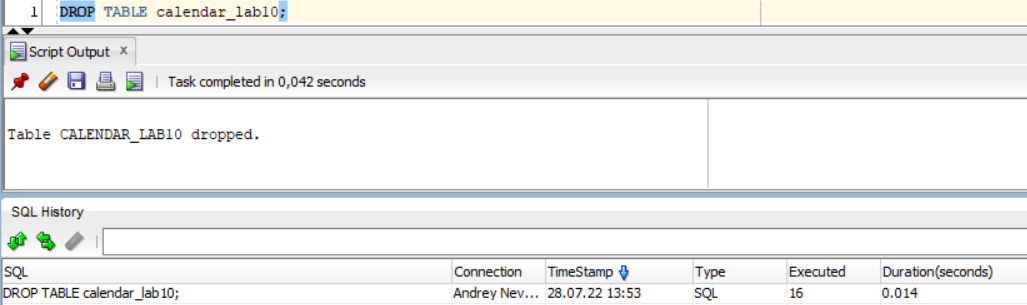
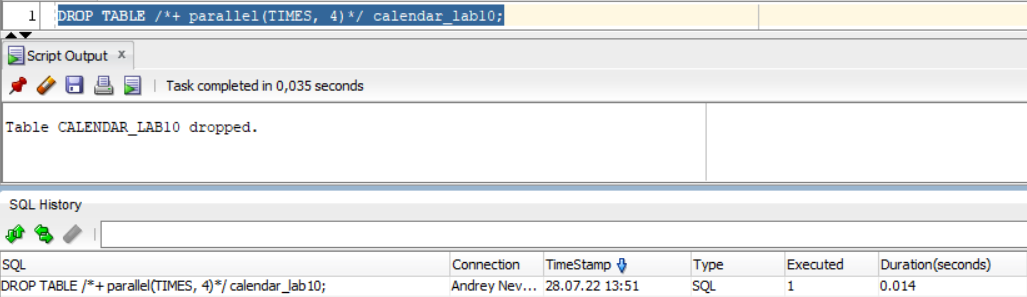
|  |  |  |
| --- | --- | --- |
| **parallel 64** | **parallel 4** | **parallel 2** |
| **7.084 sec** | **5.511 sec** | **5.364 sec** |

**Вывод:** параллелизм надо использовать с умом т.к. большее число потоков может только замедлить выполнение команды из-за ресурсоемкости процесса управления потоками.

****

Через некоторое время я проверил 64, 256, 2048 потоков. (Результаты эксперимента воспроизводимы для двух потоков время всегда в пределах 5.3, для 256 и 2048 в пределах 5,9, для 64 в пределах 7,05-7,15).

Как оказалось, **нет линейной зависимости между кол-вом потоков и временем выполнения**, но 2 потока для данной выборки оказались наиболее эффективными.

****

**Вывод:** функция drop table работает одинаково эффективно для выборки из 1 миллиона записей. Возможно это из-за того, что oracle хранит ссылки на ячейки памяти с данными (если они заняты, то их не может перезаписать другое приложение) и drop таблицы просто делает память свободной для перезаписи (операция быстрая и одинаковая).

**SUMMARY “TASK\_1-3”**

1. **Я научился** использовать parallel операции с базой данных, использовать для этого метод хинтования и метод строгого декларирования и изменять

их эффективность.

1. **Я сделал** множество выводов на основе экспериментальных наблюдений (

выводы записаны по ходу выполнения); Как следует из Execution Plan-ов, Oracle для больших выборок может сам использовать параллельные вычисления.

1. **Я понял**, принцип работы parallel операций и методологию их применения.

**FINAL SUMMARY**

Данная лабораторная работа …

1. … учит использовать параллельные вычисления и рассчитывать прирост производительность.
2. … на основе экспериментальных результатов дает понять принцип работы параллельных вычислений Oracle.
3. … показывает, что при параллельных вычислениях существуют нелинейные зависимости между производительностью и кол-вом потоков.
4. … позволяет оценить влияние параллельных вычислений на SELECT, DML и DDL операции отдельно.

**TASK\_4**

В моем бизнесе чаще всего изменяются dimention-ы: client, order и employee(т.к. компания большая). Поэтому логичнее всего использовать параллелизм в DW, CL и SA уровнях для обновления данных о них. Однако применять параллельные вычисления можно и в DM – data mart уровнях, при проектировании представлений (view), связанных с финансами. Т.к. они часто обновляются из-за большого числа транзакций, а бухгалтерам, маркетологам и т.д. необходима достоверная актуальная информация.

