**Лабораторная работа №5**

**Невейков Андрей, 2022**

**/\*Task\_1\*/**

-- Step 1

set autotrace off

set autotrace on

set autotrace traceonly

set autotrace on explain

set autotrace on statistics

set autotrace on explain statistics

set autotrace traceonly explain

set autotrace traceonly statistics

set autotrace traceonly explain statistics

set autotrace off explain

set autotrace off statistics

set autotrace off explain statistics

|  |  |  |  |
| --- | --- | --- | --- |
| № | Auto Trace Configuration Options | Expected Results | Description |
| 1 | set autotrace on | execution plan and statistics of the statement | Показывает план выполнения, а также статистику хода выполнения запроса. |
| 2 | set autotrace traceonly | execution plan and the statistics (without a query's result) | Показывает план выполнения, а также статистику хода выполнения запроса (без отображения результатов запроса) |
| 3 | set autotrace on statistics | statistics only | Показывает только статистику хода выполнения запроса |
| 4 | set autotrace on explain | execution plan only | Показывает план выполнения запроса |

**/\*Task\_2\*/**

-- Nested Loops

explain plan for

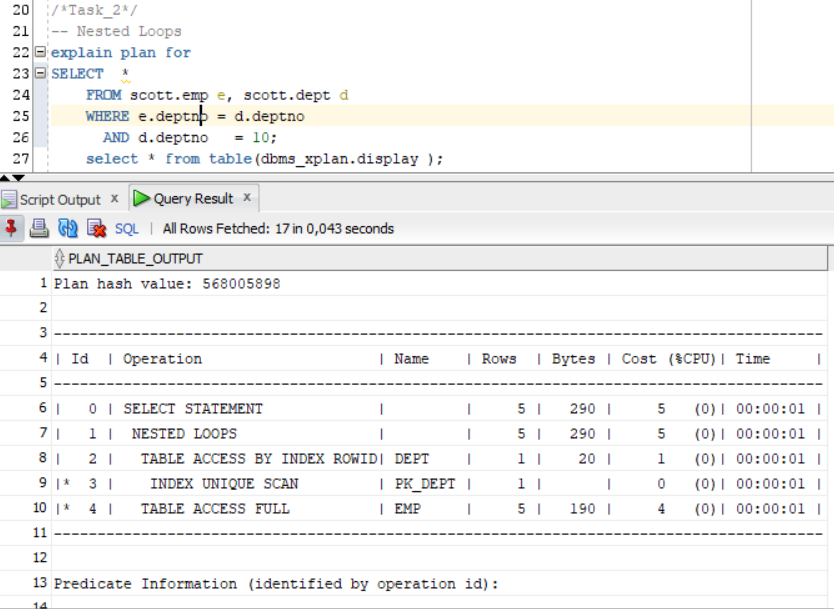
SELECT \*

FROM scott.emp e, scott.dept d

WHERE e.deptno = d.deptno

AND d.deptno = 10;

select \* from table(dbms\_xplan.display );



*Nested Loops Join (соединение вложенных циклов) сравнивает каждую строку одной*

*таблицы (внешней) с каждой строке другой таблицы (внутренней), ища те строки, которые удовлетворяют предикату соединения.*

**/\*Task\_3\*/**

-- Sort-Merge

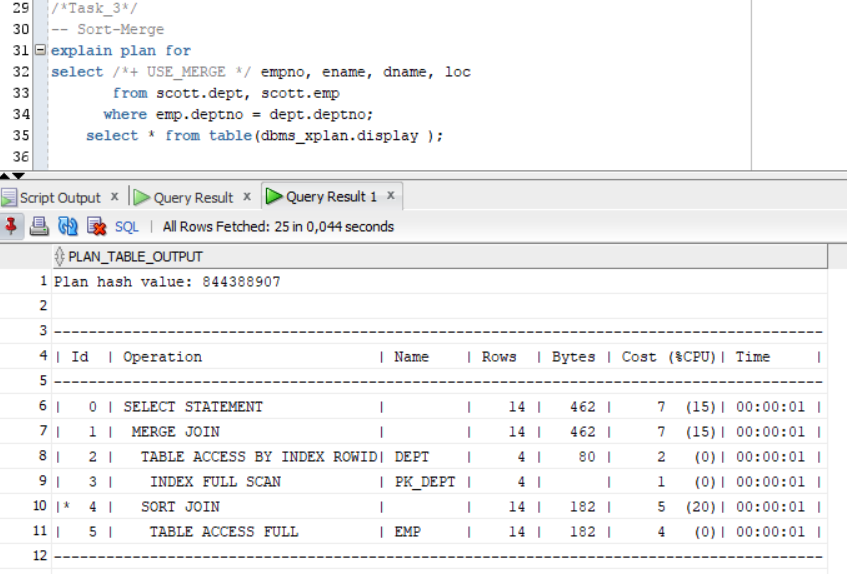
explain plan for

select /\*+ USE\_MERGE \*/ empno, ename, dname, loc

from scott.dept, scott.emp

where emp.deptno = dept.deptno;

select \* from table(dbms\_xplan.display );



*Входные таблицы должны быть отсортированы по столбцам, участвующим в условии соединения. Соединение осуществляется за одно сканирование (проход по) каждой из входных таблиц. То есть одна и та же строка считывается только один раз, что даёт преимущество перед соединением вложенными циклами.*

**/\*Task\_4\*/**

-- Hash Joins

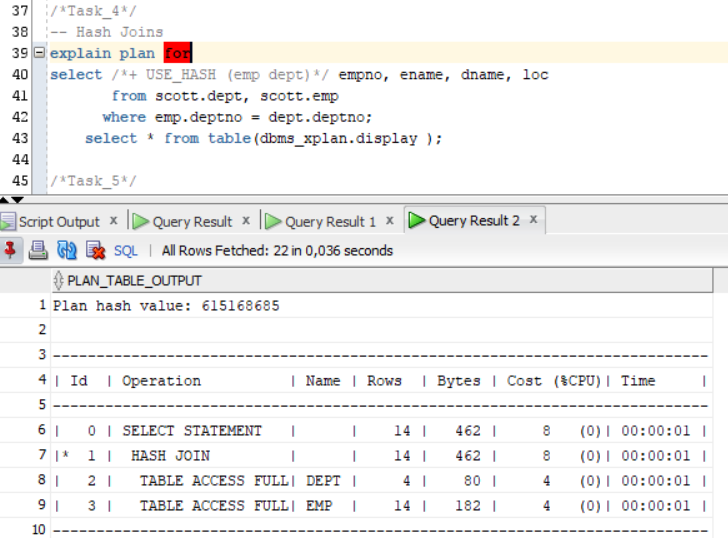
explain plan for

select /\*+ USE\_HASH (emp dept)\*/ empno, ename, dname, loc

from scott.dept, scott.emp

where emp.deptno = dept.deptno;

select \* from table(dbms\_xplan.display );



*Хэш-соединение выполняется в две стадии: компоновка и проба. Во время компоновки осуществляется чтение всех строк первого входного потока хеширование строк по ключам соединения эквивалентности и создание в оперативной памяти хеш-таблицы. Во время пробы осуществляется чтение всех строк второго входного потока, хеширование строк этого потока по тем же ключам соединения эквивалентности, а потом осуществляется просмотр или поиск соответствий строк в хеш-таблице. Так как хеш-функции могут быть подвержены, приходится проверять каждое потенциальное соответствие, что необходимо для гарантии того, что в этом случае действительно совпадение обусловлено соединением, а не коллизией.*

**/\*Task\_5\*/**

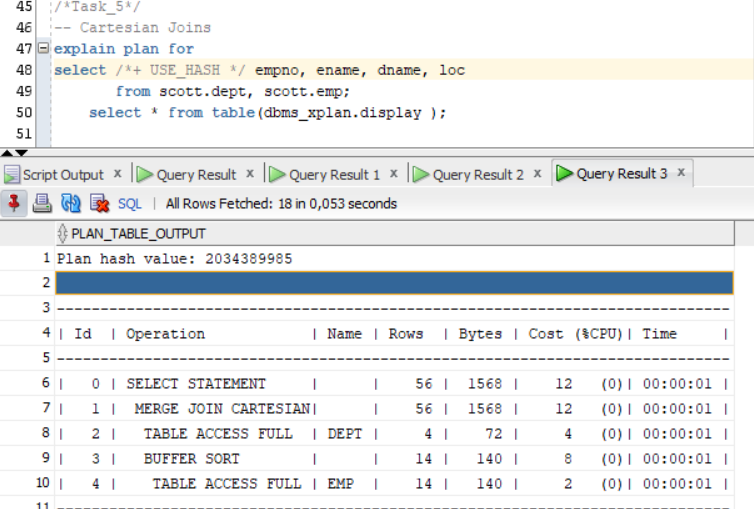
-- Cartesian Joins

explain plan for

select /\*+ USE\_HASH \*/ empno, ename, dname, loc

from scott.dept, scott.emp;

select \* from table(dbms\_xplan.display );



*Соединение каждой строки одной таблицы с каждой строкой другой таблицы. Обычно это происходит, когда соответствующий столбец или условие WHERE не указаны.*

**/\*Task\_6\*/**

-- Outer Joins

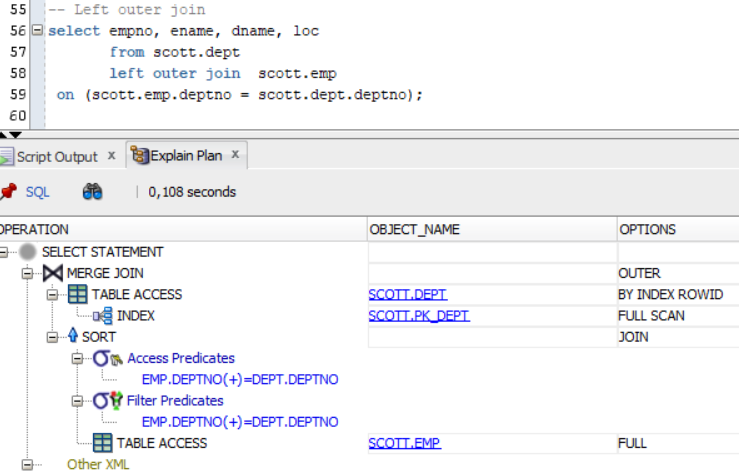
-- Left outer join

select empno, ename, dname, loc

from scott.dept

left outer join scott.emp

on (scott.emp.deptno = scott.dept.deptno);



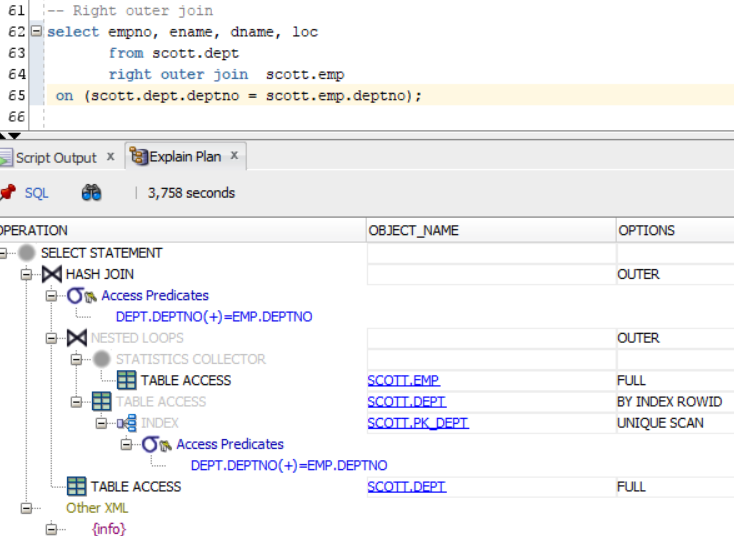
-- Right outer join

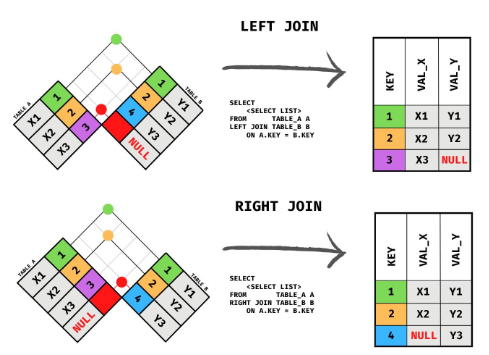
select empno, ename, dname, loc

from scott.dept

right outer join scott.emp

on (scott.dept.deptno = scott.emp.deptno);





**/\*Task\_7\*/**

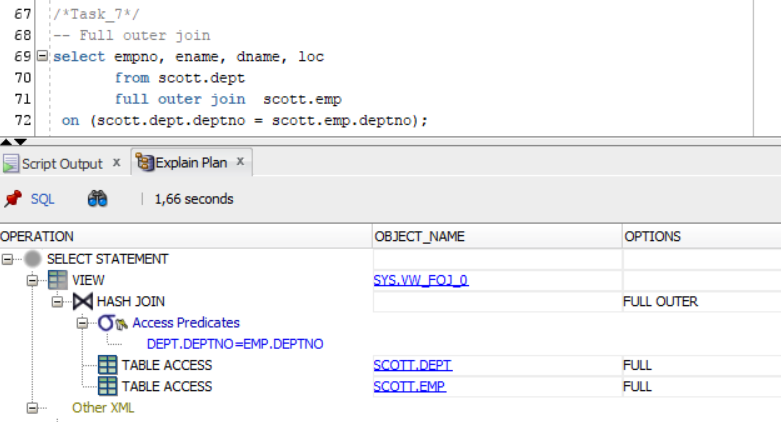
-- Full outer join

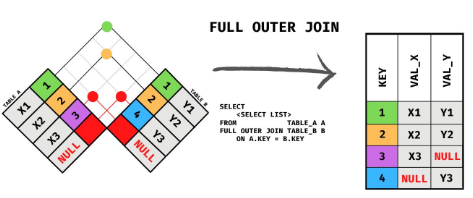
select empno, ename, dname, loc

from scott.dept

full outer join scott.emp

on (scott.dept.deptno = scott.emp.deptno);





**/\*Task\_8\*/**

-- Semi Joins

explain plan for

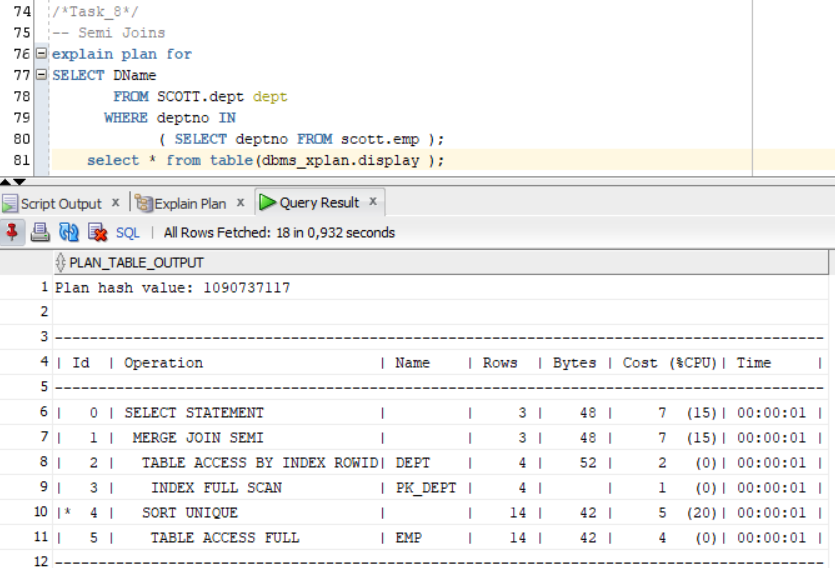
SELECT DName

FROM SCOTT.dept dept

WHERE deptno IN

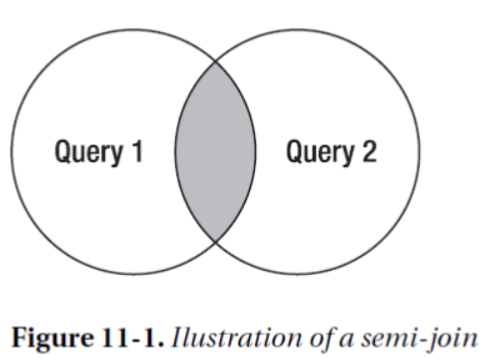
( SELECT deptno FROM scott.emp );

select \* from table(dbms\_xplan.display );



1. NL\_SJ – perform a nested loops semi-join (deprecated as of 10g)
2. HASH\_SJ – perform a hash semi-join (deprecated as of 10g)
3. MERGE\_SJ – perform a merge semi-join (deprecated as of 10g)

Были убраны в версии 10g



**/\*Task\_9\*/**

-- Anti Joins

explain plan for

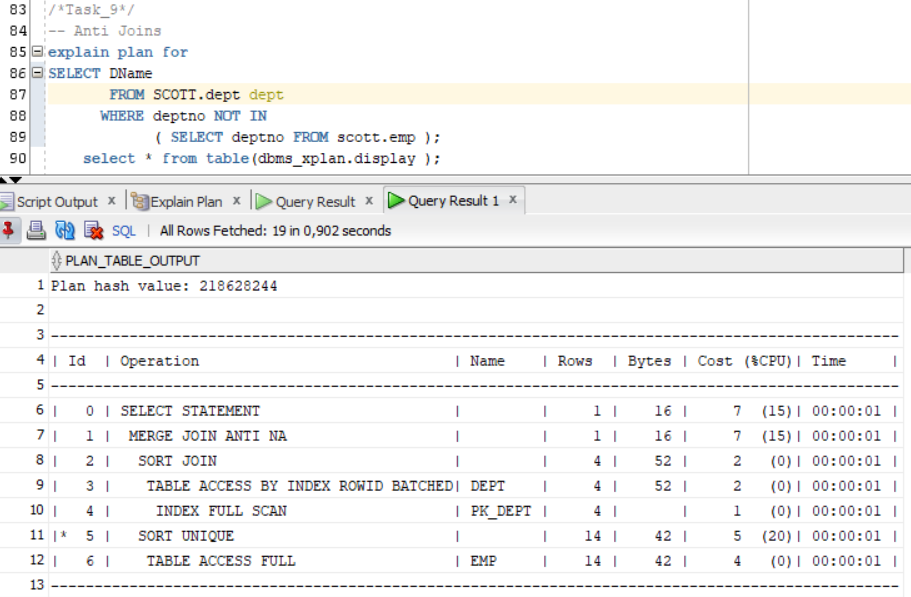
SELECT DName

FROM SCOTT.dept dept

WHERE deptno NOT IN

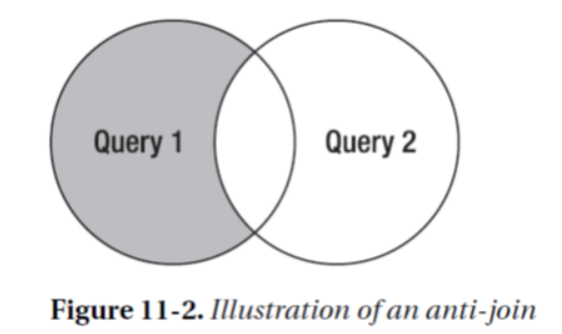
( SELECT deptno FROM scott.emp );

select \* from table(dbms\_xplan.display );



1. NL\_AJ – perform a NESTED LOOPS anti-join (deprecated as of 10g)
2. HASH\_AJ – perform a HASH anti-join (deprecated as of 10g)
3. MERGE\_AJ – perform a MERGE anti-join (deprecated as of 10g)

Были убраны в версии 10g



**/\*Task\_10\*/**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Join Access “A” | Join Access “B” | Nested Loop | Hash Join | Sort-Merge Join | Anti-Join | Semi-Join |
| Small Table | Small Table |  |  |  |  |  |
| Small Table | Indexed Small Table |  |  |  |  |  |
| Small Table | Big Table (not sorted) |  |  |  |  |  |
|  |  |  |  |  |  |  |
| Small Table | Big Table  (sorted) |  |  |  |  |  |
| Indexed Small Table | Indexed Small Table |  |  |  |  |  |
| Big | Big |  |  |  |  |  |
| Big Limited Table | Indexed Big Limited Table |  |  |  |  |  |
| Indexed Big Limited Table | Indexed Big Limited Table |  |  |  |  |  |

**SUMMARY “TASK\_1-10”**

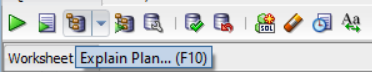
1. **Я научился** чем отличаются джоины и когда их стоит применять.
2. **Я узнал**, что джоины различаются не только по получаемому результату (операциям над множествами LEFT, RIGHT, CROSS, …), но и по способу взаимодействия с данными (более низкий уровень Nested loop, Hash join, …);
3. **Я понял**, как определить самый эффективный вид джоина для запроса (по

размерам таблиц, их отсортированности, наличию индексов)

**FINAL SUMMARY**

Данная лабораторная работа …

1. … показывает различия способов соединения таблиц (относительно низкого уровня).
2. … учит анализировать сложность запроса при помощи утилиты autotrace, конструкцией explain plan for и функцией explain plan, встроенной в SQL Developer.



1. … учит эффективно соединять таблицы, в зависимости от их размера и наличия индексов.
2. … знакомит со старыми версиями semi и anti джоинов на случай, если надо будет разбирать код версии до 10g.