

DATENBANK-ARCHITEKTUR FÜR FORTGESCHRITTENE

Execution Plans

Dani Schnider

Was ist ein Execution Plan?

- Execution Plan = Ausführungsplan = Query Evaluation Plan (QEP)
 - Abfolge von Einzelschritten, die zur Ausführung eines SQL-Befehls benötigt werden
 - Wird vom Query Optimizer während der Parse-Phase ermittelt
- Der Optimizer trifft verschiedene Entscheidungen:
 - In welcher Reihenfolge werden die Tabellen gelesen?
 - Mit welchen Methoden werden die Tabellen gejoined?
 - Wie wird auf die einzelnen Tabellen zugegriffen (Full Table Scan / Indexzugriff)?
- In diesem Kapitel werden folgende Fragen beantwortet:
 - Wie können Execution Plans in Oracle bzw. SQL Developer angezeigt werden?
 - Wie werden Execution Plans gelesen?



ANZEIGEN VON EXECUTION PLANS

EXPLAIN PLAN / DBMS_XPLAN.DISPLAY

Execution Plan, der vom Optimizer zum Parse-Zeitpunkt ermittelt wird

```
EXPLAIN PLAN FOR SELECT * FROM emp WHERE deptno = 10 ORDER BY ename;
```

```
SELECT * FROM table(dbms_xplan.display);
```

```
Plan hash value: 150391907
```

Id	Operation	Name	Rows	Bytes	Cost (%CPU)	Time	

0	SELECT STATEMENT		3	114	4 (25)	00:00:01	
1	SORT ORDER BY		3	114	4 (25)	00:00:01	
* 2	TABLE ACCESS FULL	EMP	3	114	3 (0)	00:00:01	

```
Predicate Information (identified by operation id):
```

```
-----  
2 - filter("DEPTNO">=10)
```

DBMS_XPLAN.DISPLAY_CURSOR

Execution Plan nach der Ausführung des SQL-Befehls

Enthält Informationen aus Dynamic Performance Views

- v\$sql_plan
- v\$sql_plan_statistics
- v\$sql_workarea
- v\$sql_plan_statistics_all

```
SELECT /*+ gather_plan_statistics */ c.first_name, c.last_name, a.zip_code, a.city
FROM customers c
JOIN addresses a ON (a.cust_id = c.id)
JOIN orders o ON (o.cust_id = c.id)
WHERE a.city = 'Berlin'
AND o.order_date > TRUNC(SYSDATE, 'MONTH');

SELECT * FROM TABLE(dbms_xplan.display_cursor(format => 'IOSTATS LAST'));
```

DBMS_XPLAN.DISPLAY_CURSOR

SQL_ID cggk4twx2kjkg, child number 2

Plan hash value: 1470961219

Zusätzliche Laufzeit-Informationen

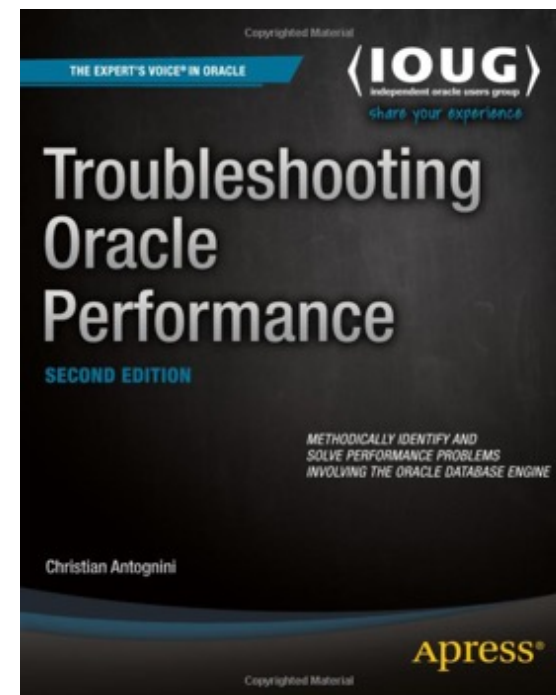
Id	Operation	Name	Starts	E-Rows	A-Rows	A-Time	Buffers
0	SELECT STATEMENT		1		2	00:00:00.01	1073
* 1	HASH JOIN		1	6	2	00:00:00.01	1073
* 2	HASH JOIN		1	127	136	00:00:00.01	467
* 3	TABLE ACCESS FULL	ADDRESSES	1	127	136	00:00:00.01	247
4	TABLE ACCESS FULL	CUSTOMERS	1	16332	16332	00:00:00.01	219
* 5	TABLE ACCESS FULL	ORDERS	1	711	655	00:00:00.01	605

Predicate Information (identified by operation id):

- 1 - access("O"."CUST_ID"="C"."ID")
- 2 - access("A"."CUST_ID"="C"."ID")
- 3 - filter("A"."CITY"='Berlin')
- 5 - filter("O"."ORDER_DATE">TRUNC(SYSDATE@!, 'fmmonth'))

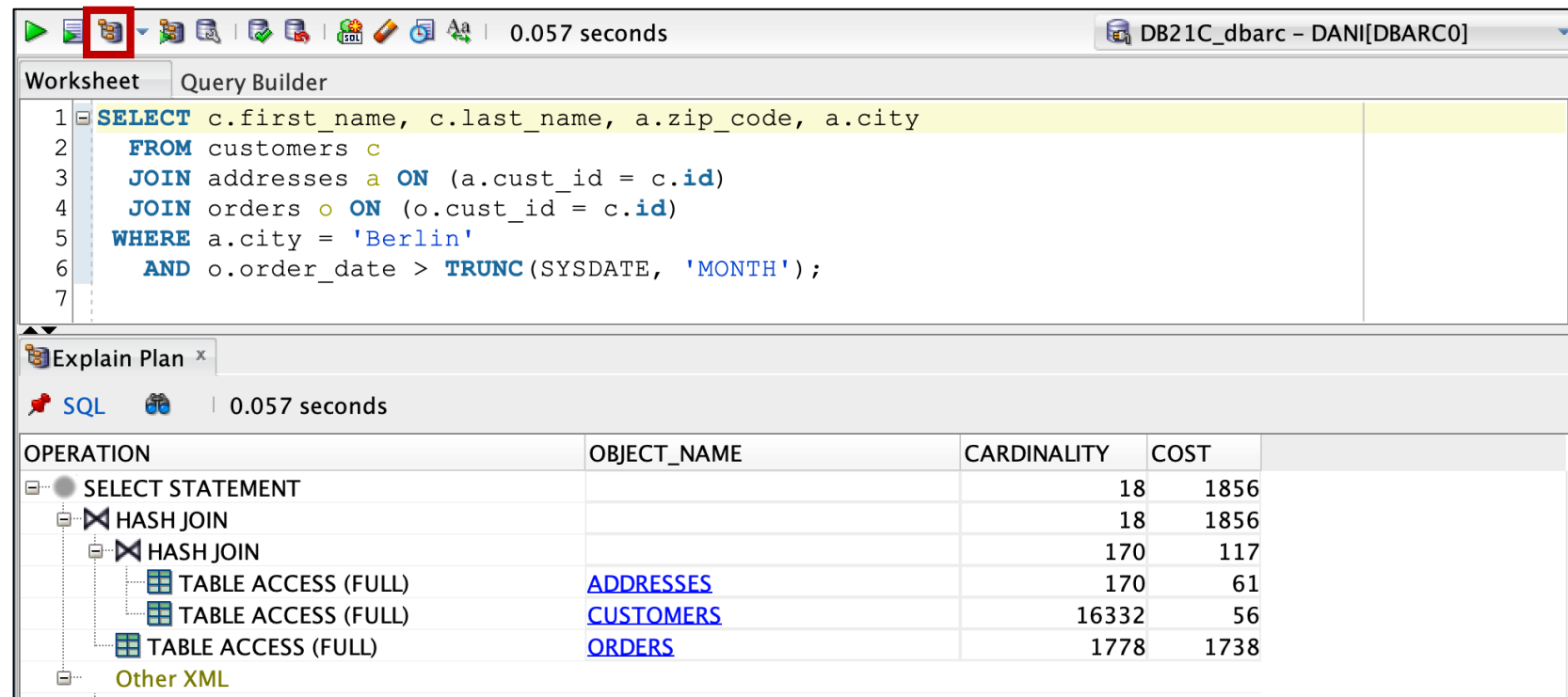
Weitere Informationen

- SQL Tuning Guide, Part III Query Execution Plans
<https://docs.oracle.com/en/database/oracle/oracle-database/21/tgsql/query-execution-plans.html>
- PL/SQL Packages and Types Reference, DBMS_XPLAN
https://docs.oracle.com/en/database/oracle/oracle-database/21/arpls/DBMS_XPLAN.html
- Christian Antognini: **Troubleshooting Oracle Performance**
Chapter 10: Execution Plans
Apress; 2nd edition (2014), ISBN 978-1430257585



SQL Developer: Explain Plan

Explain Plan... (F10), entspricht dbms_xplan.display



The screenshot shows the SQL Developer interface with a query in the Worksheet and its corresponding Explain Plan in a separate window.

Query:

```

1 SELECT c.first_name, c.last_name, a.zip_code, a.city
2 FROM customers c
3 JOIN addresses a ON (a.cust_id = c.id)
4 JOIN orders o ON (o.cust_id = c.id)
5 WHERE a.city = 'Berlin'
6 AND o.order_date > TRUNC(SYSDATE, 'MONTH');
7

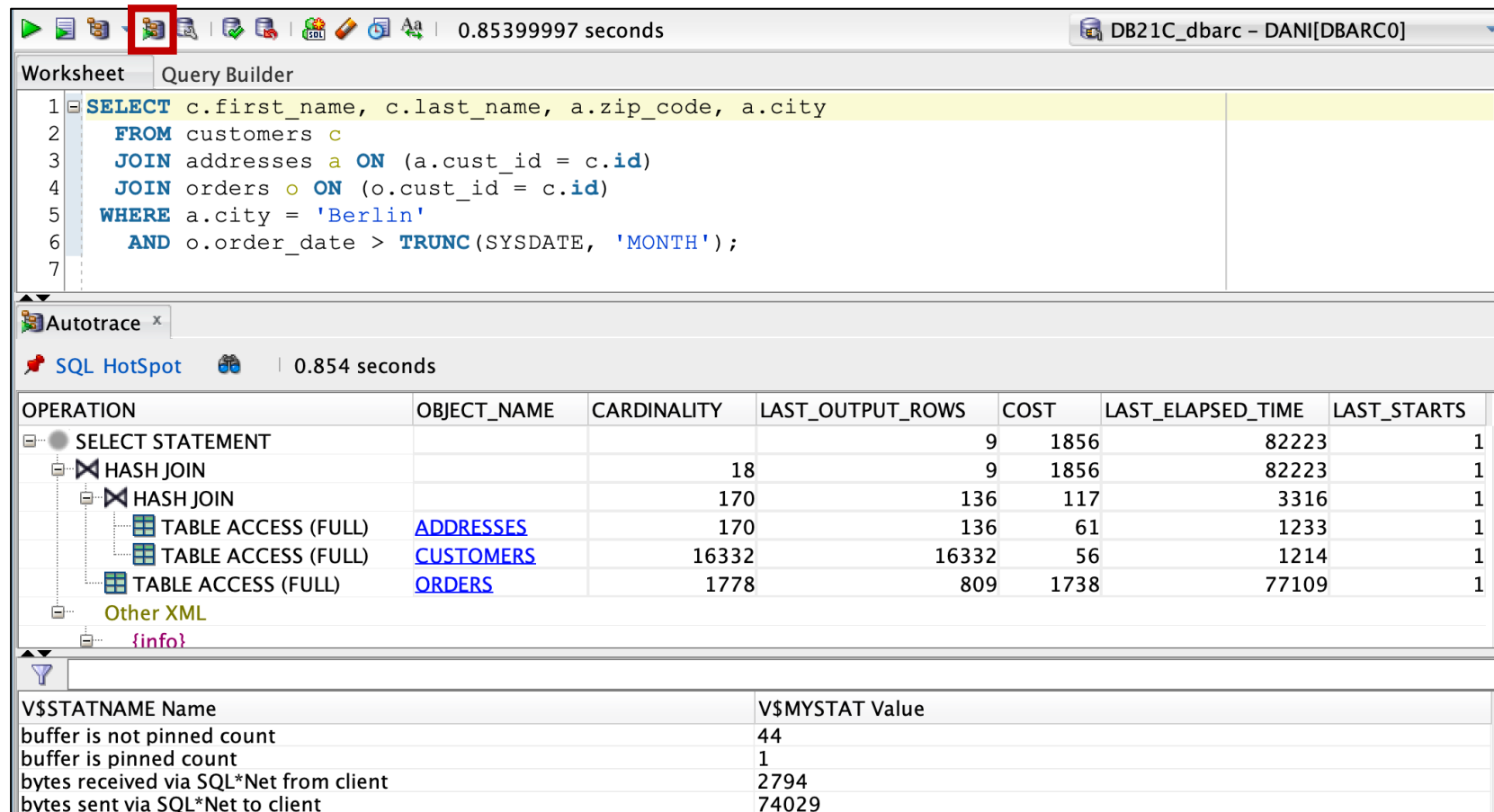
```

Explain Plan:

OPERATION	OBJECT_NAME	CARDINALITY	COST
SELECT STATEMENT		18	1856
HASH JOIN		18	1856
HASH JOIN		170	117
TABLE ACCESS (FULL)	ADDRESSES	170	61
TABLE ACCESS (FULL)	CUSTOMERS	16332	56
TABLE ACCESS (FULL)	ORDERS	1778	1738

SQL Developer: Autotrace

Autotrace... (F6), entspricht dbms_xplan.display_cursor



The screenshot shows the SQL Developer interface. The top toolbar has a red box around the Autotrace icon (a magnifying glass over a document). The main window displays a query in the Worksheet tab:

```

1 SELECT c.first_name, c.last_name, a.zip_code, a.city
2 FROM customers c
3 JOIN addresses a ON (a.cust_id = c.id)
4 JOIN orders o ON (o.cust_id = c.id)
5 WHERE a.city = 'Berlin'
6 AND o.order_date > TRUNC(SYSDATE, 'MONTH');
7

```

Below the query, the Autotrace window is open, showing the execution plan for the query. The title bar indicates "Autotrace x" and "SQL HotSpot" with a duration of "0.854 seconds".

OPERATION	OBJECT_NAME	CARDINALITY	LAST_OUTPUT_ROWS	COST	LAST_ELAPSED_TIME	LAST_STARTS
SELECT STATEMENT			9	1856	82223	1
HASH JOIN		18	9	1856	82223	1
HASH JOIN		170	136	117	3316	1
TABLE ACCESS (FULL)	ADDRESSES	170	136	61	1233	1
TABLE ACCESS (FULL)	CUSTOMERS	16332	16332	56	1214	1
TABLE ACCESS (FULL)	ORDERS	1778	809	1738	77109	1
Other XML						
{info}						

Below the execution plan, there is a section for V\$STATNAME and V\$MYSTAT values:

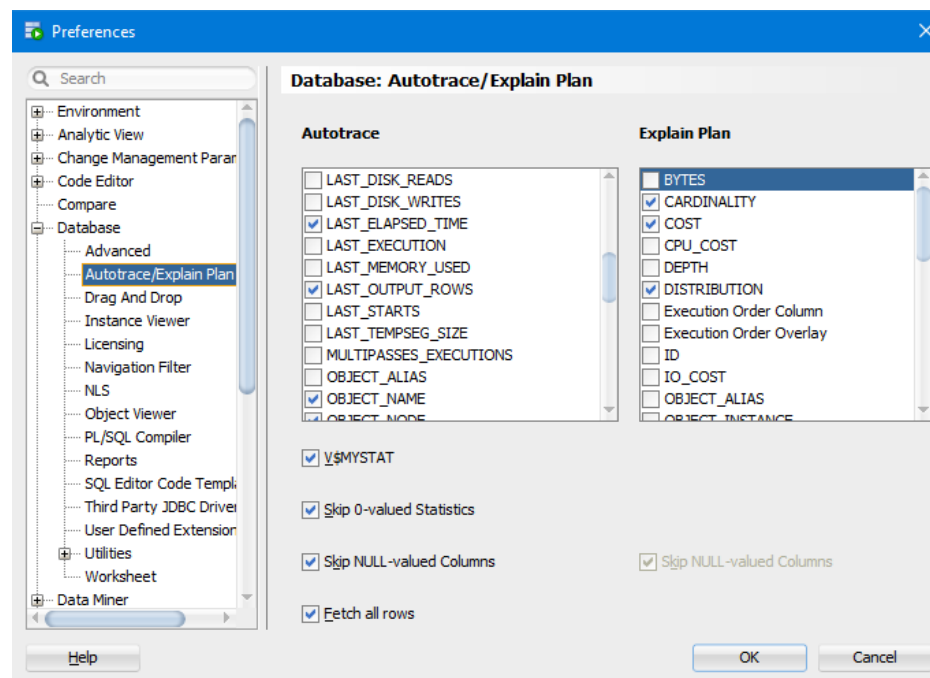
V\$STATNAME Name	V\$MYSTAT Value
buffer is not pinned count	44
buffer is pinned count	1
bytes received via SQL*Net from client	2794
bytes sent via SQL*Net to client	74029

SQL Developer Konfiguration

- Für Autotrace wird SELECT_CATALOG_ROLE benötigt

```
GRANT SELECT_CATALOG_ROLE TO dbarc0;
```

- Konfiguration von zusätzlichen Attributen (z.B. LAST_OUTPUT_ROWS)





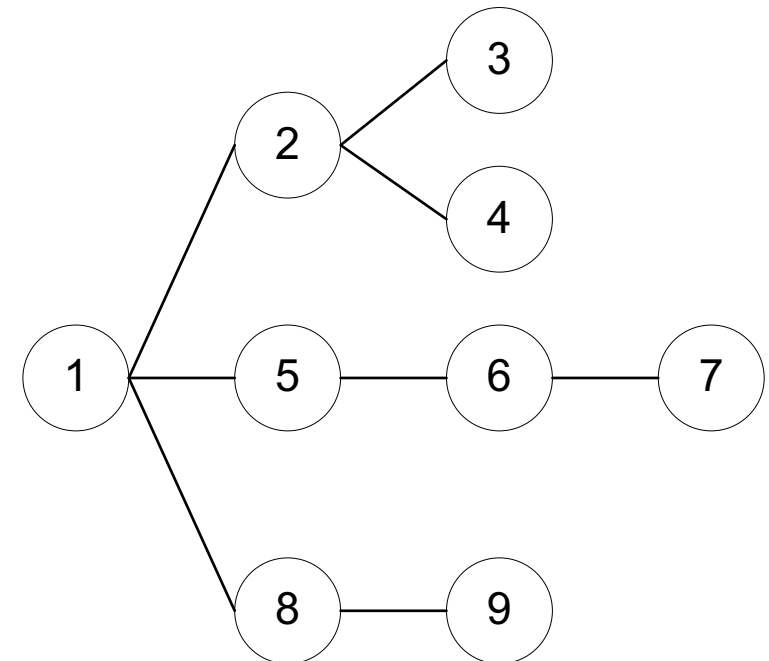
LESEN VON EXECUTION PLANS

Wie wird ein Execution Plan gelesen?

Als hierarchische Baum-Darstellung (Parent-Child)

- Von innen nach aussen
- Auf der gleichen Stufe von oben nach unten

	Id	Operation
9.	1	UPDATE
3.	2	NESTED LOOPS
1.	* 3	TABLE ACCESS FULL
2.	* 4	INDEX UNIQUE SCAN
6.	5	SORT AGGREGATE
5.	6	TABLE ACCESS BY INDEX ROWID
4.	* 7	INDEX RANGE SCAN
8.	8	TABLE ACCESS BY INDEX ROWID
7.	* 9	INDEX UNIQUE SCAN



E-Rows und A-Rows

- **E-Rows:** Vom Optimizer geschätzte Anzahl Rows während der Parse-Phase
 - SQL Developer: CARDINALITY
- **A-Rows:** Tatsächliche Anzahl Rows während der Ausführung des SQL-Befehls
 - SQL Developer: LAST_OUTPUT_ROWS

Id	Operation	Name	Starts	E-Rows	A-Rows	A-Time	Buffers
0	SELECT STATEMENT		1		2	00:00:00.01	1073
* 1	HASH JOIN		1	6	2	00:00:00.01	1073
* 2	HASH JOIN		1	127	136	00:00:00.01	467
* 3	TABLE ACCESS FULL	ADDRESSES	1	127	136	00:00:00.01	247
4	TABLE ACCESS FULL	CUSTOMERS	1	16332	16332	00:00:00.01	219
* 5	TABLE ACCESS FULL	ORDERS	1	711	655	00:00:00.01	605