

Anatomy of a SQL Tuning Session

Wolfgang Breitling

www.centrexcc.com



Copyright 2009 Centrex Consulting Corporation. Personal use of this material is permitted. However, permission to reprint/republish this material for advertising and promotional purposes or for creating new collective works for resale or redistribution to servers or lists, or to reuse any copyrighted component of this work in other works must be obtained from Centrex Consulting.



Who am I

Independent consultant since 1996
specializing in Oracle and Peoplesoft setup,
administration, and performance tuning

Member of the Oaktable Network



30+ years in database management

DL/1, IMS, ADABAS, SQL/DS, DB2, Oracle

Oracle since 1993 (7.0.12)

OCP certified DBA - 7, 8, 8*i*, 9*i*

Mathematics major from University of Stuttgart



Using the R-Method in SQL Tuning

Logical extension of TCF with the availability of `v$sql_plan_statistics` in 9i and up

At the session level

- Set `statistics_level = all`
- Set `_rowsource_execution_statistics = true`

At the sql level

- Use `GATHER_PLAN_STATISTICS` hint



The SQL

```
SELECT BUSINESS_UNIT, VOUCHER_ID, BANK_SETID
      , BANK_CD, BANK_ACCT_KEY, NAME1, INVOICE_ID
      , GROSS_AMT, TXN_CURRENCY_CD
      , TO_CHAR(SCHEDULED_PAY_DT, 'YYYY-MM-DD' )
      , PYMNT_HOLD, PYMNT_HOLD_REASON, APPR_STATUS
      , EXPRESS_PAY, VENDOR_ID
FROM PS_VCHR_HLD_VW A
WHERE '12345' IN (
      SELECT X.OPERATING_UNIT FROM PS_DISTRIB_LINE X
      WHERE A.VOUCHER_ID = X.VOUCHER_ID
      AND A.BUSINESS_UNIT = X.BUSINESS_UNIT)
```



Baseline

Id	Operation	Name	A-Time
1	NESTED LOOPS		08:46.81
2	NESTED LOOPS OUTER		07:29.40
3	NESTED LOOPS		04:52.31
4	TABLE ACCESS BY INDEX ROWID	PS_PYMNT_VCHR_XREF	04:20.48
5	INDEX SKIP SCAN	PSBPYMNT_VCHR_XREF	09.55
6	TABLE ACCESS BY INDEX ROWID	PS_VENDOR	30.75
7	INDEX UNIQUE SCAN	PS_VENDOR	06.95
8	TABLE ACCESS BY INDEX ROWID	PS_VCH_APPR_DTL	02:35.22
9	INDEX UNIQUE SCAN	PS_VCH_APPR_DTL	46.77
10	TABLE ACCESS BY INDEX ROWID	PS_VOUCHER	01:16.72
11	INDEX UNIQUE SCAN	PS_VOUCHER	01:15.33
12	INDEX RANGE SCAN	PSCDISTRIB_LINE	03.64

11 Access= "A"."BUSINESS_UNIT"="B"."BUSINESS_UNIT" AND
"A"."VOUCHER_ID"="B"."VOUCHER_ID"

12 Access= "X"."OPERATING_UNIT"='12345' AND
"X"."BUSINESS_UNIT"=:B1 AND "X"."VOUCHER_ID"=:B2



Baseline

Id	Operation	Name	Starts	Rows
1	NESTED LOOPS		1	15
2	NESTED LOOPS OUTER		1	284,029
3	NESTED LOOPS		1	284,029
4	TABLE ACCESS BY INDEX	PS_PYMNT_VCHR_XREF	1	284,029
5	INDEX SKIP SCAN	PSBPYMNT_VCHR_XREF	1	476,577
6	TABLE ACCESS BY INDEX	PS_VENDOR	284,029	284,029
7	INDEX UNIQUE SCAN	PS_VENDOR	284,029	284,029
8	TABLE ACCESS BY INDEX	PS_VCH_APPR_DTL	284,029	252,931
9	INDEX UNIQUE SCAN	PS_VCH_APPR_DTL	284,029	252,931
10	TABLE ACCESS BY INDEX	PS_VOUCHER	284,029	15
11	INDEX UNIQUE SCAN	PS_VOUCHER	284,029	15
12	INDEX RANGE SCAN	PSCDISTRIB_LINE	284,014	15



Baseline

Id	Operation	Name	Starts	Rows
1	NESTED LOOPS		1	15
2	NESTED LOOPS OUTER		1	284,029
3	NESTED LOOPS		1	284,029
4	TABLE ACCESS BY INDEX	PS_PYMNT_VCHR_XREF	1	284,029
5	INDEX SKIP SCAN	PSBPYMNT_VCHR_XREF	1	476,577
6	TABLE ACCESS BY INDEX	PS_VENDOR	284,029	284,029
7	INDEX UNIQUE SCAN	PS_VENDOR	284,029	284,029
8	TABLE ACCESS BY INDEX	PS_VCH_APPR_DTL	284,029	252,931
9	INDEX UNIQUE SCAN	PS_VCH_APPR_DTL	284,029	252,931
10	TABLE ACCESS BY INDEX	PS_VOUCHER	284,029	15
11	INDEX UNIQUE SCAN	PS_VOUCHER	284,029	15
12	INDEX RANGE SCAN	PSCDISTRIB_LINE	284,014	15



The View

```
SELECT A.BUSINESS_UNIT
, A.VOUCHER_ID
, C.NAME1
, D.EXPRESS_PAY
, B.BANK_SETID
, B.BANK_CD
, B.BANK_ACCT_KEY
...
FROM PS_VOUCHER A
, PS_PYMNT_VCHR_XREF B
, PS_VENDOR C
, PS_VCH_APPR_DTL D
WHERE A.BUSINESS_UNIT = B.BUSINESS_UNIT
AND A.VOUCHER_ID = B.VOUCHER_ID
AND C.SETID = B.REMIT_SETID
AND C.VENDOR_ID = B.REMIT_VENDOR
AND B.BUSINESS_UNIT = D.BUSINESS_UNIT(+)
AND B.VOUCHER_ID = D.VOUCHER_ID(+)
...
```




Using Scalar Subquery

```
SELECT A.BUSINESS_UNIT
, A.VOUCHER_ID
, C.NAME1
, D.EXPRESS_PAY
, B.BANK_SETID
, B.BANK_CD
, B.BANK_ACCT_KEY
...
FROM PS_VOUCHER A
, PS_PYMNT_VCHR_XREF B
, PS_VENDOR C
, PS_VCH_APPR_DTL D
WHERE A.BUSINESS_UNIT = B.BUSINESS_UNIT
AND A.VOUCHER_ID = B.VOUCHER_ID
AND C.SETID = B.REMIT_SETID
AND C.VENDOR_ID = B.REMIT_VENDOR
AND B.BUSINESS_UNIT = D.BUSINESS_UNIT(+)
AND B.VOUCHER_ID = D.VOUCHER_ID(+)
...
```



Using Scalar Subquery

```
SELECT A.BUSINESS_UNIT
, A.VOUCHER_ID
, C.NAME1
, D.EXPRESS_PAY
FROM PS_VOUCHER A
, PS_PYMNT_VCHR_XREF B
, PS_VENDOR C
, DDBBEXPRESS_PAY VOUCHER_APPR_DTL D
WHERE A.BUSINESS_UNIT = B.BUSINESS_UNIT
AND A.VOUCHER_ID = B.VOUCHER_ID
AND C.SETID = B.REMIT_SETID
AND C.VENDOR_ID = B.REMIT_VENDOR
AND DDBBBUSINESS_UNIT = DDBBBUSINESS_UNIT (+)
AND DDBBVOUCHER_ID = DDBBVOUCHER_ID (EXPRESS_PAY)
...
FROM PS_VOUCHER A
, PS_PYMNT_VCHR_XREF B
, PS_VENDOR C
WHERE A.BUSINESS_UNIT = B.BUSINESS_UNIT
AND A.VOUCHER_ID = B.VOUCHER_ID
AND C.SETID = B.REMIT_SETID
AND C.VENDOR_ID = B.REMIT_VENDOR
...
```



Using Scalar Subquery

```
SELECT A.BUSINESS_UNIT
, A.VOUCHER_ID
, (select C.NAME1 from PS_VENDOR C
   where C.SETID = B.REMIT_SETID
        AND C.VENDOR_ID = B.REMIT_VENDOR) NAME1
, (select D.EXPRESS_PAY from PS_VCH_APPR_DTL D
   where B.BUSINESS_UNIT = D.BUSINESS_UNIT
        AND B.VOUCHER_ID = D.VOUCHER_ID) EXPRESS_PAY
, B.BANK_SETID
, B.BANK_CD
, B.BANK_ACCT_KEY
...
FROM PS_VOUCHER A
, PS_PYMNT_VCHR_XREF B
WHERE A.BUSINESS_UNIT = B.BUSINESS_UNIT
      AND A.VOUCHER_ID = B.VOUCHER_ID
...
```



Subquery Factoring*

```
with PS_VCHR_HLD_VW AS (  
    SELECT A.BUSINESS_UNIT, A.VOUCHER_ID  
    ...  
    FROM PS_VOUCHER A, PS_PYMNT_VCHR_XREF B  
    ...  
)  
SELECT BUSINESS_UNIT, VOUCHER_ID, BANK_SETID  
, BANK_CD, BANK_ACCT_KEY, NAME1, INVOICE_ID  
, GROSS_AMT, TXN_CURRENCY_CD  
, TO_CHAR(SCHEDULED_PAY_DT, 'YYYY-MM-DD')  
, PYMNT_HOLD, PYMNT_HOLD_REASON, APPR_STATUS  
, EXPRESS_PAY, VENDOR_ID  
FROM PS_VCHR_HLD_VW F  
WHERE '12345' IN (  
    SELECT X.OPERATING_UNIT FROM PS_DISTRIB_LINE X  
    WHERE F.VOUCHER_ID = X.VOUCHER_ID  
    AND F.BUSINESS_UNIT = X.BUSINESS_UNIT)
```

* aka Common Table Expression (CTE)



Plan with Scalar Subqueries

Id	Operation	Name	A-Time
1	TABLE ACCESS BY INDEX ROWID	PS_VENDOR	00.01
2	INDEX UNIQUE SCAN	PS_VENDOR	00.01
3	TABLE ACCESS BY INDEX ROWID	PS_VCH_APPR_DTL	00.01
4	INDEX UNIQUE SCAN	PS_VCH_APPR_DTL	00.01
5	NESTED LOOPS		07:55.59
6	TABLE ACCESS BY INDEX ROWID	PS_PYMNT_VCHR_XREF	06:14.85
7	INDEX SKIP SCAN	PSBPYMNT_VCHR_XREF	13.86
8	TABLE ACCESS BY INDEX ROWID	PS_VOUCHER	01:39.57
9	INDEX UNIQUE SCAN	PS_VOUCHER	01:37.19
10	INDEX RANGE SCAN	PSCDISTRIB_LINE	05.91

9 Access= "A"."BUSINESS_UNIT"="B"."BUSINESS_UNIT" AND
"A"."VOUCHER_ID"="B"."VOUCHER_ID"

9 Filter= IS NOT NULL

10 Access= "X"."OPERATING_UNIT"= '12345' AND
"X"."BUSINESS_UNIT"=:B1 AND "X"."VOUCHER_ID"=:B2



Convert “in SQ” to Join

```
with PS_VCHR_HLD_VW AS (  
    SELECT A.BUSINESS_UNIT, A.VOUCHER_ID  
    ...  
    FROM PS_VOUCHER A, PS_PYMNT_VCHR_XREF B  
    ...  
)  
SELECT BUSINESS_UNIT, VOUCHER_ID, BANK_SETID  
, BANK_CD, BANK_ACCT_KEY, NAME1, INVOICE_ID  
, GROSS_AMT, TXN_CURRENCY_CD  
, TO_CHAR(SCHEDULED_PAY_DT, 'YYYY-MM-DD')  
, PYMNT_HOLD, PYMNT_HOLD_REASON, APPR_STATUS  
, EXPRESS_PAY, VENDOR_ID  
FROM MI_VCHR_HLD_VW F, PS_DISTRIB_LINE X  
WHERE X.OPERATING_UNIT = '12345'  
    AND F.VOUCHER_ID = X.VOUCHER_ID  
    AND F.BUSINESS_UNIT = X.BUSINESS_UNIT
```



Plan after “Unnest”

Id	Operation	Name	A-Time
1	TABLE ACCESS BY INDEX ROWID	PS_VENDOR	00.03
2	INDEX UNIQUE SCAN	PS_VENDOR	00.02
3	TABLE ACCESS BY INDEX ROWID	PS_VCH_APPR_DTL	00.03
4	INDEX UNIQUE SCAN	PS_VCH_APPR_DTL	00.01
5	NESTED LOOPS		01:37.64
6	NESTED LOOPS		01:37.55
7	TABLE ACCESS BY INDEX ROWID	PS_PYMNT_VCHR_XREF	22.78
8	INDEX SKIP SCAN	PSBPYMNT_VCHR_XREF	03.29
9	TABLE ACCESS BY INDEX ROWID	PS_VOUCHER	01:14.30
10	INDEX UNIQUE SCAN	PS_VOUCHER	14.23
11	INDEX RANGE SCAN	PSCDISTRIB_LINE	00.09
	8 - access("B"."PYMNT_ID"=' ')		
	filter("B"."PYMNT_ID"=' ')		
	10 - access("A"."BUSINESS_UNIT"="B"."BUSINESS_UNIT" AND		
	"A"."VOUCHER_ID"="B"."VOUCHER_ID")		
	11 - access("X"."OPERATING_UNIT"='12345' AND		
	"A"."BUSINESS_UNIT"="X"."BUSINESS_UNIT" AND		
	"A"."VOUCHER_ID"="X"."VOUCHER_ID")		



Plan after “Unnest”

Id	Operation	Name	Starts	Rows
1	TABLE ACCESS BY INDEX	PS_VENDOR	15	15
2	INDEX UNIQUE SCAN	PS_VENDOR	15	15
3	TABLE ACCESS BY INDEX	PS_VCH_APPR_DTL	21	21
4	INDEX UNIQUE SCAN	PS_VCH_APPR_DTL	21	21
5	NESTED LOOPS		1	24
6	NESTED LOOPS		1	10,099
7	TABLE ACCESS BY INDEX	PS_PYMNT_VCHR_XREF	1	284,029
8	INDEX SKIP SCAN	PSBPYMNT_VCHR_XREF	1	476,577
9	TABLE ACCESS BY INDEX	PS_VOUCHER	284,029	10,099
10	INDEX UNIQUE SCAN	PS_VOUCHER	284,029	252,931
11	INDEX RANGE SCAN	PSCDISTRIB_LINE	10,099	24



Transitive Closure

```
with PS_VCHR_HLD_VW AS (  
    SELECT B.BUSINESS_UNIT, B.VOUCHER_ID  
    ...  
    FROM PS_VOUCHER A, PS_PYMNT_VCHR_XREF B  
    ...  
)  
SELECT BUSINESS_UNIT, VOUCHER_ID, BANK_SETID  
, BANK_CD, BANK_ACCT_KEY, NAME1, INVOICE_ID  
, GROSS_AMT, TXN_CURRENCY_CD  
, TO_CHAR(SCHEDULED_PAY_DT, 'YYYY-MM-DD')  
, PYMNT_HOLD, PYMNT_HOLD_REASON, APPR_STATUS  
, EXPRESS_PAY, VENDOR_ID  
FROM MI_VCHR_HLD_VW F, PS_DISTRIB_LINE X  
WHERE X.OPERATING_UNIT = '39002'  
    AND F.VOUCHER_ID = X.VOUCHER_ID  
    AND F.BUSINESS_UNIT = X.BUSINESS_UNIT
```



Plan after Transitive Closure

Id	Operation	Name	A-Time
1	TABLE ACCESS BY INDEX ROWID	PS_VENDOR	00:00.07
2	INDEX UNIQUE SCAN	PS_VENDOR	00:00.05
3	TABLE ACCESS BY INDEX ROWID	PS_MI_VCH_APPR_DTL	00:00.09
4	INDEX UNIQUE SCAN	PS_MI_VCH_APPR_DTL	00:00.02
5	NESTED LOOPS		00:33.85
6	NESTED LOOPS		00:33.82
7	TABLE ACCESS BY INDEX ROWID	PS_PYMNT_VCHR_XREF	00:32.07
8	INDEX SKIP SCAN	PSBPYMNT_VCHR_XREF	00:04.10
9	INDEX RANGE SCAN	PSCDISTRIB_LINE	00:01.46
10	TABLE ACCESS BY INDEX ROWID	PS_VOUCHER	00:00.03
11	INDEX UNIQUE SCAN	PS_VOUCHER	00:00.01
8	- access("B"."PYMNT_ID"=' ') filter("B"."PYMNT_ID"=' ')		
9	- access("X"."OPERATING_UNIT"='12345' AND "B"."BUSINESS_UNIT"="X"."BUSINESS_UNIT" AND "B"."VOUCHER_ID"="X"."VOUCHER_ID")		
11	- access("A"."BUSINESS_UNIT"="B"."BUSINESS_UNIT" AND "A"."VOUCHER_ID"="B"."VOUCHER_ID")		



10g Outline

```
dbms_xplan.display[_cursor](...,..., 'OUTLINE [other options]')
```

Outline Data

```
/*+
  BEGIN_OUTLINE_DATA
  IGNORE_OPTIM_EMBEDDED_HINTS
  ALL_ROWS
  . . .
  INDEX_RS_ASC(@"SEL$16C51A37" "A"@"SEL$1" "PS_VOUCHER"
@"SEL$16C51A37" "X"@"SEL$4" "PSCDISTRIB_LINE"
@"SEL$16C51A37" "X"@"SEL$4" "B"@"SEL$1" "A"@"SEL$4")
  USE_NL(@"SEL$16C51A37" "A"@"SEL$1")
  USE_NL(@"SEL$16C51A37" "X"@"SEL$4")
  INDEX_RS_ASC(@"SEL$3" "D"@"SEL$3" "PS_VCH_APPR_DTL")
  INDEX_RS_ASC(@"SEL$2" "C"@"SEL$2" "PS_VENDOR")
  END_OUTLINE_DATA
*/
```



Query Block Names

```
SELECT ...
FROM VCHR_HLD_VW F
, PS_DISTRIB_LINE X
WHERE F.VOUCHER_ID = X.VOUCHER_ID
AND F.BUSINESS_UNIT = X.BUSINESS_UNIT
AND X.OPERATING_UNIT = '12345'
```

```
SELECT A.BUSINESS_UNIT
, A.VOUCHER_ID
, (select C.NAME1 from PS_VENDOR C
where C.SETID = B.REMIT_SETID
AND C.VENDOR_ID = B.REMIT_VENDOR) NAME1
, (select D.EXPRESS_PAY from PS_VCH_APPR_DTL D
where B.BUSINESS_UNIT = D.BUSINESS_UNIT
AND B.VOUCHER_ID = D.VOUCHER_ID) EXPRESS_PAY
, B.BANK_SETID
, B.BANK_CD
, B.BANK_ACCT_KEY
...
FROM PS_VOUCHER A
, PS_PYMNT_VCHR_XREF B
WHERE A.BUSINESS_UNIT = B.BUSINESS_UNIT
AND A.VOUCHER_ID = B.VOUCHER_ID
...
```

Query Block Name / Object Alias

```
1 - SEL$2 / C@SEL$2
2 - SEL$2 / C@SEL$2
3 - SEL$3 / D@SEL$3
4 - SEL$3 / D@SEL$3
5 - SEL$16C51A37
7 - SEL$16C51A37 / X@SEL$4
8 - SEL$16C51A37 / B@SEL$1
9 - SEL$16C51A37 / B@SEL$1
10 - SEL$16C51A37 / A@SEL$1
11 - SEL$16C51A37 / A@SEL$1
```



Add Hints

```
with PS_VCHR_HLD_VW AS (...)  
SELECT /*+ INDEX(@SEL$16C51A37 X@SEL$4 PSCDISTRIB_LINE)  
INDEX_RS_ASC(@SEL$16C51A37 B@SEL$1 PS_PYMNT_VCHR_XREF)  
INDEX_RS_ASC(@SEL$16C51A37 A@SEL$1 PS_VOUCHER)  
LEADING(@SEL$16C51A37 X@SEL$4 B@SEL$1 A@SEL$1)  
USE_NL(@SEL$16C51A37 B@SEL$1)  
USE_NL(@SEL$16C51A37 A@SEL$1)  
INDEX_RS_ASC(@SEL$3 D@SEL$3 PS_VCH_APPR_DTL)  
INDEX_RS_ASC(@SEL$2 C@SEL$2 PS_VENDOR) */  
BUSINESS_UNIT, VOUCHER_ID, BANK_SETID, BANK_CD  
, BANK_ACCT_KEY, NAME1, INVOICE_ID, GROSS_AMT  
, TXN_CURRENCY_CD, TO_CHAR(SCHEDULED_PAY_DT, 'YYYY-MM-DD')  
, PYMNT_HOLD, PYMNT_HOLD_REASON, APPR_STATUS  
, EXPRESS_PAY, VENDOR_ID  
FROM MI_VCHR_HLD_VW F, PS_DISTRIB_LINE X  
WHERE X.OPERATING_UNIT = '12345'  
      AND F.VOUCHER_ID = X.VOUCHER_ID  
      AND F.BUSINESS_UNIT = X.BUSINESS_UNIT
```



Plan with Hints

Id	Operation	Name	A-Time
1	TABLE ACCESS BY INDEX ROWID	PS_VENDOR	00.01
2	INDEX UNIQUE SCAN	PS_VENDOR	00.01
3	TABLE ACCESS BY INDEX ROWID	PS_VCH_APPR_DTL	00.01
4	INDEX UNIQUE SCAN	PS_VCH_APPR_DTL	00.01
5	NESTED LOOPS		00.01
6	NESTED LOOPS		00.01
7	INDEX RANGE SCAN	PSCDISTRIB_LINE	00.01
8	TABLE ACCESS BY INDEX ROWID	PS_PYMNT_VCHR_XREF	00.01
9	INDEX RANGE SCAN	PS_PYMNT_VCHR_XREF	00.01
10	TABLE ACCESS BY INDEX ROWID	PS_VOUCHER	00.01
11	INDEX UNIQUE SCAN	PS_VOUCHER	00.01
7	- access("X"."OPERATING_UNIT"='12345')		
8	- filter(("B"."PYMNT_ID"=' ' AND ...))		
9	- access("B"."BUSINESS_UNIT"="X"."BUSINESS_UNIT" AND "B"."VOUCHER_ID"="X"."VOUCHER_ID")		
11	- access("A"."BUSINESS_UNIT"="B"."BUSINESS_UNIT" AND "A"."VOUCHER_ID"="B"."VOUCHER_ID")		



Plan with Hints

Id	Operation	Name	Starts	Rows
1	TABLE ACCESS BY INDEX	PS_VENDOR	15	15
2	INDEX UNIQUE SCAN	PS_VENDOR	15	15
3	TABLE ACCESS BY INDEX	PS_VCH_APPR_DTL	21	21
4	INDEX UNIQUE SCAN	PS_VCH_APPR_DTL	21	21
5	NESTED LOOPS		1	24
6	NESTED LOOPS		1	28
7	INDEX RANGE SCAN	PSCDISTRIB_LINE	1	395
8	TABLE ACCESS BY INDEX	PS_PYMNT_VCHR_XREF	395	28
9	INDEX RANGE SCAN	PS_PYMNT_VCHR_XREF	395	395
10	TABLE ACCESS BY INDEX	PS_VOUCHER	28	24
11	INDEX UNIQUE SCAN	PS_VOUCHER	28	28



Compare to Baseline (~9 min)

Id	Operation	Name	Starts	Rows
1	NESTED LOOPS		1	15
2	NESTED LOOPS OUTER		1	284,029
3	NESTED LOOPS		1	284,029
4	TABLE ACCESS BY INDEX	PS_PYMNT_VCHR_XREF	1	284,029
5	INDEX SKIP SCAN	PSBPYMNT_VCHR_XREF	1	476,577
6	TABLE ACCESS BY INDEX	PS_VENDOR	284,029	284,029
7	INDEX UNIQUE SCAN	PS_VENDOR	284,029	284,029
8	TABLE ACCESS BY INDEX	PS_VCH_APPR_DTL	284,029	252,931
9	INDEX UNIQUE SCAN	PS_VCH_APPR_DTL	284,029	252,931
10	TABLE ACCESS BY INDEX	PS_VOUCHER	284,029	15
11	INDEX UNIQUE SCAN	PS_VOUCHER	284,029	15
12	INDEX RANGE SCAN	PSCDISTRIB_LINE	284,014	15



Techniques Used

- ✓ **Scalar Subqueries**
- ✓ **Subquery Factoring**
- ✓ **Transitive Closure**
- ✓ **Un-nesting – Convert “in” Subquery to Join**
- ✓ **Hints**



Information Used

- ✓ **dbms_xplan.display_cursor**
- ✓ **Rowsource Timing Information**
- ✓ **Rowsource Starts and Actual Rows**
- ✓ **Access and Filter Predicates**
- ✓ **Outline**
- ✓ **Index Definitions**
- ✓ **Application Knowledge**

Wolfgang Breitling

breitliw@centrexcc.com

Centrex Consulting Corp.

www.centrexcc.com

