



**Warum wird mein Index  
nicht benutzt?**

---

# MuniQSoft GmbH

## ◆ Tätigkeitsbereiche:

- ▶ **Oracle Support**  
Hotline: Mo-Fr 8.00 – 18.00 Uhr  
Erweiterung um eine Rufbereitschaft auch am Wochenende möglich
- ▶ **Oracle IT Consulting & Services**
- ▶ **Oracle Schulungen (SQL, PL/SQL, DBA, APEX, B&R, ...)**
- ▶ **Software-Lösungen**
- ▶ **Oracle Lizenzen**

MuniQSoft GmbH  
**Bereich Schulung**  
Grünwalder Weg 13a  
82008 Unterhaching  
Tel.: 089 / 679090 40



MuniQSoft GmbH  
**Bereich Consulting, Support,  
Entwicklung und Vertrieb**  
Witneyst. 1  
82008 Unterhaching  
Tel.: 089 / 6228 6789 0

# Ist ein Index IMMER besser ?

- ◆ **Nicht jeder Ausführungsplan wird durch Index Zugriff anstelle eines Full Table Scans schneller**
- ◆ **Das einzige, das zählt ist die**
  - ▶ **Ausführungszeit !**
  - ▶ **Ausführungszeit !**
  - ▶ **Ausführungszeit !**

# Fall 1: Index ist unusable

◆ Sie haben einen Index, nur ist der leider *unbrauchbar*

◆ Beispiel:

▶ ALTER TABLE emp  
MOVE;

▶ SELECT 1 FROM emp  
WHERE empno=7369;

OPERATION	OBJECT_NAME	OPTIONS	COST
SELECT STATEMENT			3
TABLE ACCESS	EMP	FULL	3
Filter Predicates		EMPNO=7369	

◆ Lösung:

▶ ALTER INDEX pk\_emp **REBUILD;**

▶ SELECT 1 FROM emp  
WHERE empno=7369;

OPERATION	OBJECT_NAME	OPTIONS	COST
SELECT STATEMENT			1
TABLE ACCESS	EMP	BY INDEX ROWID	1
INDEX	PK_EMP	UNIQUE SCAN	0
Access Predicates		EMPNO=7369	

# Fall 2: Es gibt keine Statistiken

◆ Sie haben keine aktuellen Statistiken

◆ Lösung:

▶ Index Statistiken erzeugen:

▶ BEGIN

```
dbms_stats.gather_index_stats(  
  '<OWNER>', '<IND_NAME>');
```

```
END;
```

▶ Tabellen Statistiken erzeugen:

▶ BEGIN

```
dbms_stats.gather_index_stats(  
  '<OWNER>', '<TAB_NAME>');
```

```
END;
```

# Fall 3a: Es gibt keine (korrekten) Histogramme

- ◆ Die Werte in der indizierten Spalte treten ganz unterschiedlich oft auf
- ◆ Es liegen keine korrekten Histogramminformationen vor
- ◆ Ermittlung der Verteilungsstatistiken

```
▶ BEGIN
  dbms_stats.gather_table_stats(
    '<OWNER>', '<TAB_NAME>', method_opt=>
      'FOR ALL INDEXED COLUMNS SIZE AUTO,
      ESTIMATE_PERCENT=>100); -- Kann lange dauern!
  END;
/
```

- ◆ Statistiken dazu stehen in **USER\_HISTOGRAMS**

# Fall 3b: Die Statistiken sind falsch

- ◆ Sie können mit einem Hint, den Optimizer bitten, die Estimated Rows mit den Actual Rows zu vergleichen

- ▶ `SELECT /*#1*/ /*+ GATHER_PLAN_STATISTICS */  
FROM emp WHERE sal=3000;`

- ▶ `WITH s AS (SELECT sql_id FROM v$sqlarea  
WHERE sql_text like '%/*#1*/%')  
SELECT plan_table_output FROM  
s, table(dbms_xplan.display_cursor(  
s.sql_id, null, 'ALLSTATS'))`

Id	Operation	Name	Starts	E-Rows	A-Rows	A-Time	Buffers
0	SELECT STATEMENT		1		2	00:00:00.01	6
* 1	TABLE ACCESS FULL	EMP	1	1	2	00:00:00.01	6

# Fall 4a: Abfrage verwendet LIKE

## ◆ Sie haben die VARCHAR2 Spalte (ename) indiziert

▶ `SELECT * FROM emp  
WHERE ename like 'S%';` -- OK

▶ `SELECT * FROM emp  
WHERE ename like '%S%';`  
-- 11.2.0.3 OK

OPERATION	OBJECT_NAME	OPTIONS	COST
SELECT STATEMENT			1
INDEX	EMP_ENAME_IX	FULL SCAN	1
Filter Predicates			
AND			
ENAME LIKE '%S'			
ENAME IS NOT NULL			

## ◆ Workaround für ältere Versionen

▶ `SELECT /*+ INDEX (emp) */ * FROM emp  
WHERE ename like '%S%';`



# Fall 4b: Abfrage verwendet LIKE

## ◆ Sie haben die NUMBER Spalte (sal) indiziert

- ▶ `SELECT * FROM emp  
WHERE sal like '%33%';  
-- Index wird nicht verwendet`

## ◆ Workaround:

- ▶ Function Based Index anlegen
- ▶ `CREATE INDEX emp_sal_fbi ON  
EMP(to_char(sal));`
- ▶ `select 1 from scott.emp  
where sal like '%33%'`

OPERATION	OBJECT_NAME	OPTIONS	COST
SELECT STATEMENT			1
INDEX	EMP_SAL_FBI	FULL SCAN	1
Filter Predicates			
AND			
TO_CHAR(SAL)			
TO_CHAR(SAL)			

# Fall 4c: Abfrage verwendet LIKE

## ◆ Sie haben die DATE Spalte (hiredate) indiziert

- ▶ `SELECT * FROM emp  
WHERE hiredate like '%1980%'; -- Geht nicht`

## ◆ Workaround: Function Based Index anlegen oder BETWEEN

- ▶ `CREATE INDEX emp_hiredate_fbi ON  
EMP(to_char(hiredate, 'DDMMYYYY'));`

- ▶ `SELECT /*+ INDEX(emp) */ *  
FROM scott.emp  
WHERE hiredate LIKE '%1980%'; --oder`

- ▶ `SELECT * FROM scott.emp  
WHERE hiredate between  
to_date('01.01.80', 'DD.MM.RR')  
and to_date('31.12.80', 'DD.MM.RR');`

# Fall 5: Es wird der falsche Index verwendet

- ◆ Sie haben mehrere Indizes auf der Tabelle, aber der falsche wird verwendet
- ◆ Workaround:
  - ▶ 1. Statistiken aktualisieren, oder
  - ▶ 2. Index Hint mit richtigen Index verwenden, oder
  - ▶ 

```
SELECT /*+ INDEX(emp,mein_index) */ *  
FROM emp  
WHERE sal=3000  
AND comm=1000;
```
  - ▶ 3. Unnötige Indizes von der Tabelle entfernen

# Fall 6: Falsche Reihenfolge indiziert

- ◆ Sie verwenden eine nicht führende Index-Spalte (Index auf ename, sal, hiredate)
- ◆ Workaround:
  - ▶ Reihenfolge ändern oder Index Skip Scan Hint verwenden (Ab 11g kann der Optimizer das automatisch)
  - ▶ `SELECT /*+ INDEX_SS(emp) */ FROM emp WHERE hiredate=to_date('17.12.1980', 'DD.MM.YYYY');`

OPERATION	OBJECT_NAME	OPTIONS	COST
SELECT STATEMENT			2
TABLE ACCESS	EMP	BY INDEX ROWID	2
INDEX	EMP_IND	SKIP SCAN	1
Access Predicates			
HIREDATE=TO_DATE('17.12.80 00:00:00')			
Filter Predicates			
HIREDATE=TO_DATE('17.12.80 00:00:00')			

# Fall 7: Sortierung soll über Index gehen

◆ Index liegt auf SAL

◆ `SELECT * FROM emp  
ORDER BY sal;`

The execution plan for the first query shows a SELECT STATEMENT with a cost of 4. It involves a SORT operation with a cost of 4 and an ORDER BY clause. The SORT operation is supported by a TABLE ACCESS to the EMP table with a cost of 3 and a FULL scan option.

OPERATION	OBJECT_NAME	OPTIONS	COST
SELECT STATEMENT			4
SORT		ORDER BY	4
TABLE ACCESS	EMP	FULL	3

◆ Die Spalte muss über ein NOT NULL Constraint verfügen

◆ `ALTER TABLE emp  
MODIFY (sal NOT NULL);`

The execution plan for the second query shows a SELECT STATEMENT with a cost of 3. It involves a SORT operation with a cost of 3 and an ORDER BY clause. The SORT operation is supported by a TABLE ACCESS to the EMP table with a cost of 2 and a BY INDEX ... option. The TABLE ACCESS is supported by an INDEX on the EMP\_SAL\_FBI table with a cost of 1 and a FULL SCAN option.

OPERATION	OBJECT_NAME	OPTIONS	COST
SELECT STATEMENT			3
SORT		ORDER BY	3
TABLE ACCESS	EMP	BY INDEX ...	2
INDEX	EMP_SAL_FBI	FULL SCAN	1

◆ `SELECT /*+ INDEX(emp) */ * FROM scott.emp  
ORDER BY sal;`

# Fall 8: Stored Outline / SQL Profil

- ◆ **Es gibt eine Stored Outline oder ein SQL-Profil, dass der Optimizer vorzieht**
- ◆ **Workaround:**
- ◆ **Prüfen Sie, ob Ihre Query als Stored Outline oder SQL Profil existiert**
  - ▶ `SELECT * FROM dba_outlines;`
  - ▶ `SELECT * FROM dba_sql_profiles;`
  - ▶ **Löschen Sie ggf. die Stored Outline/ SQL Profil**
  - ▶ `BEGIN DBMS_SQLTUNE.DROP_SQL_PROFILE (name => 'mein_sql_profile'); END;`
  - ▶ `DROP OUTLINE <meine_outline>;`

# Fall 9: Initialisierungsparameter

- ◆ **Sie haben Initialisierungsparameter verwendet, die eine Index-Verwendung für den Optimizer ungünstig erscheinen lassen.**
- ◆ **Beispiele für Pro Full TABLE Scan**
  - ▶ **db\_file\_multiblock\_read\_count (hoch)**
  - ▶ **optimizer\_index\_cost\_adj (hoch (>100))**
  - ▶ **optimizer\_index\_caching (niedrig)**
  - ▶ **optimizer\_mode=all\_rows**
  - ▶ **Hinweis: ca. 300 Initialisierungsparameter werden für einen Ausführungsplan ausgewertet**

# Fall 10: Sie haben gar keinen Index

- ◆ Spalte in der WHERE Bedingung ist gar nicht indiziert
- ◆ Workaround:
  - ▶ Spalte indizieren
  - ▶ 

```
CREATE INDEX scott.my_ind  
ON scott.emp(sal)  
NOLOGGING ONLINE; -- ONLINE nur Enterprise Ed.
```



# MuniQSoft GmbH

## ◆ Tätigkeitsbereiche:

- ▶ **Oracle Support**  
Hotline: Mo-Fr 8.00 – 18.00 Uhr  
Erweiterung um eine Rufbereitschaft auch am Wochenende möglich
- ▶ **Oracle IT Consulting & Services**
- ▶ **Oracle Schulungen (SQL, PL/SQL, DBA, APEX, B&R, ...)**
- ▶ **Software-Lösungen**
- ▶ **Oracle Lizenzen**

MuniQSoft GmbH  
**Bereich Schulung**  
Grünwalder Weg 13a  
82008 Unterhaching  
Tel.: 089 / 679090 40



MuniQSoft GmbH  
**Bereich Consulting, Support,  
Entwicklung und Vertrieb**  
Witneyst. 1  
82008 Unterhaching  
Tel.: 089 / 6228 6789 0