IIT School of Applied Technology

ILLINOIS INSTITUTE OF TECHNOLOGY

information technology & management

526 Data Warehousing

April 6 & 13, 2016

Week 12 & 13 Presentation

Week 12 Topic: SQL Optimization for DW
Introduction of Index

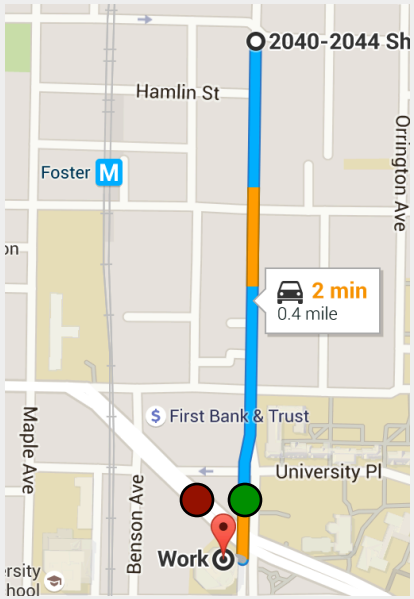
- We will cover
  - Unusable Index
  - External Index Suppression
  - Internal Index Suppression
  - Index with NOT Operator
  - Index with NULL & NOT NULL

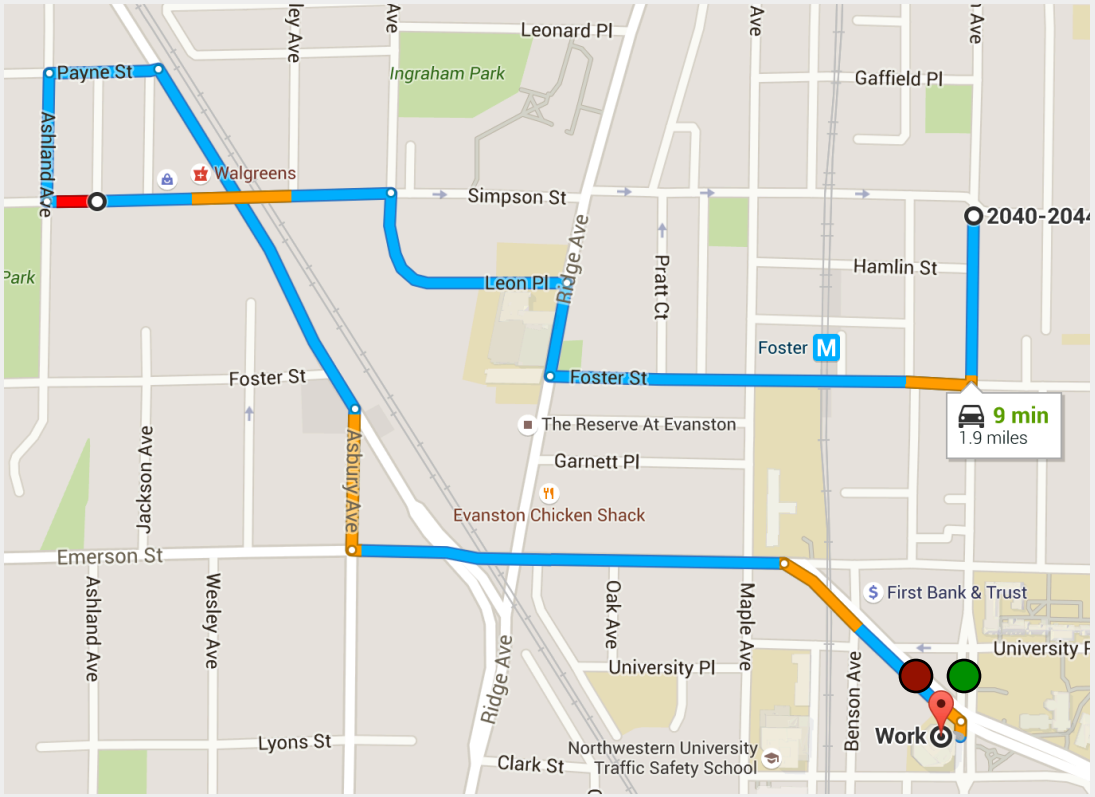
ITMD - 526

Week 12 Topic: SQL Optimization for DW

Optimizer is Not Perfect

- Bad Index
- Good Index





SQL Optimization for DW

Unusable Index

- Index can not be leveraged when…

Index column suppressed

```
SELECT *
  FROM DEPT
 WHERE SUBSTR(DNAME,1,3) = 'ABC'
```

Using NOT operator

```
SELECT *
  FROM EMP
 WHERE JOB <> 'SALES'
```

Using NULL or NOT NULL

```
SELECT *
  FROM EMP
 WHERE ENAME IS NOT NULL
```

Another index is preferred

```
SELECT *
  FROM EMP
 WHERE JOB LIKE 'AB%'
   AND EMPNO = '7890'
```

ITMD - 526

SQL Optimization for DW

External Index Suppression

```
SELECT *
  FROM EMP
 WHERE SUBSTR(DNAME,1,3) = 'ABC'
```

→

```
SELECT *
  FROM EMP
 WHERE DNAME LIKE 'ABC%'
```

```
SELECT *
  FROM EMP
 WHERE SAL * 12 = 12000000
```

→

```
SELECT *
  FROM EMP
 WHERE SAL = 12000000 / 12
```

```
SELECT *
  FROM EMP
 WHERE TO_CHAR(HIREDATE, 'YYMMDD')
                        = '20140101'
```

→

```
SELECT *
  FROM EMP
 WHERE HIREDATE =
  TO_DATE('20140101','YYYYMMDD')
```

```
SELECT *
  FROM EMP
 WHERE NVL(COMM,0) < 100
```

→ ?

ITMD - 526

SQL Optimization for DW (cont'd)

External Index Suppression

```
SELECT *
  FROM EMP
 WHERE EMPNO BETWEEN 100 AND 200
   AND NVL(JOB,'X') = 'CLERK'
```

```
SELECT *
  FROM EMP
 WHERE EMPNO BETWEEN 100 AND 200
   AND JOB = 'CLERK'
```

```
SELECT *
  FROM EMP
 WHERE DEPTNO||JOB = '10SALESMAN'
```

```
SELECT *
  FROM EMP
 WHERE DEPTNO = '10'
   AND JOB = 'SALSMAN'
```

Intentional Suppression

```
SELECT *
  FROM EMP
 WHERE JOB = 'MANAGER'
```

```
SELECT *
  FROM  EMP
 WHERE RTRIM(JOB) = 'MANAGER'
```

```
SELECT *
  FROM EMP
 WHERE EMPNO = 8978
```

```
SELECT *
  FROM EMP
 WHERE RTRIM(EMPNO) = 8978
```

SQL Optimization for DW (cont'd)

External Index Suppression

```
SELECT CUSTNO, INVDATE
  FROM INVOICE
 WHERE CUSTNO = 'DN02'
   AND STATUS = '90'
```

4 rows, 0.51 sec

```
TABLE ACCESS BY ROWID INVOICE
    AND-EQUAL
        INDEX RANGE SCAN INV_STATUS
        INDEX RANGE SCAN INV_CUSTNO
```

```
SELECT CUSTNO, INVDATE
  FROM INVOICE
 WHERE CUSTNO = 'DN02'
   AND RTRIM(STATUS) = '90'
```

4 rows, 0.03 sec

```
TABLE ACCESS BY ROWID INVOICE
    INDEX RANGE SCAN INV_CUSTNO
```

```
SELECT CUSTNO, INVDATE
  FROM INVOICE
 WHERE CUSTNO LIKE 'DN%'
   AND STATUS LIKE '9%'
```

rows, sec

```
SELECT CUSTNO, INVDATE
  FROM INVOICE
 WHERE CUSTNO LIKE 'DN%'
   AND RTRIM(STATUS) LIKE '9%'
```

rows, sec

SQL Optimization for DW (cont'd)

External Index Suppression


```
SELECT X.CUSTNO, INVDATE, CUSTNAME
  FROM SALES01 X, SALES02 Y
 WHERE X.SALENO = Y.SALENO
   AND X.SALEDEPT = '710'
   AND Y.SALEDATE LIKE '201411%'
```

```
SELECT X.CUSTNO, INVDATE, CUSTNAME
  FROM SALES01 X, SALES02 Y
 WHERE X.SALENO = Y.SALENO
   AND RTRIM(X.SALEDEPT) = '710'
   AND Y.SALEDATE LIKE '201411%'
```

```
SELECT X.ORDNO, ORDDATE, ITEM
  FROM ORDER01 X, ORDER02 Y
 WHERE X.ORDNO = Y.ORDNO
   AND X.ORDDATE LIKE '201411%'
   AND Y.ORDDEPT = '710'
 ORDER BY ORDDATE
```

```
SELECT X.ORDNO, ORDDATE, ITEM
  FROM ORDER01 X, ORDER02 Y
 WHERE RTRIM(X.ORDNO) = Y.ORDNO
   AND X.ORDDATE LIKE '201411%'
   AND Y.ORDDEPT = '710'
```

ITMD - 526

SQL Optimization for DW

Internal Index Suppression

```
SELECT *
  FROM ATABLE
 WHERE CHA = 10
```

```
SELECT *
  FROM ATABLE
 WHERE TO_NUMBER(CHA) = 10
```

```
SELECT *
  FROM ATABLE
 WHERE NUM LIKE '201410%'
```

```
SELECT *
  FROM ATABLE
 WHERE TO_CHAR(NUM) LIKE '201410%'
```

```
SELECT *
  FROM ATABLE
 WHERE DAT = '01-JAN-2014'
```

```
SELECT *
  FROM ATABLE
 WHERE DAT = TO_DATE('01-JAN-2014')
```

ITMD - 526

SQL Optimization for DW (cont'd)

Internal Index Suppression

```
SELECT *
  FROM ATABLE
 WHERE VAR = 10
```

```
SELECT *
  FROM ATABLE
 WHERE TO_NUMBER(VAR) = 10
```

```
SELECT *
  FROM ATABLE
 WHERE NUM = CHA
```

```
SELECT *
  FROM ATABLE
 WHERE NUM = TO_CHAR(NUM)
```

```
SELECT *
  FROM ATABLE
 WHERE DAT = '01-JAN-2014'
```

```
SELECT *
  FROM ATABLE
 WHERE DAT = TO_DATE('01-JAN-2014')
```

```
SELECT *
  FROM ATABLE
 WHERE DAT = CHA
```

```
SELECT *
  FROM ATABLE
 WHERE DAT = TO_DATE(CHA)
```

ITMD - 526

SQL Optimization for DW (cont'd)

Internal Index Suppression

```
SELECT SUM(UNCOST)
  FROM INVOICE
 WHERE STATUS = 90
```

1 row, 28.5 sec

```
TABLE ACCESS FULL INVOICE
```

```
SELECT SUM(UNCOST)
  FROM INVOICE
 WHERE STATUS = '90'
```

1 row, 0.15 sec

```
SORT AGGREGATE
  TABLE ACCESS BY ROWID INVOICE
    INDEX RANGE SCAN INV_STATUS
```

```
SELECT INVNO, CUSTNO, UNCOST
  FROM INVOICE
 WHERE CFMDEPT LIKE '71%'
```

rows, sec

```
TABLE ACCESS FULL INVOICE
```

Number

```
SELECT ORDNO, INVNO, STATUS
  FROM ORDER1T X, INVOICE Y
 WHERE X.CUSTNO = Y.CUSTNO
   AND X.ORDDEPT = Y.CFMDEPT
   AND y.INVDATE LIKE '201411%'
```

rows, 71 sec

```
NESTED LOOPS
  TABLE ACCESS  FULL  ORDER1T
  TABLE ACCESS BY ROWID INVOICE
    INDEX RANGE SCAN INV_CFMDEPT
```

ITMD - 526

SQL Optimization for DW

Index with NOT Operator

```
SELECT 'Not fornd !' INTO :COL1
  FROM EMP
 WHERE EMPNO <> '1234'
```

```
SELECT 'OK'  INTO :COL1
  FROM DUAL
 WHERE NOT EXISTS (SELECT '' FROM EMP
                    WHERE EMPNO = '1234')
```

```
SELECT *
  FROM EMP
 WHERE ENAME LIKE 'JA%'
   AND JOB <> 'SALES'
```

```
SELECT *
  FROM EMP a
 WHERE a.ENAME LIKE 'JA%'
   AND NOT EXISTS (SELECT '' FROM EMP b
                    WHERE a.ENAME = b.ENAME
                      AND b.JOB = 'SALES')
```

```
SELECT *
  FROM EMP
 WHERE ENAME LIKE 'JA%'
 MINUS
SELECT *
  FROM EMP b
 WHERE b.JOB = 'SALES'
```

SQL Optimization for DW

Index with NULL & NOT NULL


```
SELECT *
  FROM EMP
 WHERE ENAME IS NOT NULL
```

```
SELECT *
  FROM EMP
 WHERE ENAME > ' '
```

```
SELECT *
  FROM EMP
 WHERE COMM IS NOT NULL
```

```
SELECT *
  FROM EMP
 WHERE COMM > 0
```

```
SELECT *
  FROM EMP
 WHERE COMM IS NULL
```

ITMD - 526

SQL Optimization for DW

Optimizer Decision Factors

RANKING

```
SELECT *
  FROM EMP
 WHERE ENAME LIKE 'AB%'
   AND EMPNO = '7890'
```

Use EMPNO Index Only

Avoid INDEX Merge

```
SELECT *
  FROM EMP
 WHERE ENAME LIKE 'AB%'
   AND JOB LIKE 'SA%'
```

User either ENAME or JOB index, Or Full scan

Low COST

```
SELECT *
  FROM EMP
 WHERE EMPNO > '10'
```

FULL Table Scan

HINT

```
SELECT /*+ INDEX(EMP JOB_IDX) */ *
  FROM EMP
 WHERE ENAME LIKE 'AB%'
   AND JOB LIKE 'SA%'
```

Use JOB Index Only

SQL Optimization for DW

Optimizer: Cost vs. Rule Based

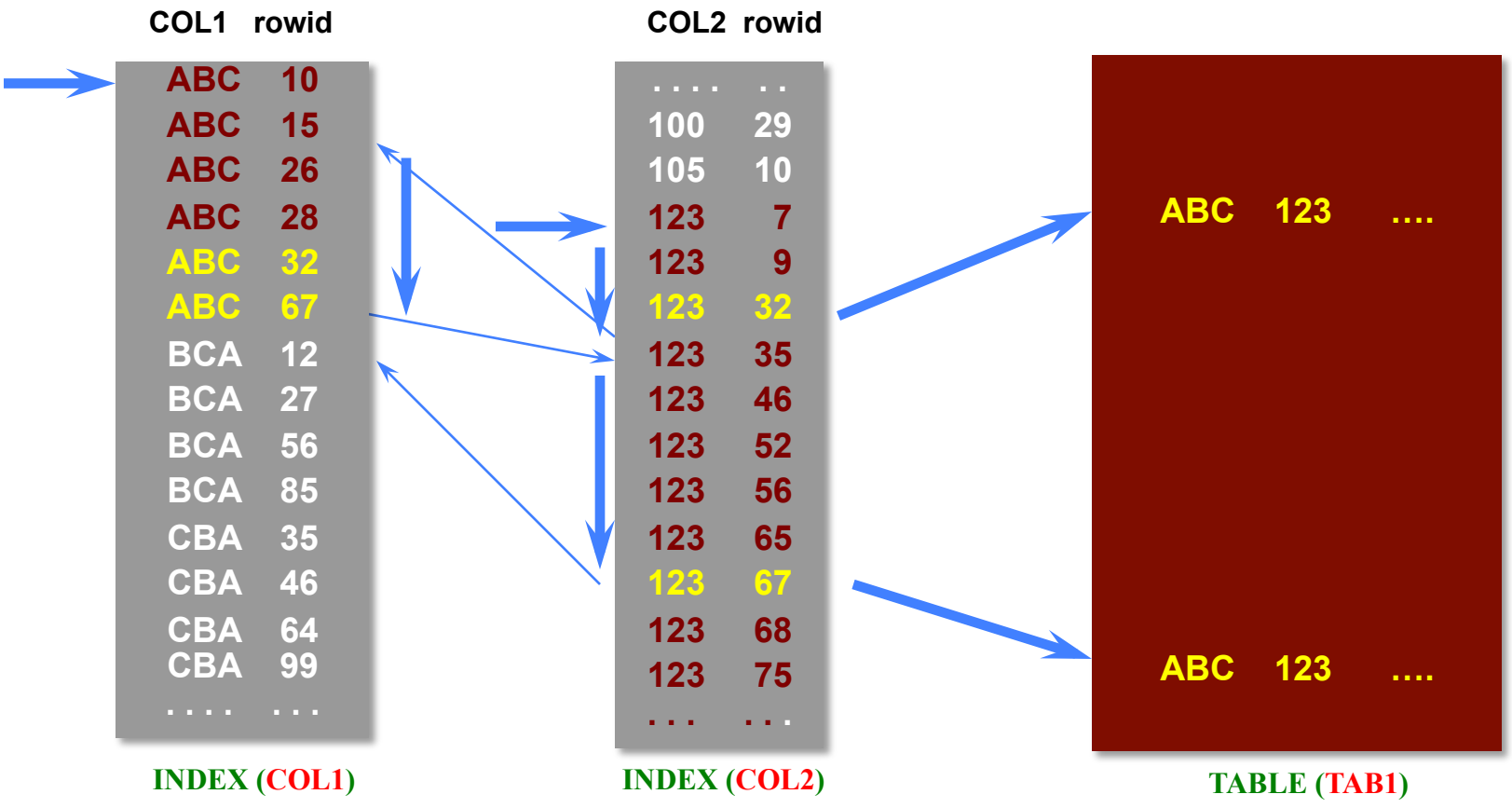
Rule Based		Cost Based
INDEX merge (and_equel)	`SELECT * FROM EMP WHERE JOB = 'SALESMAN' AND EMPNO = '7890'`	INDEX merge (and_equel) or one Index
EMP Index	`SELECT * FROM EMP WHERE ENAME LIKE 'AB%' AND EMPNO = '7890'`	EMP or ENAME Index (depends on stats)
Index created recently	`SELECT * FROM EMP WHERE ENAME LIKE 'AB%' AND JOB LIKE 'SA%'`	EMP or ENAME Index or full scan (depends on stats)

ITMD - 526

SQL Optimization for DW

Index Merge

```
SELECT  COL1, COL2, .....
  FROM  TAB1
WHERE  COL1 = 'ABC'   AND  COL2  = '123'  ;
```



ITMD - 526

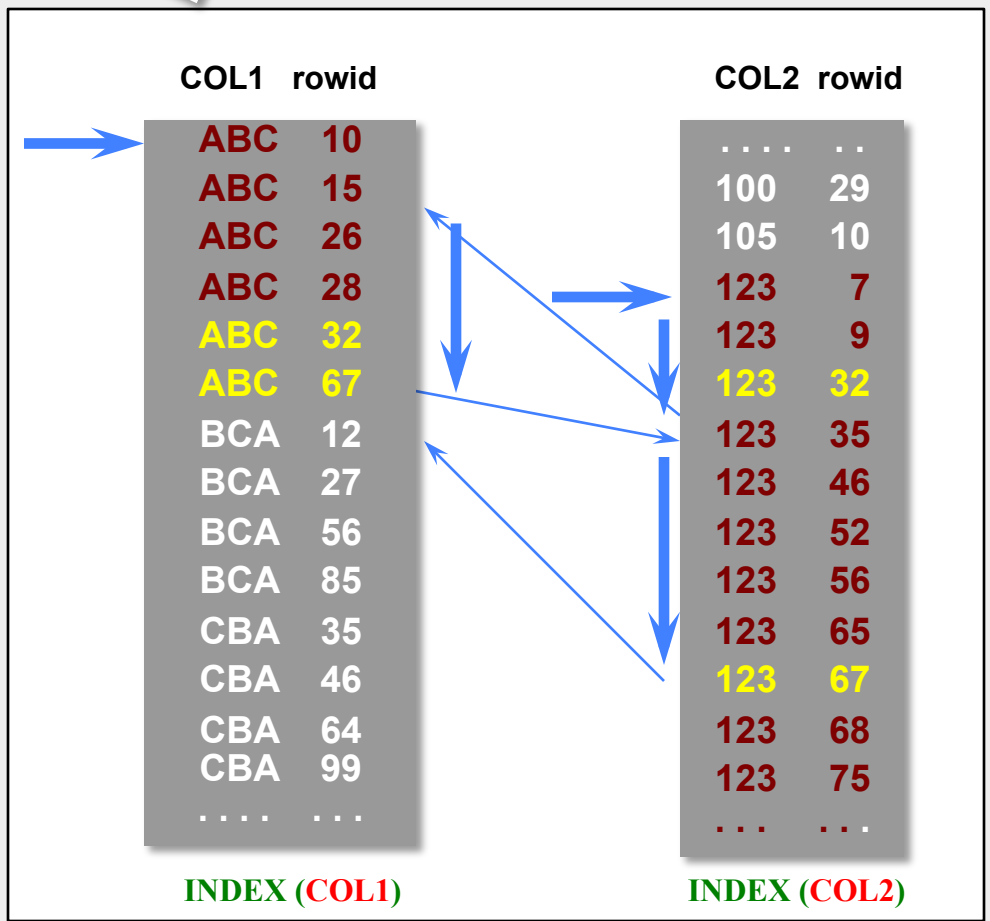
SQL Optimization for DW

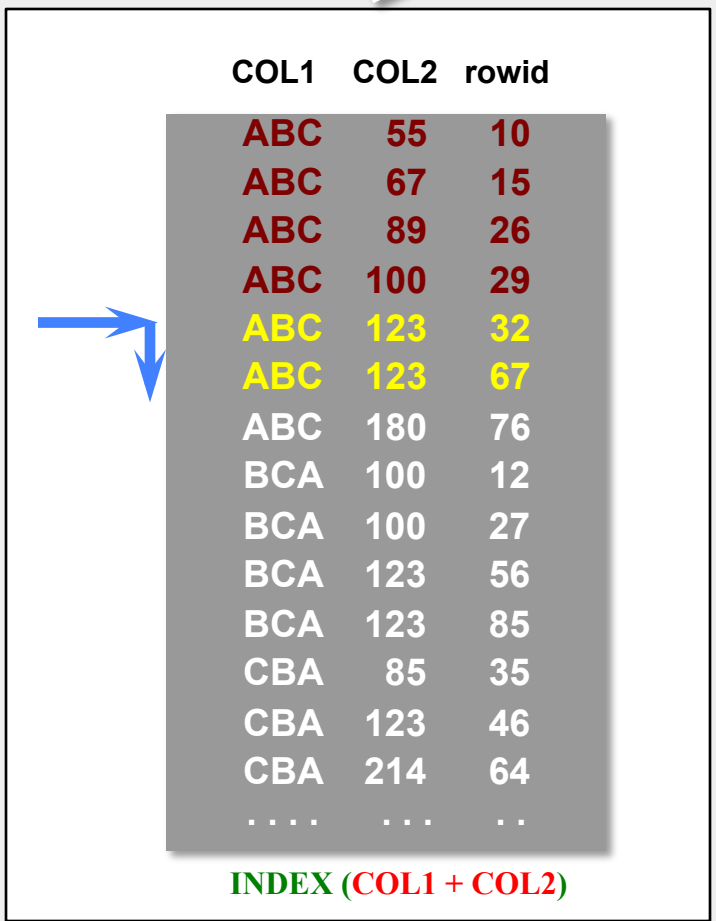
Index Merge vs. Composite Index

Index Merge

```
SELECT  COL1, COL2, .....
  FROM  TAB1
WHERE  COL1 = 'ABC'  AND  COL2  = '123'  ;
```

Composite Index



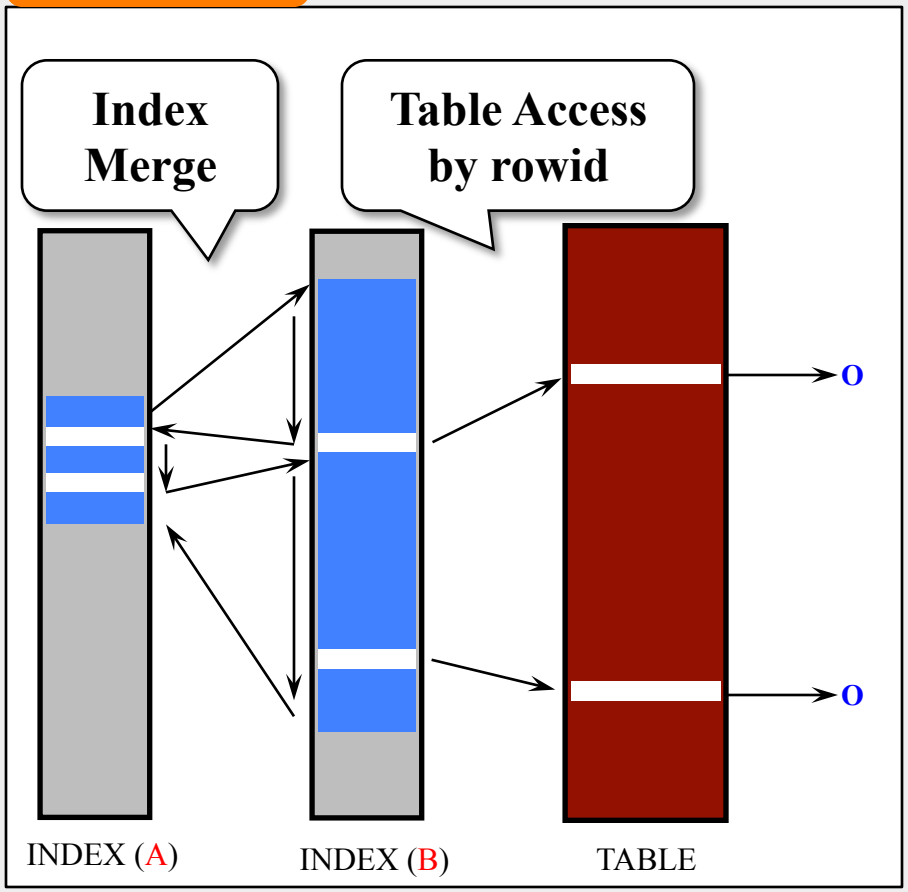


SQL Optimization for DW

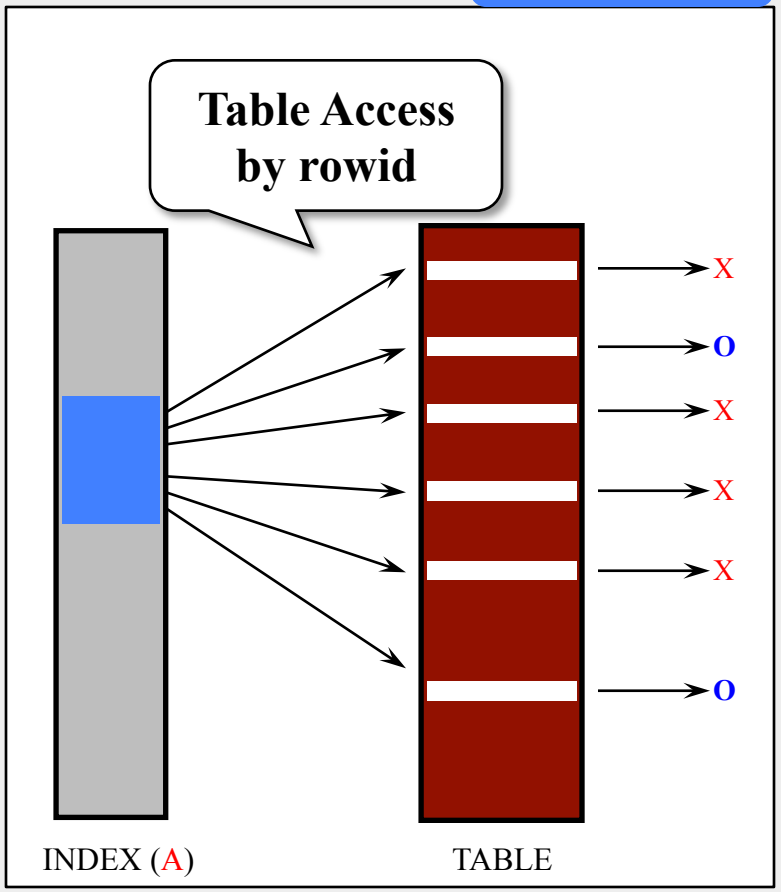
Index Merge - Suboptimal


```
SELECT  COL1, COL2, .....
  FROM  TAB1
WHERE  A = 'ABC'   AND  B LIKE '12%';
```

Suboptimal



Optimal

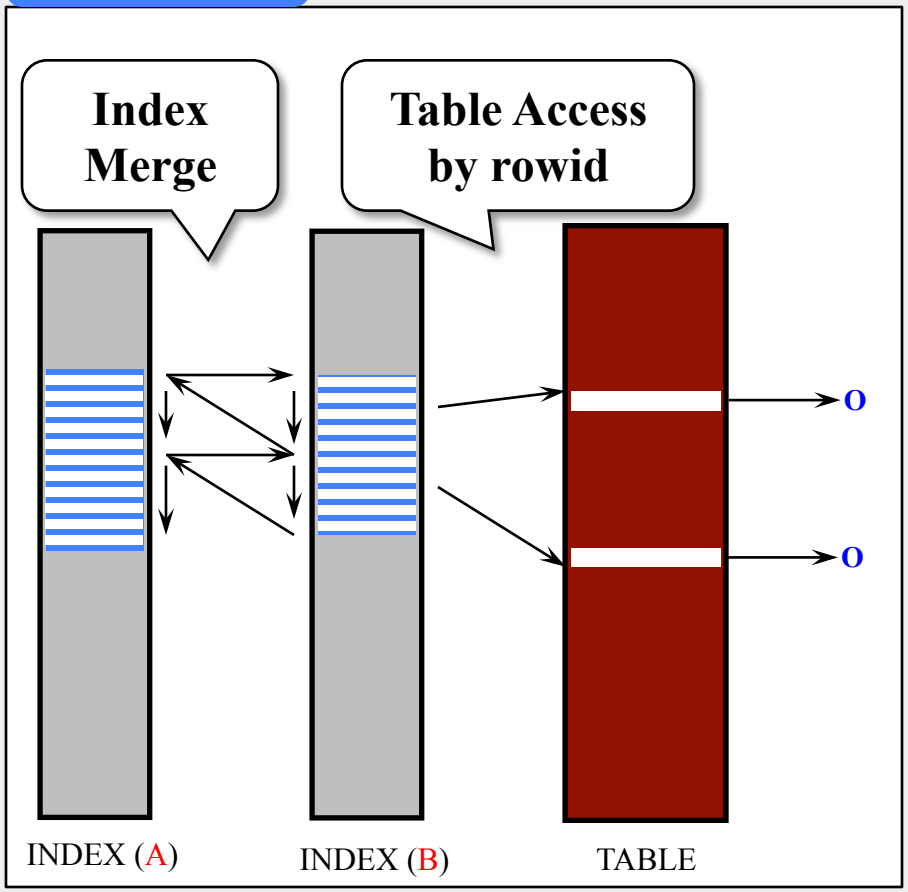


SQL Optimization for DW

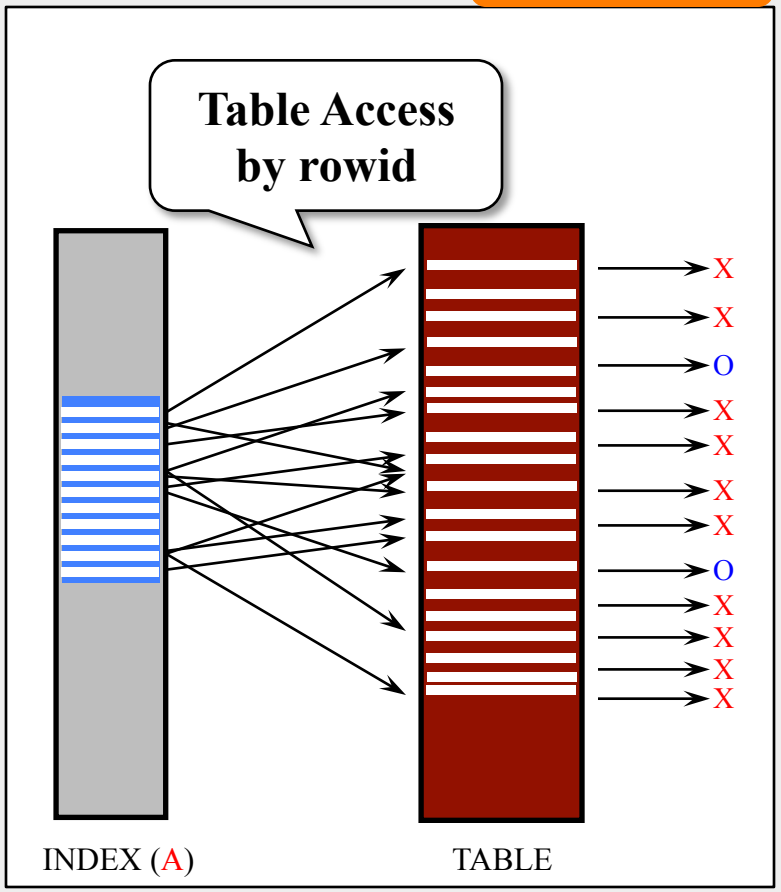
Index Merge – Optimal


```
SELECT  COL1, COL2
   FROM TAB1
WHERE A = 'ABC' AND B = '123';
```

Optimal



Suboptimal

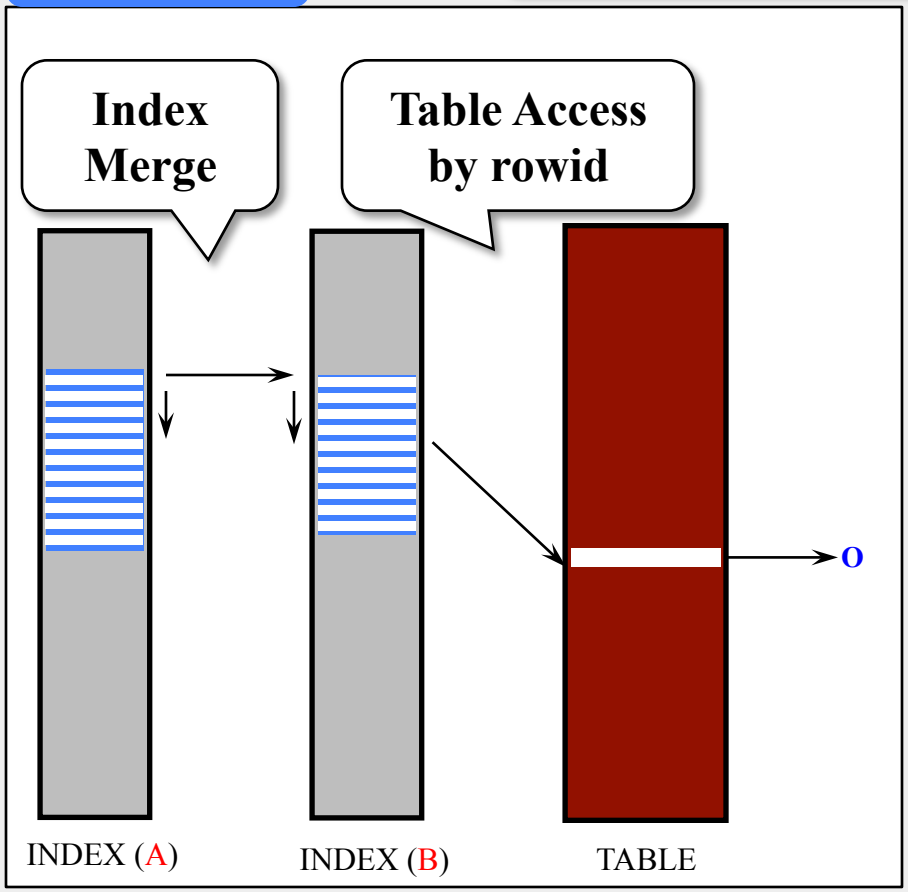


SQL Optimization for DW

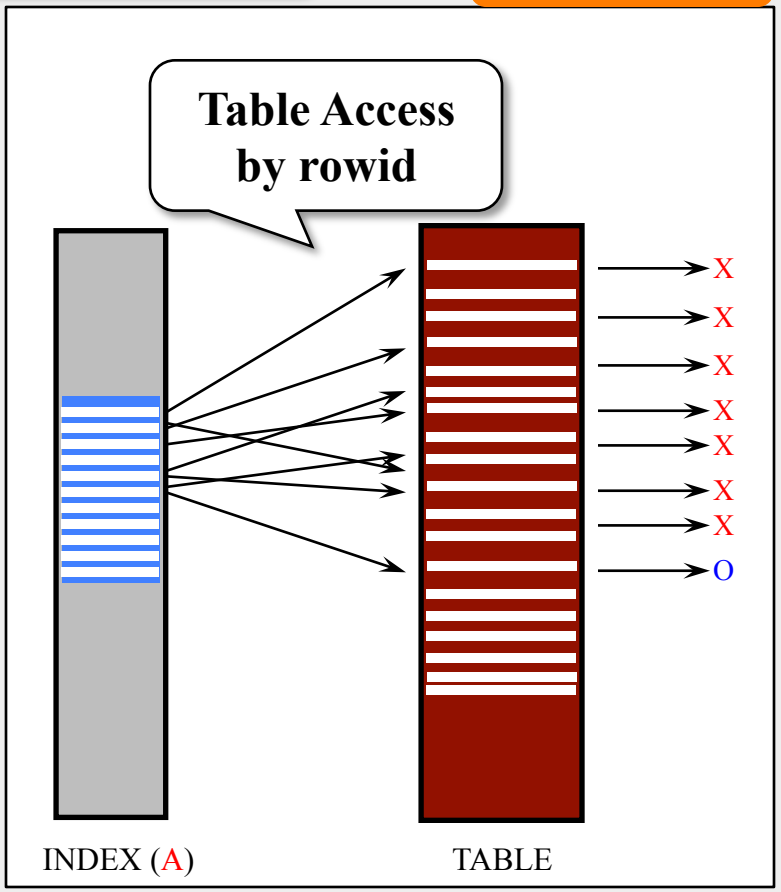
Index Merge – Optimal (cont'd)


```
SELECT  'X' FROM DUAL
WHERE  EXISTS (
          SELECT 'X' FROM TAB1
           WHERE A = 'ABC' AND B = '123');
```

Optimal



Suboptimal



ITMD - 526

SQL Optimization for DW
Index Merge Examples

```
SELECT CUSTNO, INVDATE
  FROM INVOICE
 WHERE CUSTNO = 'DN02'
   AND INVDATE = '941003'
```

1 row, 0.02 sec

```
TABLE ACCESS BY ROWID INVOICE
  AND-EQUAL
    INDEX RANGE SCAN CH_CUSTNO
    INDEX RANGE SCAN CH_INVDATE
```

```
SELECT CUSTNO, INVDATE
  FROM INVOICE
 WHERE CUSTNO = 'DN02'
   AND INVDATE LIKE '201410%
```

1 row, 0.02 sec

```
TABLE ACCESS BY ROWID INVOICE
   INDEX RANGE SCAN CH_CUSTNO
```

```
SELECT /*+ AND_EQUAL(A ch_custno
           ch_status) */
       CUSTNO, INVDATE
  FROM INVOICE A
 WHERE CUSTNO = 'DN02'
   AND STATUS = '90'
```

4 rows 0.74 sec

```
TABLE ACCESS BY ROWID INVOICE
  AND-EQUAL
    INDEX RANGE SCAN  CH_STATUS
    INDEX RANGE SCAN  CH_CUSTNO
```

```
SELECT /*+ INDEX(A ch_custno) */
       CUSTNO, INVDATE
  FROM INVOICE A
 WHERE CUSTNO = 'DN02'
   AND STATUS = '90'
```

4 rows, 0.02 sec

```
TABLE ACCESS BY ROWID INVOICE
   INDEX RANGE SCAN CH_CUSTNO
```

SQL Optimization for DW

Index Merge vs. Composite Index Examples

➢ Single Column Index

```
SELECT /*+ AND_EQUAL(A ch_invdate
           ch_status) */
       CUSTNO, INVDATE
  FROM INVOICE A
 WHERE INVDATE = '941003'
   AND STATUS = '90'
```

13 rows, 0.21 sec

```
TABLE ACCESS BY ROWID INVOICE
  AND-EQUAL
    INDEX RANGE SCAN  CH_STATUS
    INDEX RANGE SCAN  CH_INVDATE
```

```
SELECT CUSTNO, INVDATE
  FROM INVOICE
 WHERE INVDATE LIKE '201501%'
   AND STATUS = '90'
```

0 row, 6.05 sec

```
TABLE ACCESS BY ROWID INVOICE
    INDEX RANGE SCAN  CH_STATUS
```

➢ Composite Index (INVDATE + STATUS)

```
SELECT CUSTNO, INVDATE
  FROM INVOICE A
 WHERE INVDATE = '20150120'
   AND STATUS = '90'
```

13 rows, sec

```
SELECT CUSTNO, INVDATE
  FROM INVOICE
 WHERE INVDATE LIKE '201501%'
   AND STATUS = '40'
```

rows, sec

ITMD - 526

SQL Optimization for DW

Composite Index with EQUAL

```
SELECT  *
   FROM TAB1
WHERE COL1 = 'A'  AND COL2 = '112'
```

COL1	COL2	ROWID
A	110	10
A	111	11
A	**112**	**5**
A	113	18
A	114	54
A	115	23
A	116	29
A	117	25
A	118	26
A	119	30
A	120	19
A	121	32
B	110	41
B	111	45

INDEX

COL2	COL1	ROWID
110	A	10
110	B	41
110	C	57
110	D	81
111	A	11
111	B	39
111	C	76
111	D	98
112	**A**	**5**
112	B	73
112	C	89
112	D	77
113	A	18
113	B	65

INDEX

ITMD - 526

SQL Optimization for DW

Composite Index with EQUAL (cont'd)

```
SELECT  *
   FROM TAB1
WHERE COL1 = 'A'
    AND COL2 LIKE '111' AND '112'
```

COL1	COL2	ROWID
A	110	10
A	111	11
A	112	5
A	113	18
A	114	54
A	115	23
A	116	29
A	117	25
A	118	26
A	119	30
A	120	19
A	121	32
B	110	41
B	111	45

INDEX

COL2	COL1	ROWID
110	A	10
110	B	41
111	A	11
111	B	65
111	C	96
111	D	78
112	A	5
112	B	73
112	C	79
112	D	97
113	A	18
113	B	45
113	C	67
114	A	22

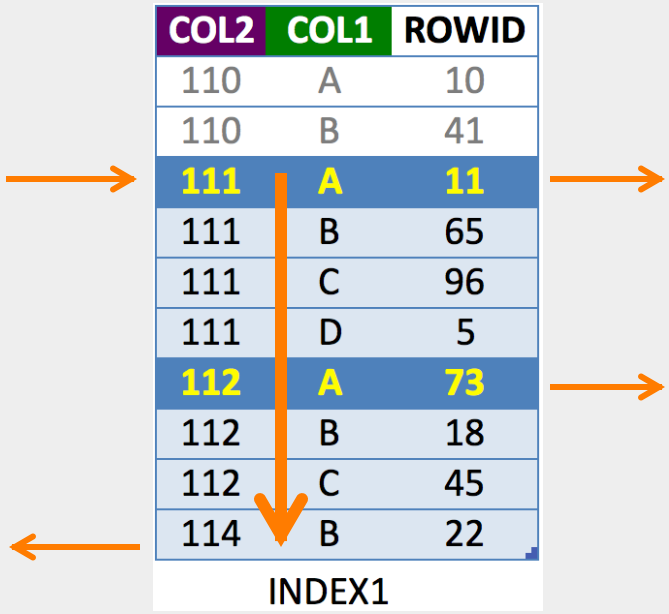
INDEX

ITMD - 526

SQL Optimization for DW

Composite Index: BETWEEN vs. IN

```
SELECT  * FROM TAB1
WHERE COL1 = 'A'
    AND COL2 between '111' AND '112'
```

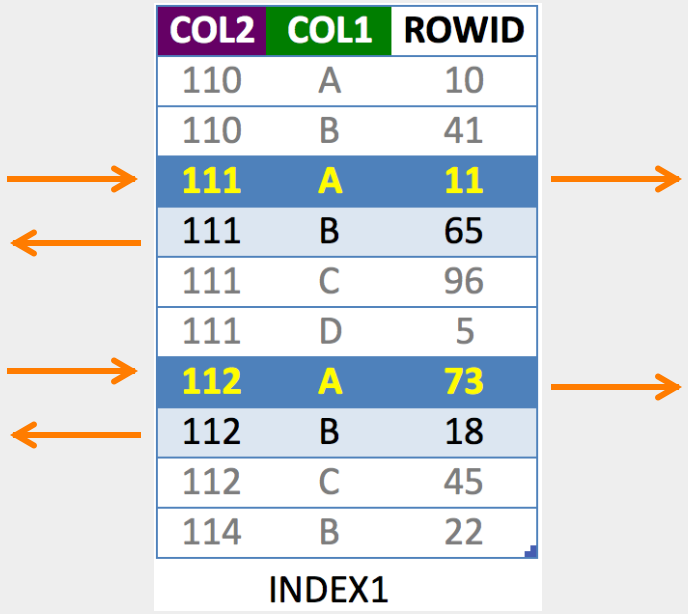


COL2	COL1	ROWID
110	A	10
110	B	41
111	A	11
111	B	65
111	C	96
111	D	5
112	A	73
112	B	18
112	C	45
114	B	22

INDEX1

```
TABLE ACCESS BY ROWID TAB1
   INDEX RANGE SCAN INDEX1
```

```
SELECT  * FROM TAB1
WHERE COL1 = 'A'
    AND COL2 in ('111', '112')
```



COL2	COL1	ROWID
110	A	10
110	B	41
111	A	11
111	B	65
111	C	96
111	D	5
112	A	73
112	B	18
112	C	45
114	B	22

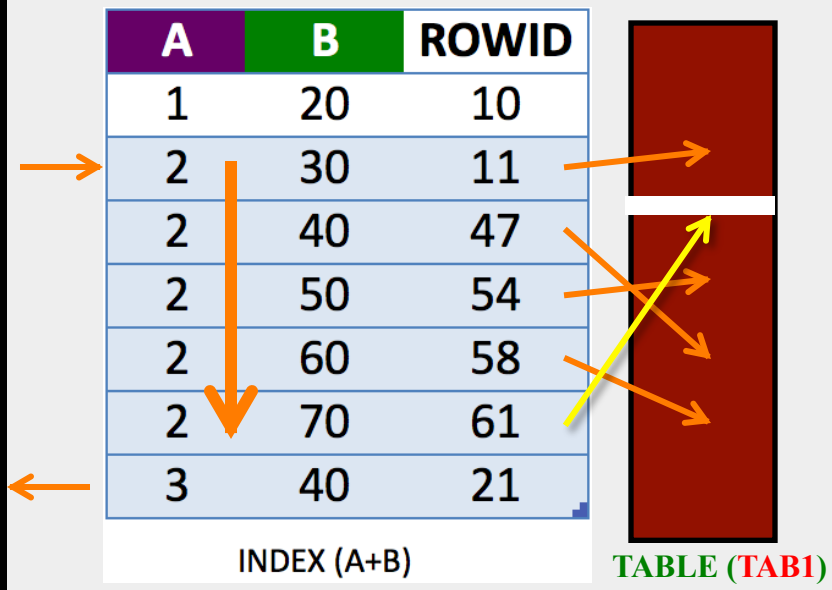
INDEX1

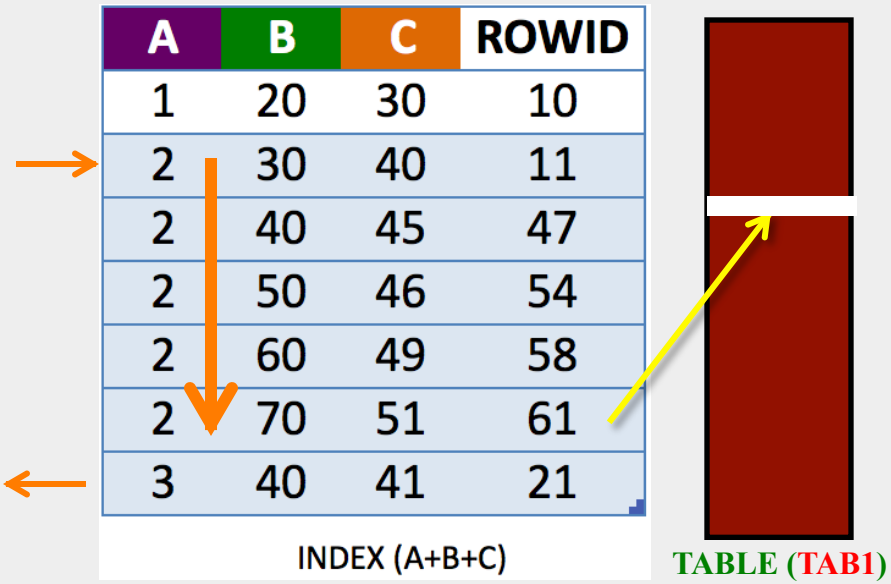
```
CONCATENATION
  TABLE ACCESS BY ROWID TAB1
    INDEX RANGE SCAN  INDEX1
  TABLE ACCESS BY ROWID TAB1
    INDEX RANGE SCAN INDEX1
```

SQL Optimization for DW

Composite Index: Adding a Column


```
SELECT *
  FROM TAB1
 WHERE A = '2' AND C = 51
```

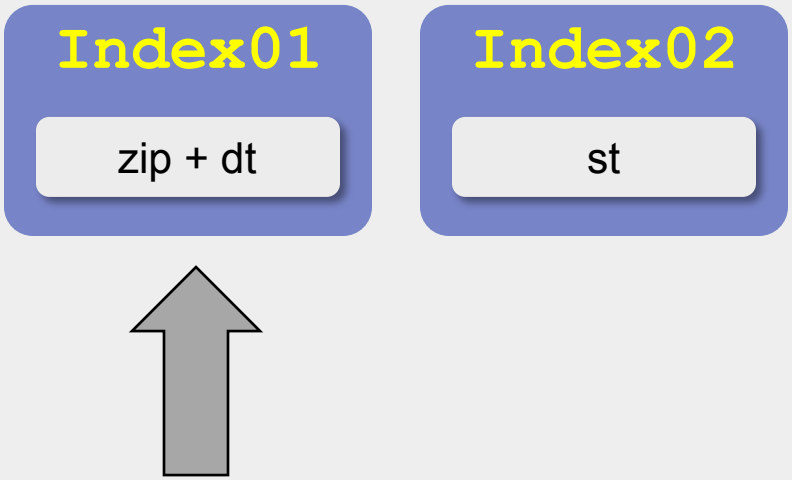




SQL Optimization for DW

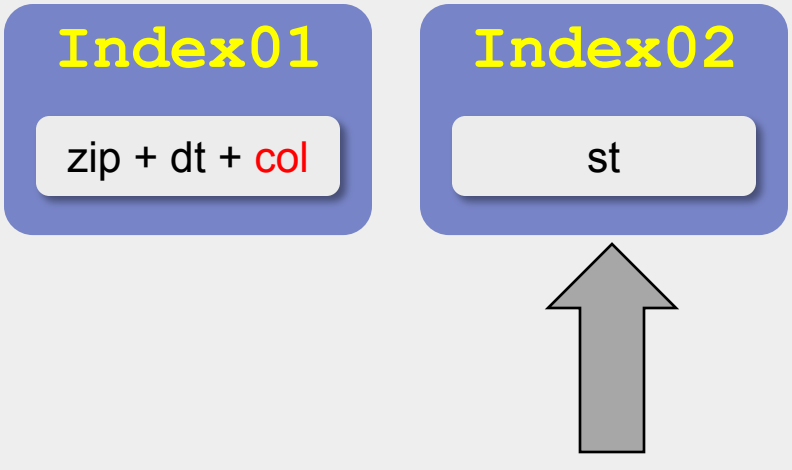
Composite Index: Adding a Column (cont'd)

```
SELECT *
  FROM Sales
 WHERE zip = '60201'
   AND dt = '20141005'
   AND st = 'DELIVERED'
```



Optimizer's Pick

Good



Optimizer's Pick

Bad