IIT School of Applied Technology

ILLINOIS INSTITUTE OF TECHNOLOGY

information technology & management

526 Data Warehousing

March 30, 2016

Week 11 Presentation

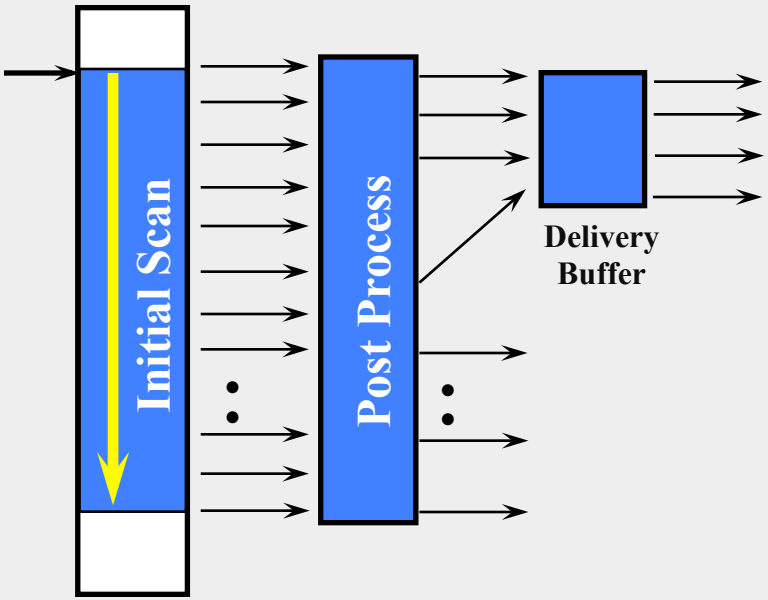
Week 11 Topic: SQL Optimization for DW
Partial vs. Full Range Scan

- We will cover
  - Partial/Full Range Scan Concept
  - Partial vs. Full Range Scan & Examples
  - BI Applications Demo
  - Assignment 02 Specifications

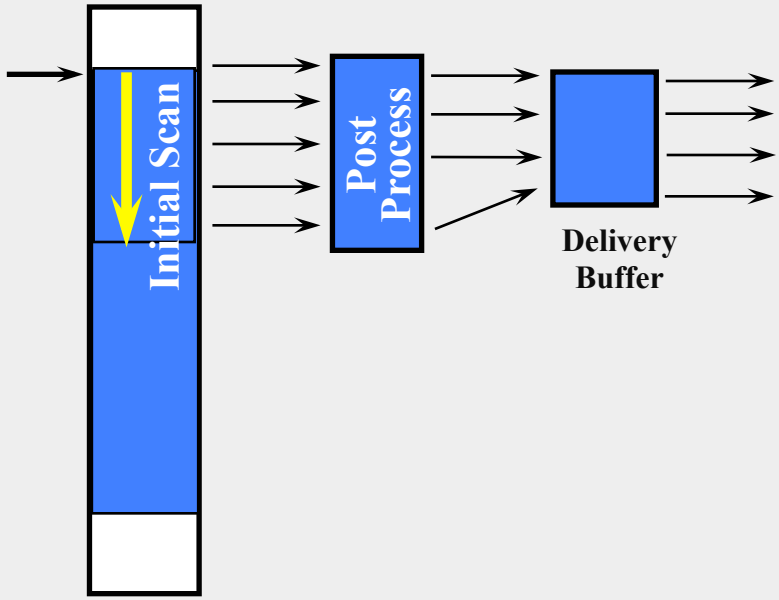
SQL Optimization for DW

Partial vs. Full Range Scan

- Access subset of the entire result before pushing out the first delivery buffer
- Improve response time



Full Range Scan



Partial Range Scan

SQL Optimization for DW

Partial Range Scan

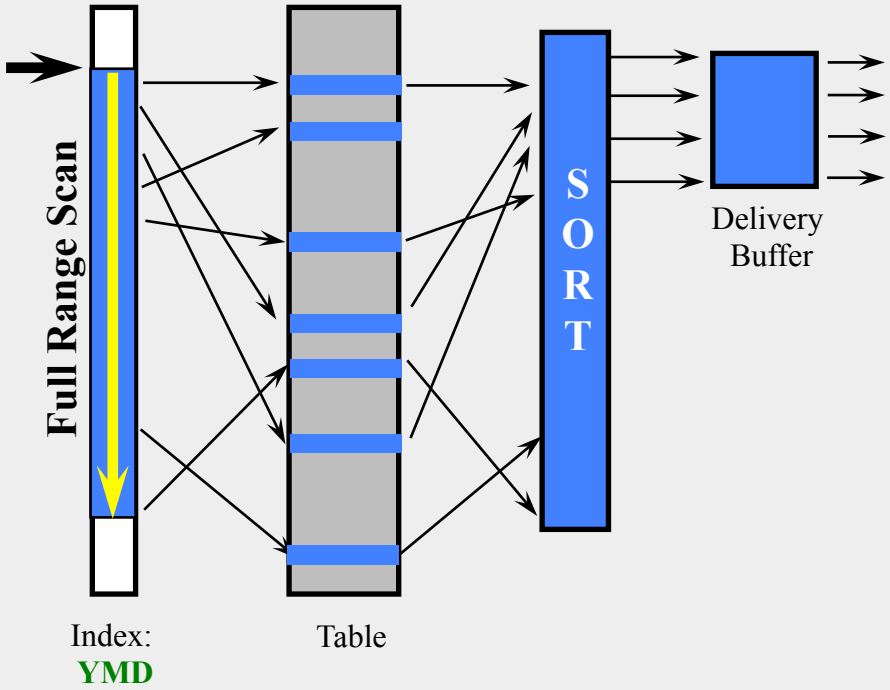
- Using Index to replace SORT operation
- Using Index to replace MAX operation
- Access index file only without accessing the data file
- Using EXISTS
- Using ROWNUM
- Using Stored Function

ITMD - 526

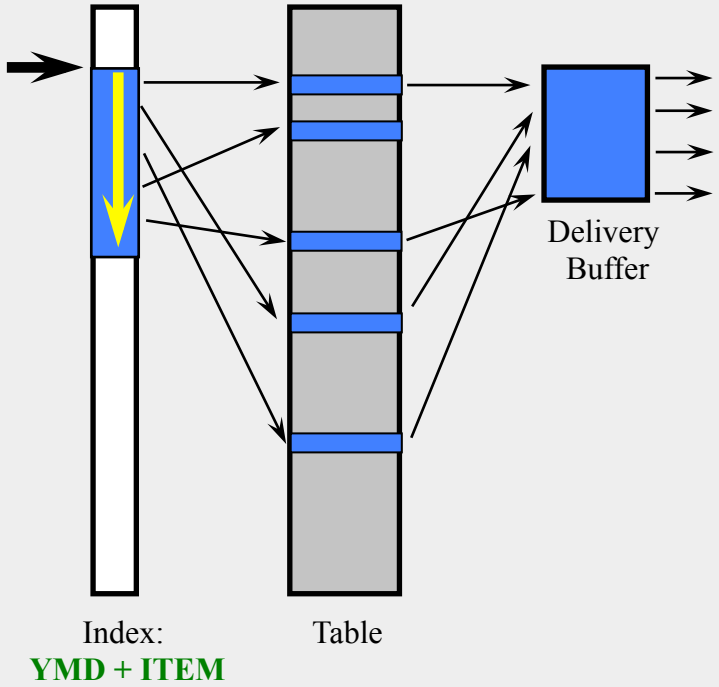
SQL Optimization for DW

Using Index to Replace SORT

```
SELECT *
  FROM PRODUCT
 WHERE YMD = '20151023'
   AND ITEM LIKE 'AB%'
 ORDER BY YMD, ITEM;
```




```
SELECT *
  FROM PRODUCT
 WHERE YMD = '20151023'
   AND ITEM LIKE 'AB%';
```



SQL Optimization for DW
Partial Range Scan: Replacing SORT


```
SELECT ORDDATE, CUSTNO
  FROM ORDER1T
 WHERE ORDDATE  between
       '940101' and '941130'
 ORDER BY ORDDATE  DESC
```

5.2 sec

```
21200   SORT ORDER BY
21200     TABLE ACCESS BY ROWID ORDER1T
21201       INDEX RANGE SCAN ORD_ORDDATE
```

```
SELECT /*+ INDEX_DESC(A orddate) */
       ORDDATE, CUSTNO
  FROM ORDER1T  A
 WHERE ORDDATE  between
       '940101' and '941130'
```

0.01 sec

```
20      INDEX RANGE SCAN DESCENDING ORDDATE
```

```
SELECT ORDDATE, CUSTNO
  FROM ORDER1T
 WHERE ORDDEPT LIKE '7%'
 ORDER BY ORDDATE DESC
```

12.5 sec

```
42000   SORT ORDER BY
42000     TABLE ACCESS BY ROWID ORDER1T
42001       INDEX RANGE SCAN ORD_DEPT
```

```
SELECT /*+ INDEX_DESC(A orddate) */
       ORDDATE, CUSTNO
  FROM ORDER1T A
 WHERE ORDDEPT LIKE '7%'
   AND ORDDATE <= '991231'
```

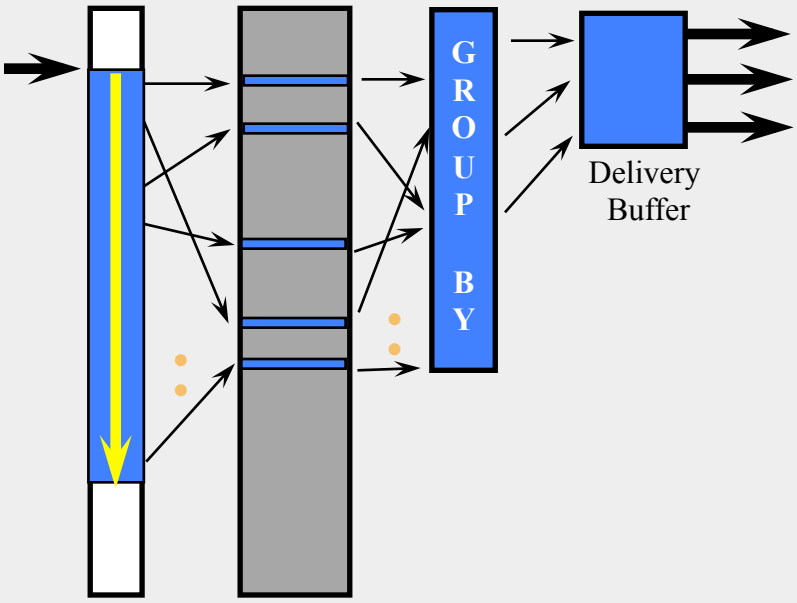
0.02 sec

```
20      INDEX RANGE SCAN DESCENDING ORDDATE
```

SQL Optimization for DW

Partial Range Scan: Access Index Only

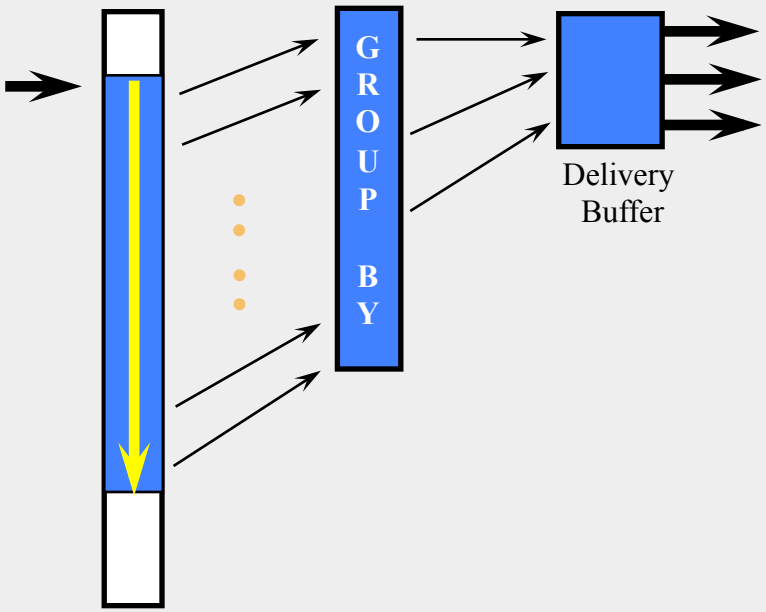

```
SELECT DEPT, SUM(QTY)
  FROM PRODUCT
 WHERE DEPT LIKE '12%'
 GROUP BY DEPT;
```



Index: **DEPT**

Table

```
SELECT DEPT, SUM(QTY)
  FROM PRODUCT
 WHERE DEPT LIKE '12%'
 GROUP BY DEPT;
```



Index: **DEPT + QTY**

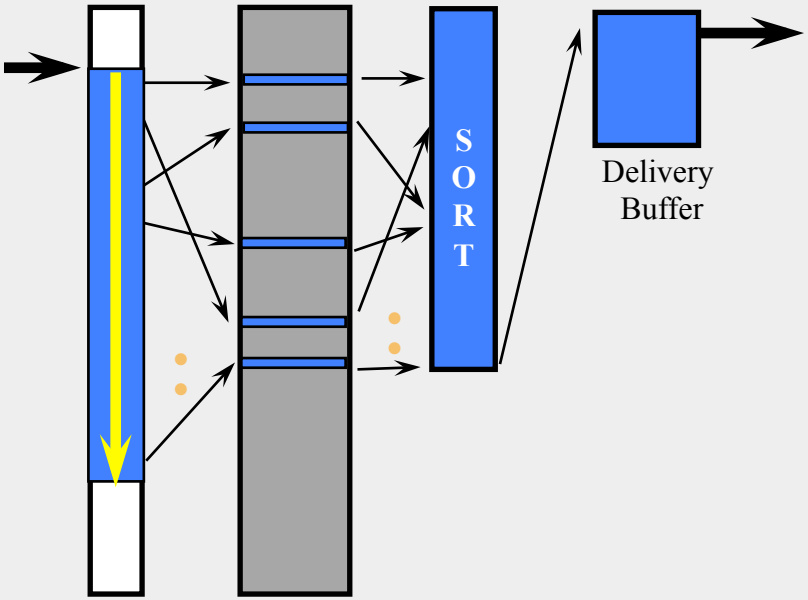
ITMD - 526

SQL Optimization for DW

Partial Range Scan: Handling MAX

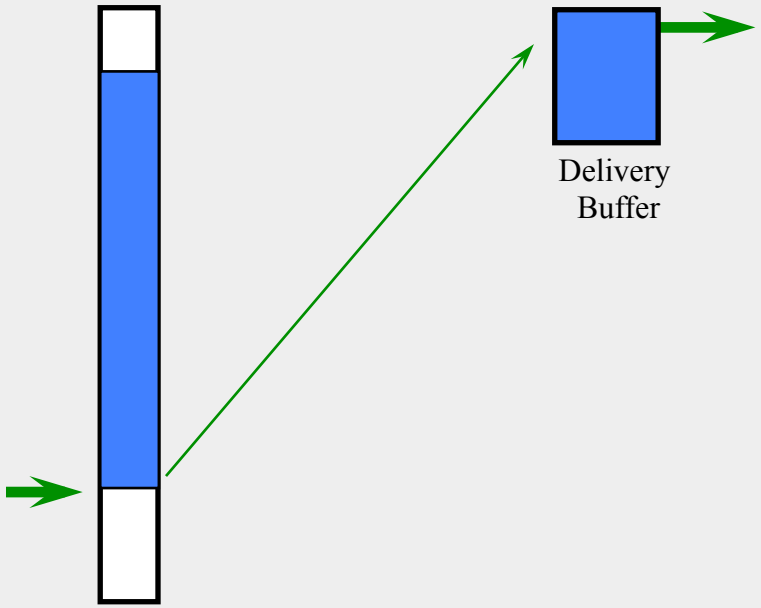
```
SELECT MAX(SEQ) + 1
  FROM PRODUCT
 WHERE DEPT = '12300';
```

```
SELECT /*+ INDEX_DESC(A IX_DEPT_SEQ) */
       A.SEQ+1
  FROM PRODUCT A
 WHERE A.DEPT LIKE '12300'
   AND ROWNUM = 1;
```



Index: **DEPT**

Table



Index: **DEPT + SEQ**

SQL Optimization for DW

Partial Range Scan: SQL Examples


```
SELECT MAX(ORDDATE)
  FROM ORDER1T
 WHERE ORDDEPT = '430'
   AND STATUS  = '30'
```

2.53 sec

```
1       SORT AGGREGATE
2892      TABLE ACCESS BY ROWID ORDER1T
15230       INDEX RANGE SCAN DEPT_DATE
```

```
SELECT /*+ INDEX_DESC(A dept_date) */
       ORDDATE
  FROM ORDER1T  A
 WHERE ORDDEPT = '430' AND STATUS='30'
   AND ROWNUM = 1
```

0.01 sec

```
1 COUNT STOPKEY
1   TABLE ACCESS BY ROWID ORDER1T
2     INDEX RANGE SCAN DESCENDING DEPT_DATE
```

```
SELECT TYPE, COUNT(*)
  FROM ORDER2T
 WHERE ITEM LIKE 'HJ%'
 GROUP BY TYPE
```

10.3 sec

```
20    SORT GROUP BY
36631   TABLE ACCESS BY ROWID ORDER2T
36631     INDEX RANGE SCAN ITEM_STATUS
```

```
SELECT STATUS, COUNT(*)
  FROM ORDER2T
 WHERE ITEM LIKE 'HJ%'
 GROUP BY STATUS
```

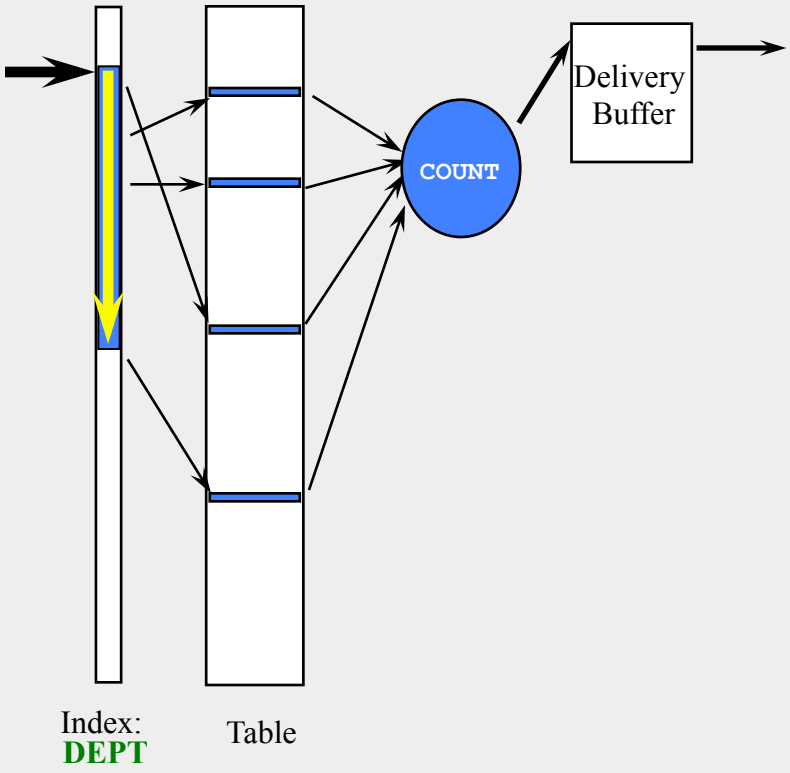
2.5 sec

```
20    SORT GROUP BY
36631   INDEX RANGE SCAN ITEM_STATUS
```

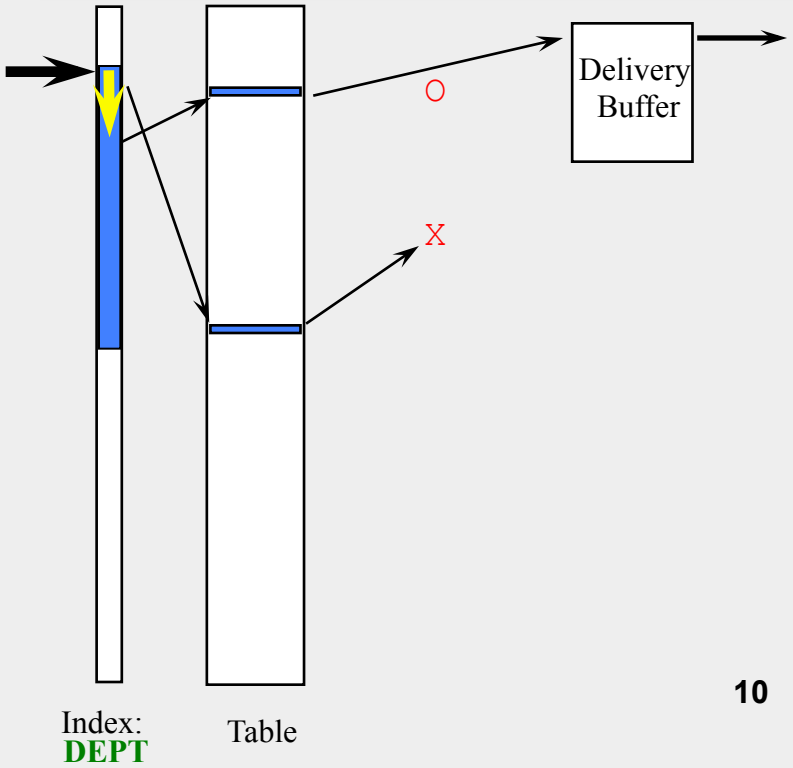
SQL Optimization for DW

Partial Range Scan: EXISTS


```
SELECT COUNT(*) INTO :CNT
  FROM ITEM_TAB
 WHERE DEPT = '101'
   AND SEQ > 100
....
IF CNT > 0 ...
....
```



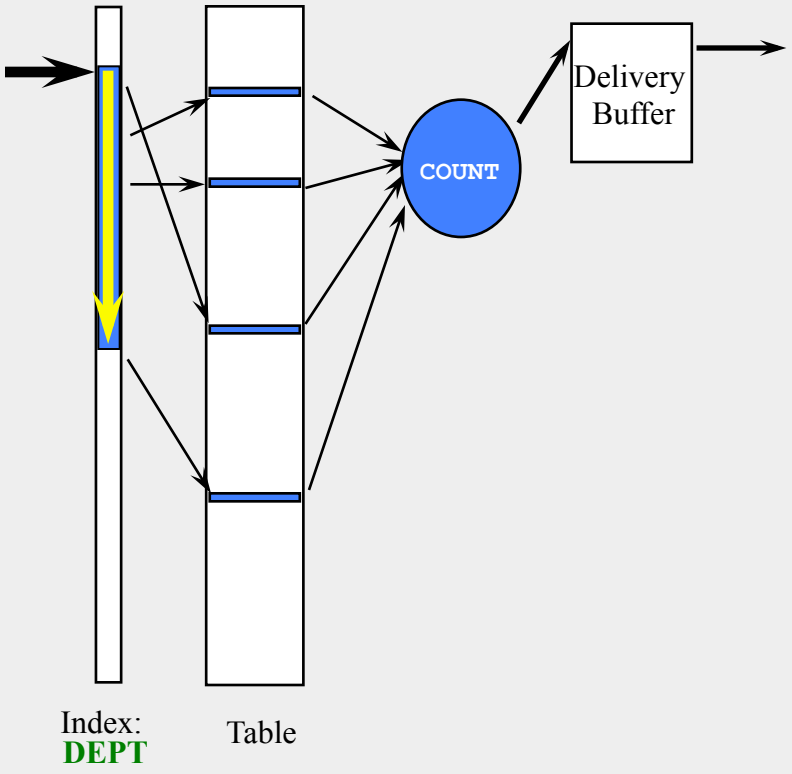

```
SELECT 1 INTO :CNT FROM  DUAL
 WHERE EXISTS
       (SELECT 'X'
          FROM ITEM_TAB
         WHERE DEPT = '101'
           AND SEQ  >  100)
....
IF CNT > 0
....
```



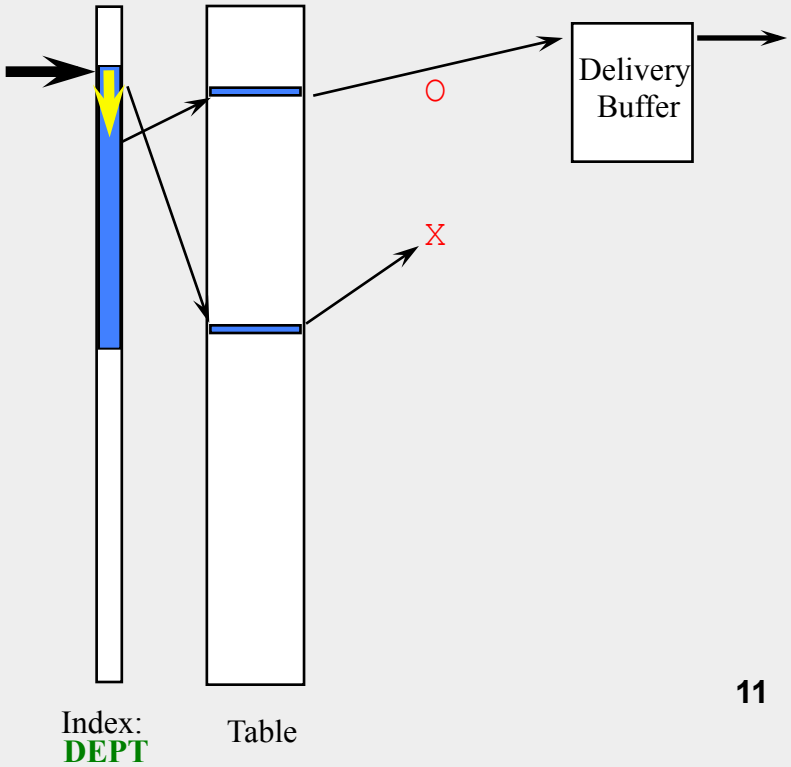
SQL Optimization for DW

Partial Range Scan: ROWNUM


```
SELECT COUNT(*) INTO :CNT
  FROM ITEM_TAB
 WHERE DEPT = '101'
   AND  SEQ > 100
....
IF CNT > 0
....
```




```
SELECT 1 INTO :CNT
  FROM ITEM_TAB
 WHERE DEPT = '101'
   AND SEQ  > 100
   AND ROWNUM = 1
....
IF CNT > 0
....
```



SQL Optimization for DW

Partial Range Scan: 1:M JOIN


```
SELECT x.CUST_NO, x.ADDR, x.NAME, ...
  FROM CUST x, REQT y
 WHERE x.CUST_NO = y.CUST_NO
   AND x.CUST_STAT IN ('A', 'C', 'F')
   AND y.UN_PAY > 0
 GROUP BY x.CUST_NO
HAVING SUM(y.UN_PAY) BETWEEN :VAL1 and :VAL2
```

Throughput

Full Range Scan

Response Time

Partial Range Scan

```
SELECT x.CUST_NO, x.ADDR, x.NAME, ....
  FROM CUST x
 WHERE CUST_STAT in ('A', 'C', 'F')
   AND EXISTS
       (SELECT 'X'
          FROM REQT y
         WHERE y.CUST_NO = x.CUST_NO
           AND UN_PAY > 0
         GROUP BY x.CUST_NO
        HAVING SUM(y.UN_PAY)
               BETWEEN :VAL1 and :VAL2
       )
```

SQL Optimization for DW

Partial Range Scan: 1:M JOIN (cont'd)


```
SELECT x.CUST_NO, SUM(y.UN_PAY) AS UN_PAY,.
  FROM CUST x, REQT y
 WHERE x.CUST_NO = y.CUST_NO
   AND x.CUST_STAT IN ('A', 'C', 'F')
   AND y.UN_PAY > 0
 GROUP BY x.CUST_NO
HAVING SUM(y.UN_PAY) BETWEEN :VAL1 and :VAL2
```

Throughput

Full Range Scan

```
 CREATE or REPLACE FUNCTION unpay_sum
            (v_custno  in varchar2)
      return number is
      sum_unpay number;
begin
    ....
    select sum(un_pay)
           into sum_unpay
    from reqt
    where cust_no = v_custno
     and un_pay > 0;
    ....
    return sum_unpay;
end unpay_sum ;
```

Response Time

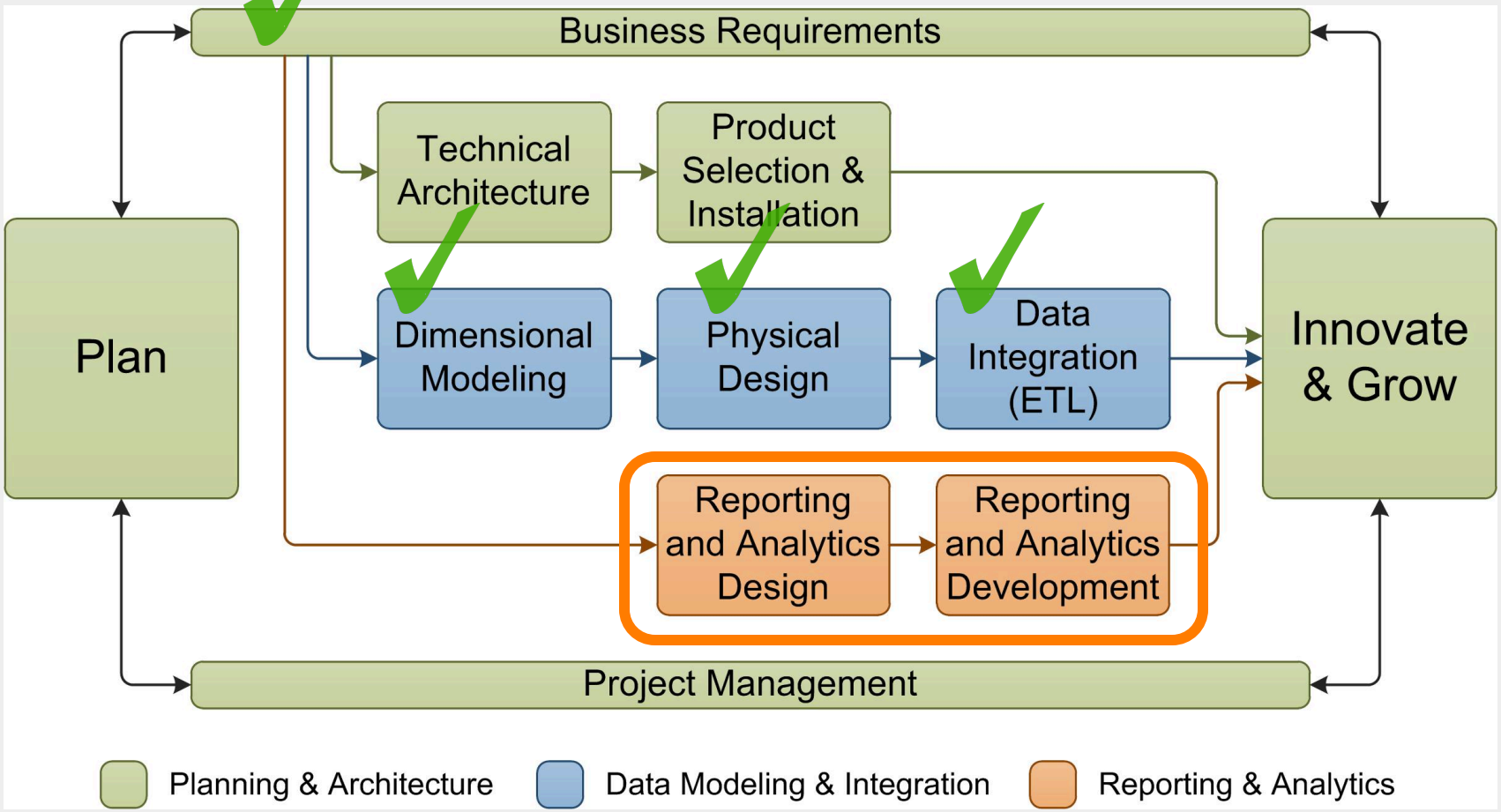
Partial Range Scan

```
SELECT cust_no, addr, un_pay, ....
  FROM (
        SELECT cust_no, addr,
               unpay_sum(cust_no)
               AS un_pay,
                ....
          FROM cust
         WHERE cust_stat IN
               ('A','C','F')
       )
 WHERE un_pay BETWEEN :VAL1 AND :VAL2
```

Week 11 Class Exercises

- BI Applications Demo
  - Metadata Editor
  - Metadata Workbench
  - Saiku Analytics (Mondrian OLAP)
  - Pentaho BI Suite
  - Tableau (http://www.tableau.com/academic/students)
- The Last Home Work (Group Home work)
  - Saiku Analytics Installation
- Assignment 02 Preview
  - Proof Of Concept Demo
  - Other Details

Kimball Lifecycle Approach Revisited

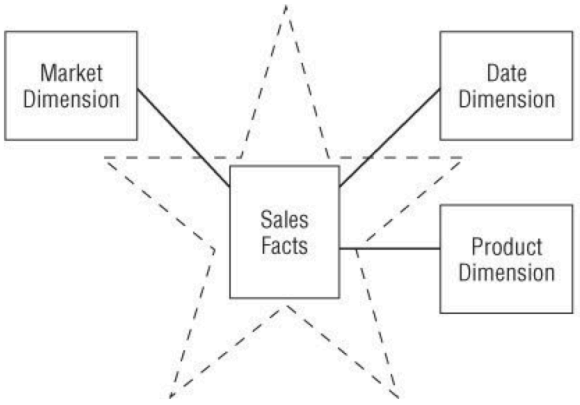


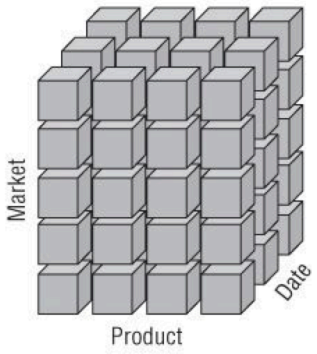
DW/BI Deliverables Revisited

- Dimensional models
  - Relational star schemas
  - Multidimensional online analytical processing (OLAP) cubes
- Business intelligence applications

Star Schema Versus OLAP Cubes

- At a logical level, there is no difference
- It is a matter of physical database implementation.
- Star schema is implemented in a relational database and is queried through SQL
- OLAP Cubes (multidimensional databases) are implemented for extreme performance and are queried through MDX.





Star Schema vs. OLAP Cubes Revisited

- The star schema can store large amounts of detailed data.
- OLAP Cubes provide higher performance with pre-calculated summary data.
- In general, OLAP cubes are populated from the star schema

- Kimball focuses on the star schema rather than the OLAP cubes
- Star schema usually has 15 dimensions
- OLAP usually has 8-10 dimensions

Assignment 02 Specification (1/3)

1. Dimensional Model
   - **Dimensions**
     - Minimum 1 Role Playing Dimension
     - 1 Junk Dimension (DIM_ETHNICITY)
     - Minimum 1 Bridge Table
     - Suggested Total Number of Dimensions: 5 ~ 8
   - **Measures**
     - GRE percentiles (Quant., Analytics, and Verbal)
     - Numeric flags on admission status

PROG_ACTN	IS_APPLIED	IS_ADMITTED	IS_ACCEPTED
APPL	1	0	0
DENY	1	0	0
WAPP	1	1	0
MATR	1	1	1

Assignment 02 Specification (2/3)

2. ETL Implementation
   - **Dimensions**
     - Implement minimum 5 dimensions
     - No bridge table implementation necessary
   - **Measures**
     - GRE percentiles (Quant., Analytics, and Verbal)
     - Numeric flags on admission status
3. Reporting and Analytics
   - Minimum 5 findings on the following measures using BI application(s)
     - Selectivity and Yield Ratio
     - GRE percentiles
   - BI application options
     - Saiku Analytics, Tableau, SQL Drilling down, etc.

Assignment 02 Specification (3/3)

4. Project Deliverables
   - Dimensional Model (in any format)
   - Fully functional ETL codes
   - Documentations
     - Setup instructions & DDL Scripts
     - ETL execution instructions
     - Minimum 5 findings from the Reporting & Analytics
5. Extra Credit
   - Group presentation on April 27
   - Must be notified by April 20th (FCFS)
   - The credit can be applied to all but the Final