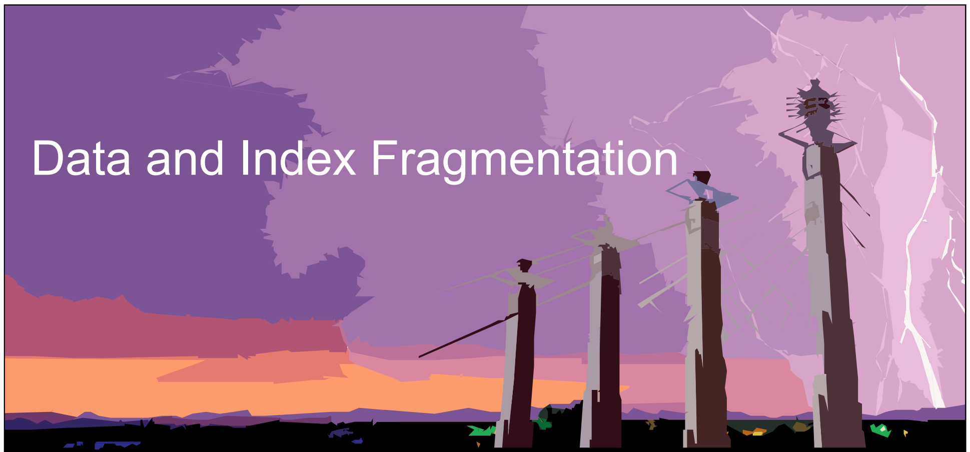
Data and Index Fragmentation



Bingjie Miao
IBM

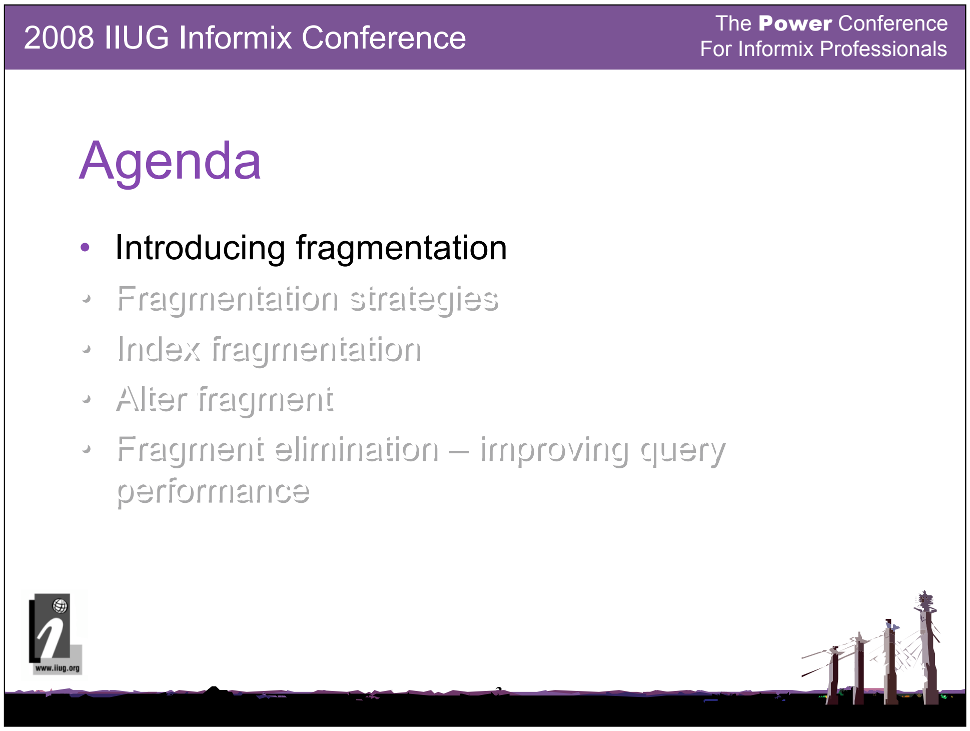
Change Session Code
Day, April 0, 2008 • 00:00 a.m. – 00:00 a.m.

2008 IIUG Informix Conference

www.iiug.org

Agenda

- Introducing fragmentation
- Fragmentation strategies
- Index fragmentation
- Alter fragment
- Fragment elimination – improving query performance



Agenda

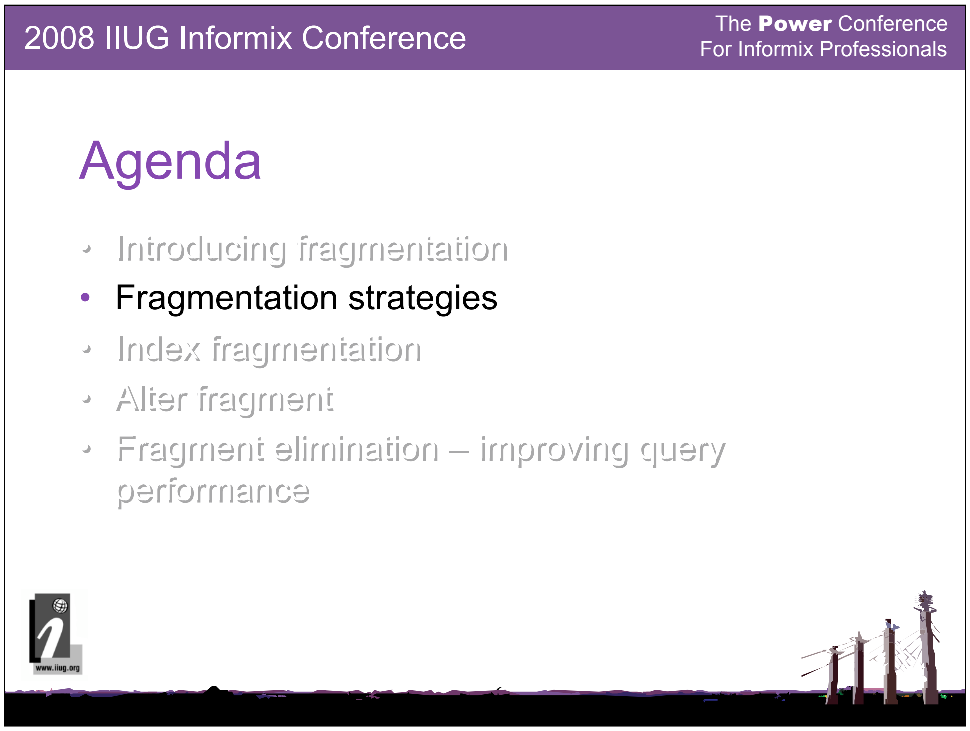
- Introducing fragmentation
- Fragmentation strategies
- Index fragmentation
- Alter fragment
- Fragment elimination – improving query performance

What is fragmentation

- Fragmentation is a database server feature that allows you to control where data is stored at the table level.
- Tables and indexes can be fragmented across different dbspaces or named partitions within a dbspace
- Transparent to user and client applications

Why fragmentation

- Larger table
- Concurrency
- Performance (fragment elimination, PDQ)
- Ease of administration (attach / detach)
- Availability

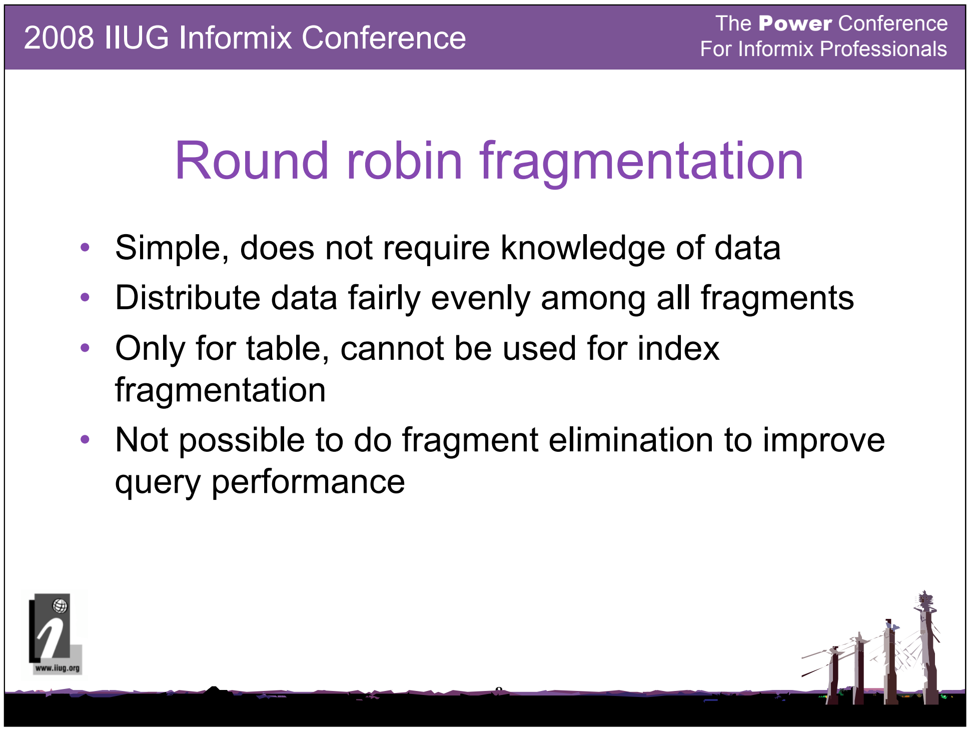


Agenda

- Introducing fragmentation
- Fragmentation strategies
- Index fragmentation
- Alter fragment
- Fragment elimination – improving query performance

Fragmentation strategies

- Round robin
- Expression



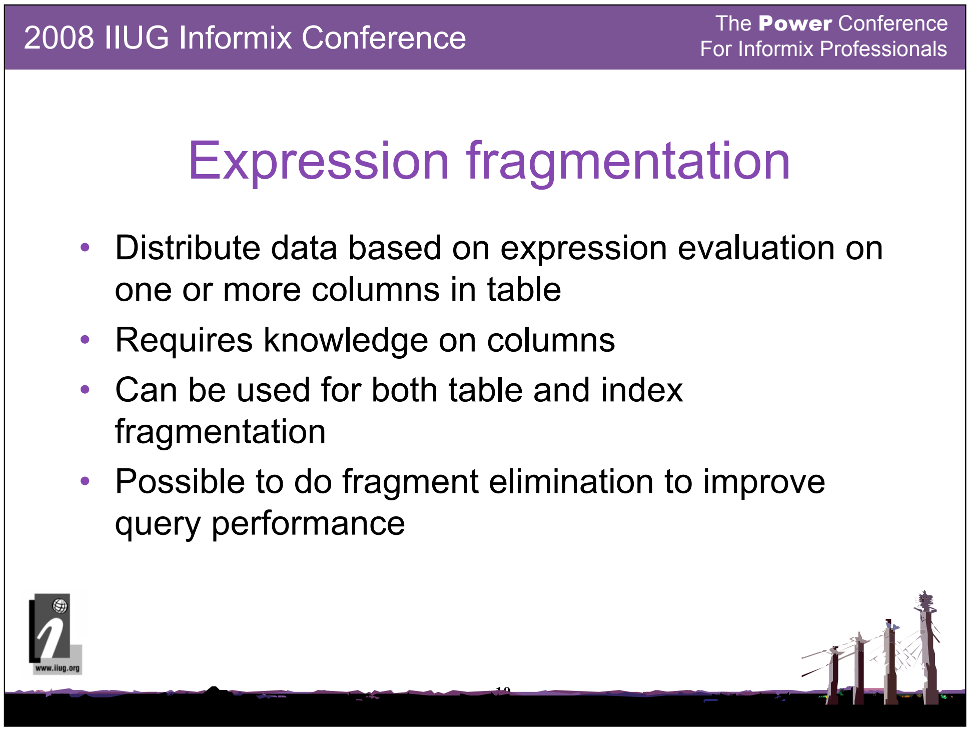
Round robin fragmentation

- Simple, does not require knowledge of data
- Distribute data fairly evenly among all fragments
- Only for table, cannot be used for index fragmentation
- Not possible to do fragment elimination to improve query performance

www.iiug.org

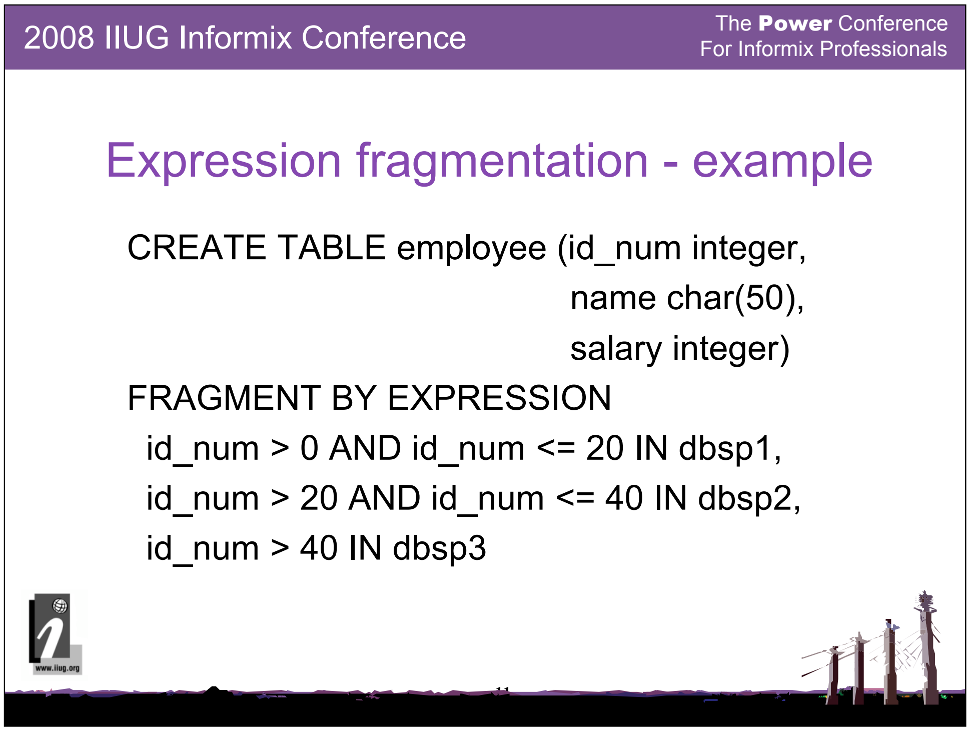
Round robin fragmentation - example

```
CREATE TABLE tab1 (col1 integer, col2 date)
FRAGMENT BY round robin in dbsp1, dbsp2
```



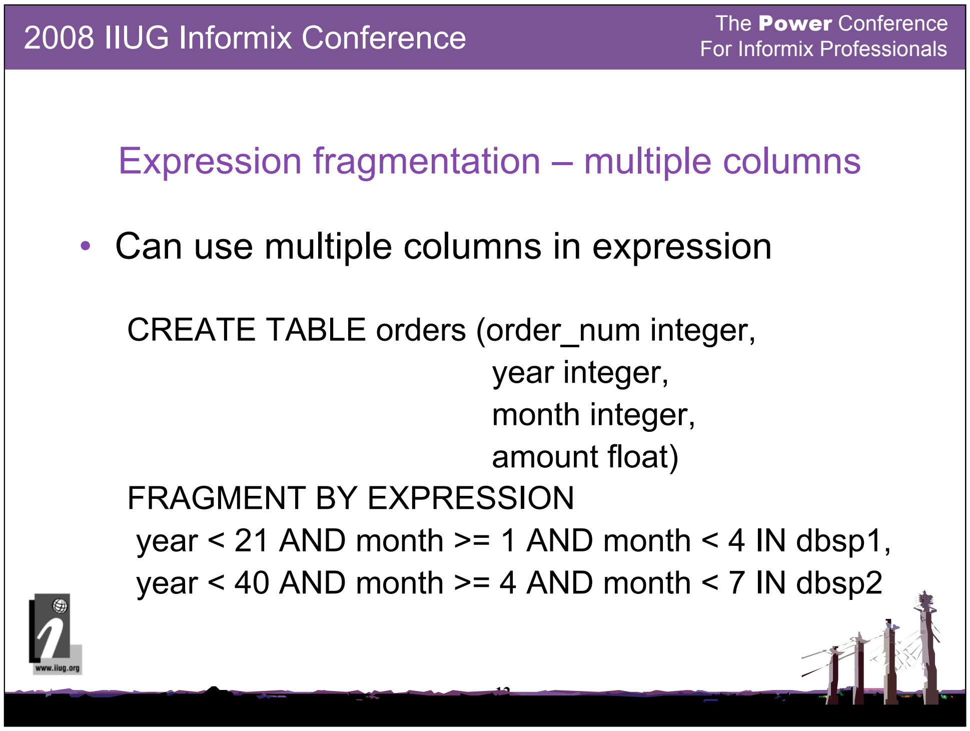
Expression fragmentation

- Distribute data based on expression evaluation on one or more columns in table
- Requires knowledge on columns
- Can be used for both table and index fragmentation
- Possible to do fragment elimination to improve query performance



Expression fragmentation - example

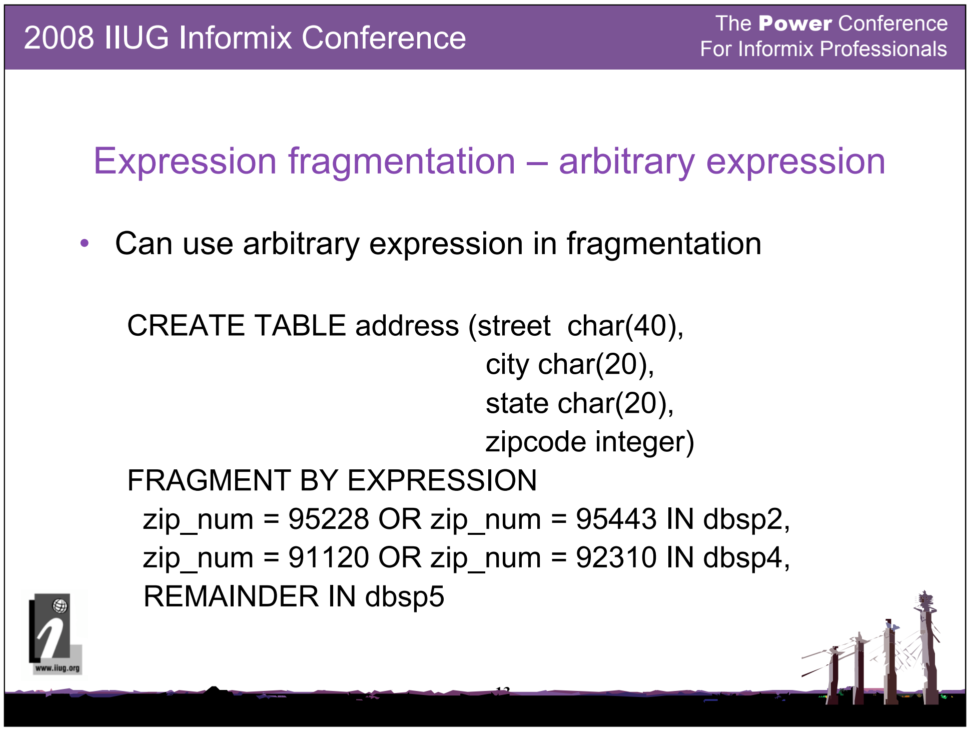
```
CREATE TABLE employee (id_num integer,
                       name char(50),
                       salary integer)
FRAGMENT BY EXPRESSION
  id_num > 0 AND id_num <= 20 IN dbsp1,
  id_num > 20 AND id_num <= 40 IN dbsp2,
  id_num > 40 IN dbsp3
```



Expression fragmentation – multiple columns

- Can use multiple columns in expression

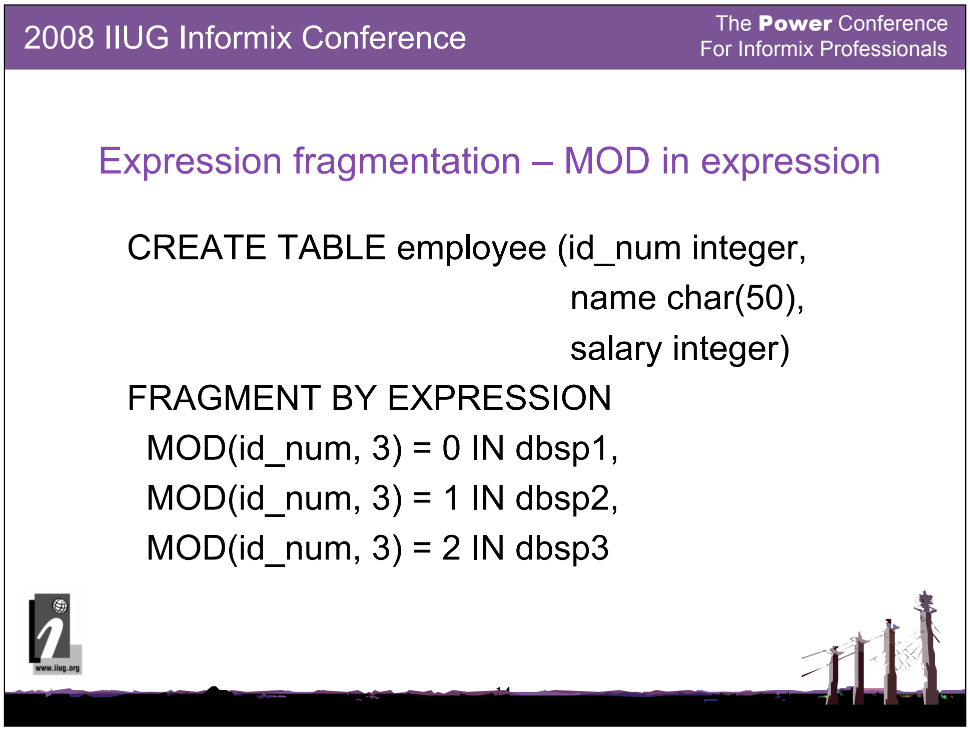
```
CREATE TABLE orders (order_num integer,
                     year integer,
                     month integer,
                     amount float)
FRAGMENT BY EXPRESSION
 year < 21 AND month >= 1 AND month < 4 IN dbsp1,
 year < 40 AND month >= 4 AND month < 7 IN dbsp2
```



Expression fragmentation – arbitrary expression

- Can use arbitrary expression in fragmentation

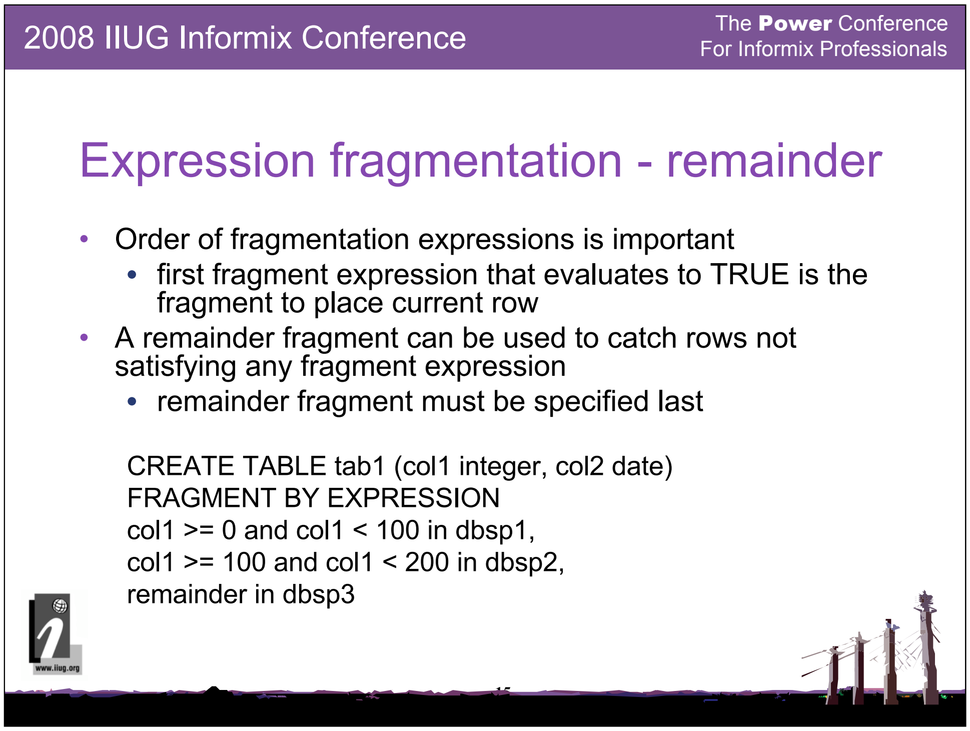
```
CREATE TABLE address (street  char(40),
                               city char(20),
                               state char(20),
                               zipcode integer)
FRAGMENT BY EXPRESSION
  zip_num = 95228 OR zip_num = 95443 IN dbsp2,
  zip_num = 91120 OR zip_num = 92310 IN dbsp4,
  REMAINDER IN dbsp5
```



Expression fragmentation – MOD in expression

```
CREATE TABLE employee (id_num integer,
                       name char(50),
                       salary integer)
FRAGMENT BY EXPRESSION
  MOD(id_num, 3) = 0 IN dbsp1,
  MOD(id_num, 3) = 1 IN dbsp2,
  MOD(id_num, 3) = 2 IN dbsp3
```

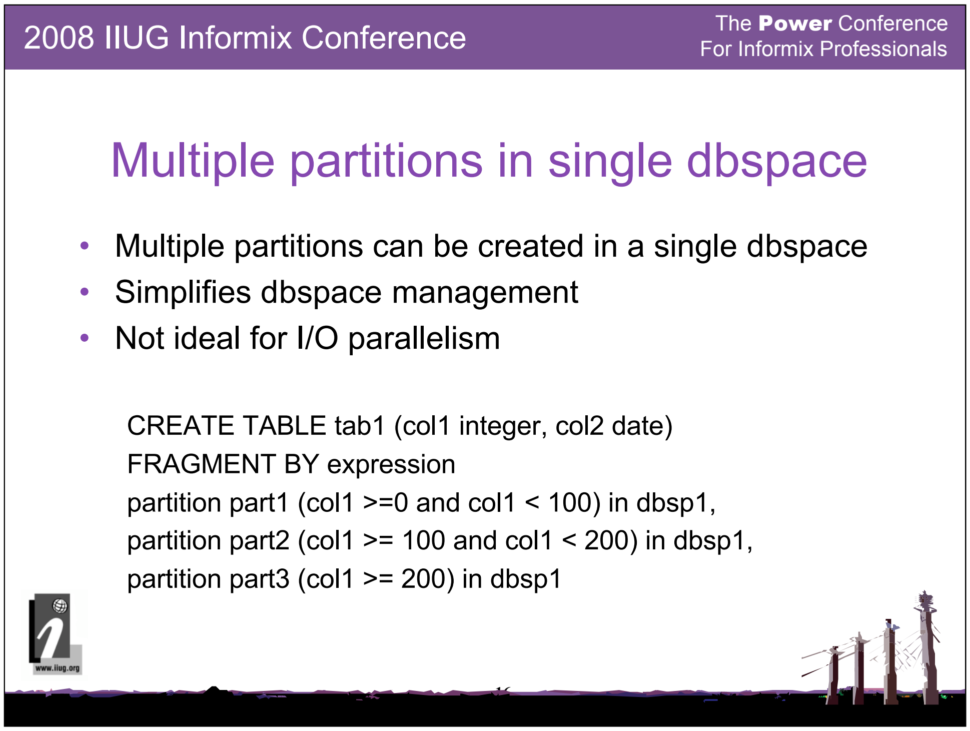
www.iiug.org



Expression fragmentation - remainder

- Order of fragmentation expressions is important
  - first fragment expression that evaluates to TRUE is the fragment to place current row
- A remainder fragment can be used to catch rows not satisfying any fragment expression
  - remainder fragment must be specified last

```
CREATE TABLE tab1 (col1 integer, col2 date)
FRAGMENT BY EXPRESSION
col1 >= 0 and col1 < 100 in dbsp1,
col1 >= 100 and col1 < 200 in dbsp2,
remainder in dbsp3
```



Multiple partitions in single dbspace

- Multiple partitions can be created in a single dbspace
- Simplifies dbspace management
- Not ideal for I/O parallelism

```
CREATE TABLE tab1 (col1 integer, col2 date)
FRAGMENT BY expression
partition part1 (col1 >=0 and col1 < 100) in dbsp1,
partition part2 (col1 >= 100 and col1 < 200) in dbsp1,
partition part3 (col1 >= 200) in dbsp1
```

Agenda

- Introducing fragmentation
- Fragmentation strategies
- Index fragmentation
- Alter fragment
- Fragment elimination – improving query performance

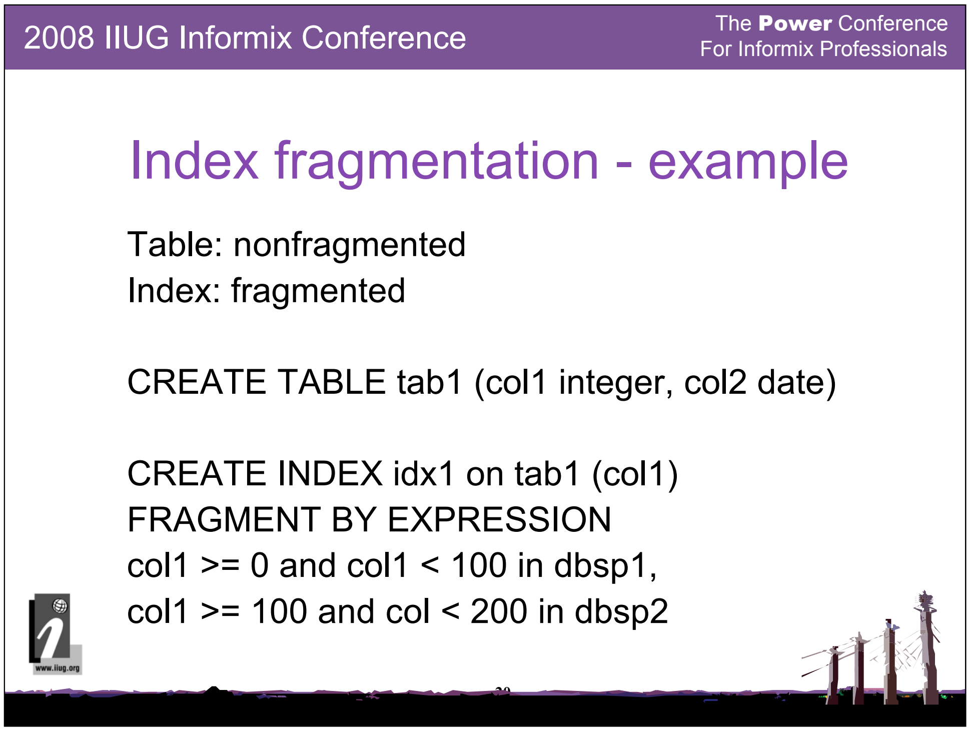
Index fragmentation

- Index can be fragmented
- Index can have different fragmentation strategy than table
- Index fragments can have different page size than table fragments (page size needs to be the same across all fragments of the index)
- Can be explicitly fragmented using expression fragmentation, or inherit table's fragmentation, or non-fragmented

Index fragmentation - example

- If no FRAGMENT BY clause is specified on index creation, index inherits table’s fragmentation strategy

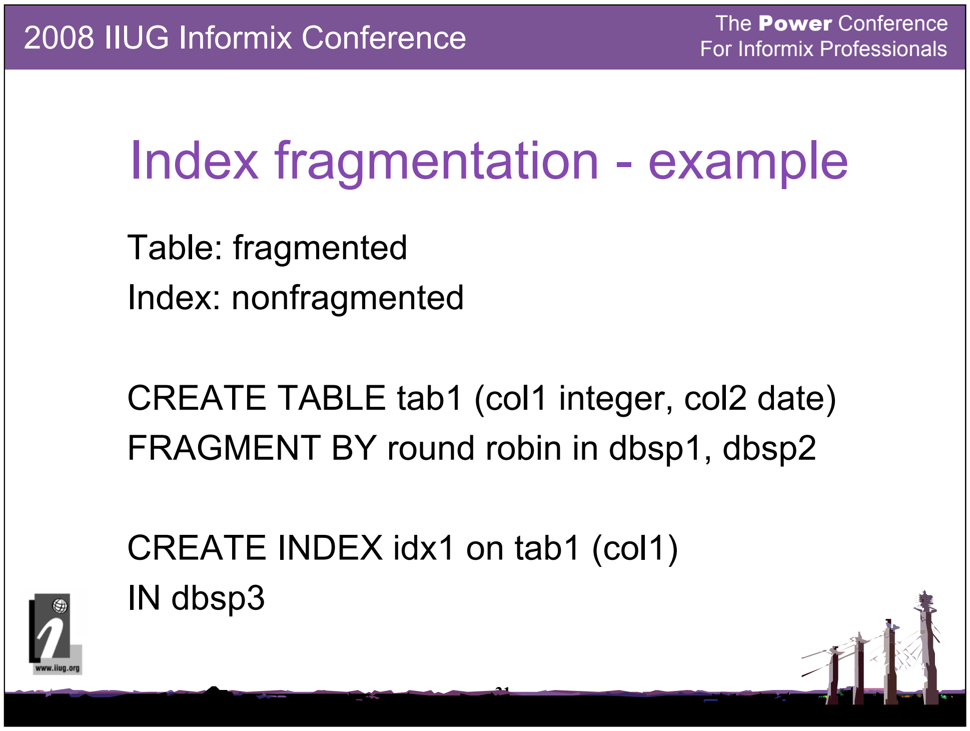
```
CREATE TABLE tab1 (col1 integer, col2 date)
FRAGMENT BY expression
col1 >= 0 and col1 < 100 in dbsp1,
col1 >= 100 and col1 < 200 in dbsp2

CREATE INDEX idx1 on tab1 (col1)
CREATE INDEX idx2 on tab1 (col2)
```



Index fragmentation - example

Table: nonfragmented
Index: fragmented

```
CREATE TABLE tab1 (col1 integer, col2 date)

CREATE INDEX idx1 on tab1 (col1)
FRAGMENT BY EXPRESSION
col1 >= 0 and col1 < 100 in dbsp1,
col1 >= 100 and col < 200 in dbsp2
```



Index fragmentation - example

Table: fragmented

Index: nonfragmented

```
CREATE TABLE tab1 (col1 integer, col2 date)
FRAGMENT BY round robin in dbsp1, dbsp2

CREATE INDEX idx1 on tab1 (col1)
IN dbsp3
```

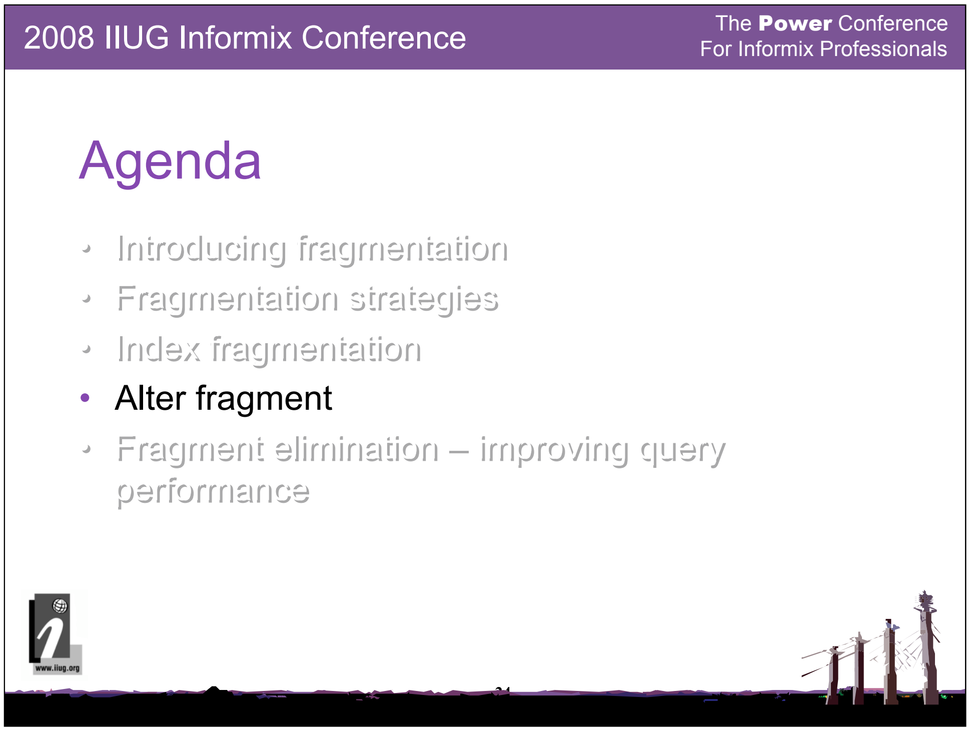
Index fragmentation - example

Table: fragmented
Index: fragmented (different than table)

```
CREATE TABLE tab1 (col1 integer, col2 date)
FRAGMENT BY round robin in dbsp1, dbsp2

CREATE INDEX idx1 on tab1 (col1)
FRAGMENT BY EXPRESSION
col1 >= 0 and col1 < 100 in dbsp1,
col1 >= 100 and col < 200 in dbsp2
```

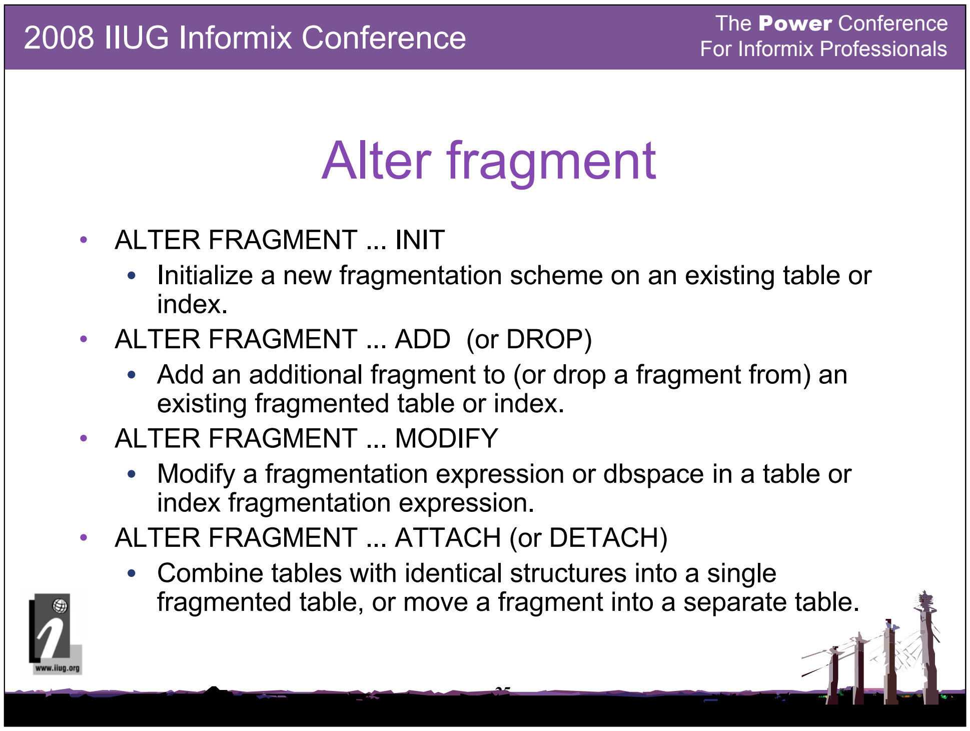
Index fragmentation – misc.

- If a unique index is fragmented, the unique column(s) must be in the fragmentation columns
- System generated indexes (for constraints) are never fragmented (generated in dbspace where database is created)



Agenda

- Introducing fragmentation
- Fragmentation strategies
- Index fragmentation
- Alter fragment
- Fragment elimination – improving query performance



Alter fragment

- ALTER FRAGMENT ... INIT
  - Initialize a new fragmentation scheme on an existing table or index.
- ALTER FRAGMENT ... ADD (or DROP)
  - Add an additional fragment to (or drop a fragment from) an existing fragmented table or index.
- ALTER FRAGMENT ... MODIFY
  - Modify a fragmentation expression or dbspace in a table or index fragmentation expression.
- ALTER FRAGMENT ... ATTACH (or DETACH)
  - Combine tables with identical structures into a single fragmented table, or move a fragment into a separate table.

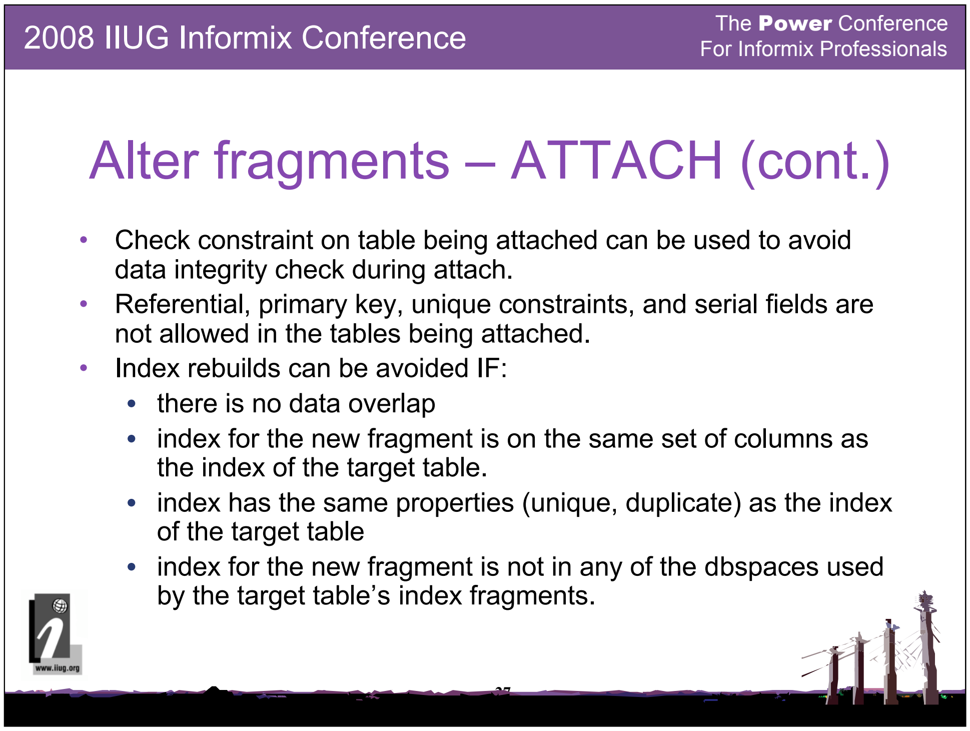
Alter fragments - ATTACH

- Combine two non-fragmented tables with identical schemas into one fragmented table.
  - Two tables can be in different dbspaces or in the same dbspace

  Alter fragment on table mytab1 ATTACH mytab1 as partition p1, mytab2 as partition p2
- Combine a non-fragmented table into part of a fragmented table.

  Alter fragment on table mytab1 attach mytab3 as partition p3
- Can use BEFORE and AFTER clause if using fragmentation by expression.

  Alter fragment on table f1 attach f3 as (col1 <= 0) before dbspace1



Alter fragments – ATTACH (cont.)

- Check constraint on table being attached can be used to avoid data integrity check during attach.
- Referential, primary key, unique constraints, and serial fields are not allowed in the tables being attached.
- Index rebuilds can be avoided IF:
  - there is no data overlap
  - index for the new fragment is on the same set of columns as the index of the target table.
  - index has the same properties (unique, duplicate) as the index of the target table
  - index for the new fragment is not in any of the dbspaces used by the target table's index fragments.

Alter fragment - DETACH

- Separate a fragmented table into multiple tables.

```
Alter fragment on table f1 DETACH dbspace2 f2
Alter fragment on table mytab1 DETACH partition p2 mytab2
```

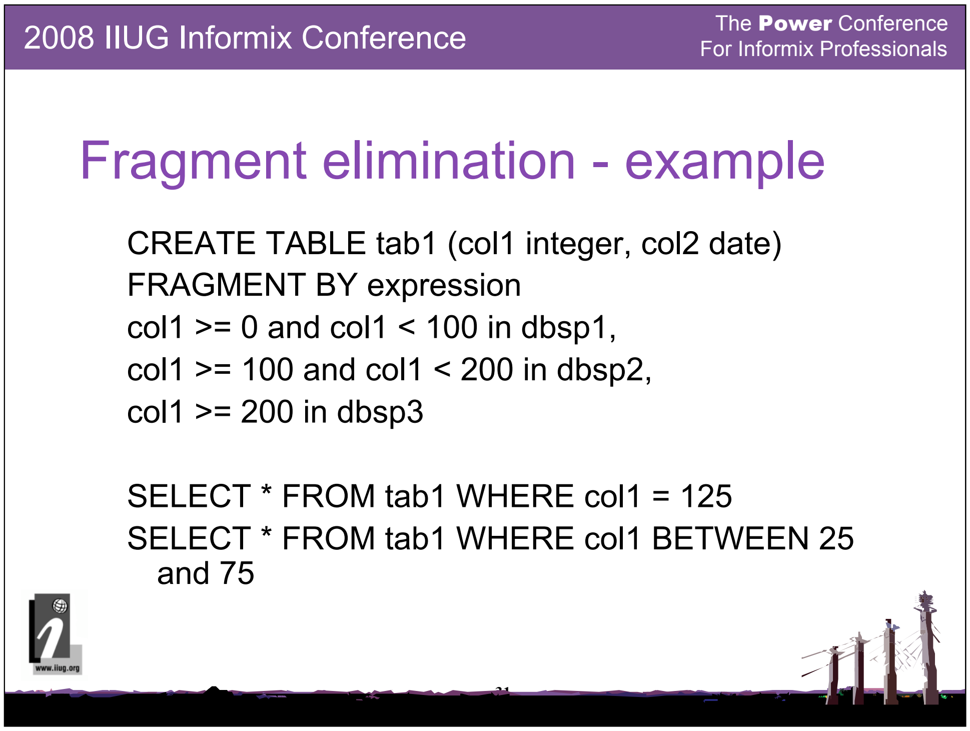
- Index rebuilds may be necessary if the index fragmentation strategy is not similar to the table fragmentation. (If similar, the index fragments corresponding to the detached fragment will be dropped.)

Agenda

- Introducing fragmentation
- Fragmentation strategies
- Index fragmentation
- Alter fragment
- Fragment elimination – improving query performance

Fragment elimination

- Eliminate fragments of a table/index from query processing if no relevant data result in those fragments (based on query predicates)
- Works on expression fragmentation
    - “simple”, non-overlapping fragments
- Works on both table and index

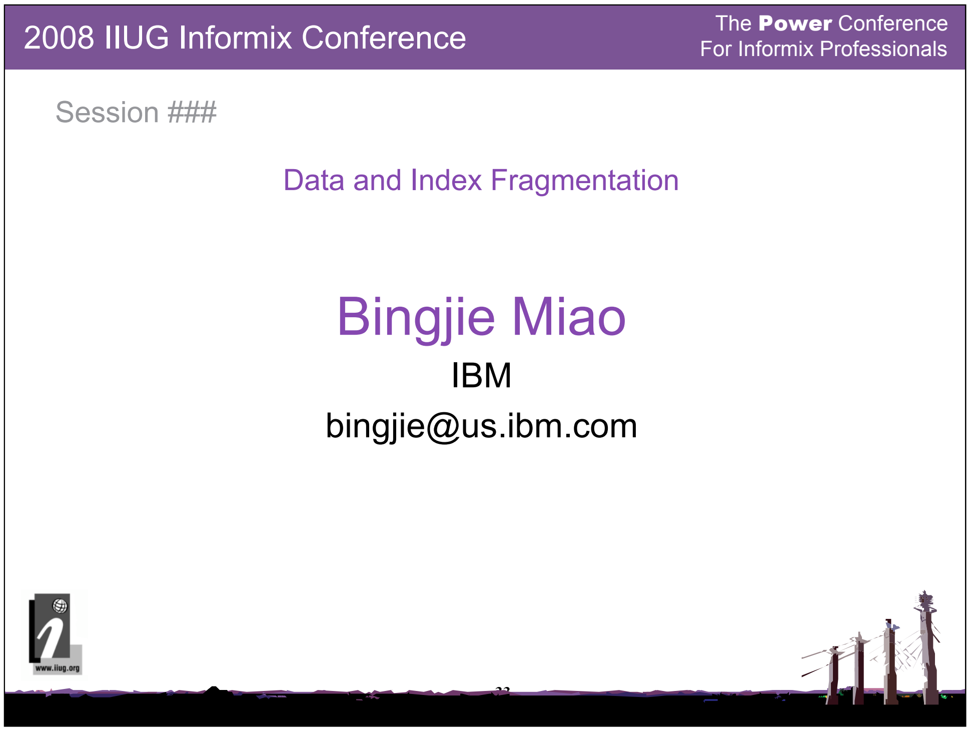


Fragment elimination - example

```
CREATE TABLE tab1 (col1 integer, col2 date)
FRAGMENT BY expression
col1 >= 0 and col1 < 100 in dbsp1,
col1 >= 100 and col1 < 200 in dbsp2,
col1 >= 200 in dbsp3

SELECT * FROM tab1 WHERE col1 = 125
SELECT * FROM tab1 WHERE col1 BETWEEN 25
   and 75
```

Fragment elimination – explain file

```
SELECT * FROM tab1 WHERE col1 = 125

 1) usr1.tab1: SEQUENTIAL SCAN  (Serial, fragments: 1)

        Filters: usr1.tab1.col1 = 125

SELECT * FROM tab1 WHERE col1 BETWEEN 25 and 75

 1) usr1.tab1: SEQUENTIAL SCAN  (Serial, fragments: 0)

        Filters: (usr1.tab1.col1 <= 75 AND usr1.tab1.col1 >= 25 )
```



Session ###

Data and Index Fragmentation

Bingjie Miao

IBM

bingjie@us.ibm.com