

Storage Areas: Fact and Fiction



Scott M. Dulecki
Product Manager

BRAVEPOINT

Agenda



- **Setting the Stage**
- The big bucket
- Many small buckets
- Enter the sandwich bags
- Questions?
- Take-Homes

Setting the Stage: Me

- Scott M. Dulecki
- Presenter at Explore! and regional user groups
- Board Member, Midwest MFG/PRO Users Group
- President, West Michigan Progress Users Group
- Vice-President, Michigan Progress Users Group
- PEG member 1998061901
- Author of:
 - *Safe Haven: Archiving in MFG/PRO*

Setting the Stage: Us



- BravePoint Inc. (www.BravePoint.com)
- 110+ Employees (30+ MFG/PRO Consultants)
- Three of us have used Progress since 1984
- QAD Alliance Partner
- Progress Service Provider
- Helping Progress users succeed since 1987

Setting the Stage: You



- Progress Version?
- Live since?
- Database size?
- Using Storage Areas now?
- Using them effectively?

Agenda



- Setting the Stage
- **The big bucket**
- Many small buckets
- Enter the sandwich bags
- Questions?
- Take-Homes

In the beginning...



- There was the database, and it was good.
- It acted like a bucket filled with data
- All data and indexes were contained in the bucket, and life was good.
- Until the bucket filled up...
 - Who would ever need more than 2GB of data?

Then came multiple volumes...

- The bucket became a giant bucket, made of smaller buckets.
- Logically, they made up one bucket.
- They were managed as one bucket
- Life was good... until people saw that not all the sand was equal.

Then came the sneaky ones...



- Multiple databases
 - Help, production, others
- Remove some of the data from the database
- Put it elsewhere... and connect to it

There were problems...



- The accursed `-g` and `-a` parameters
- The evil split schema
- All data was still not equal... and there was no easy way to split it out any more
- There were hidden limits...
 - Like 2GB BI size

But there were tools



- Multi-volume structures, to link the AI and BI files
- Promon, to examine all database statistics
- Progress utilities, to manage the database as a unit

But what if....?



- You could separate data, but still manage it easily?
- You could separate indexes, and still manage them?
- You could put certain tables by themselves?

Agenda



- Setting the Stage
- The big bucket
- **Many small buckets**
- Enter the sandwich bags
- Questions?
- Take-Homes

Then came Progress 9



- Enter the storage area!
- Break the big bucket into smaller buckets
- Each could have a subset of data or indexes
- And only that subset!
- Static data can be set once, and left alone
- Dynamic data can be allowed to grow...
safely

But you could do glorious things!



- Set records per block
 - Based on data size... from 1 to 256
 - Impact the size of each area
- Dump and load by table
 - No value at all in olden days
- Eliminate split schema databases!
 - Put them all into one, and split there

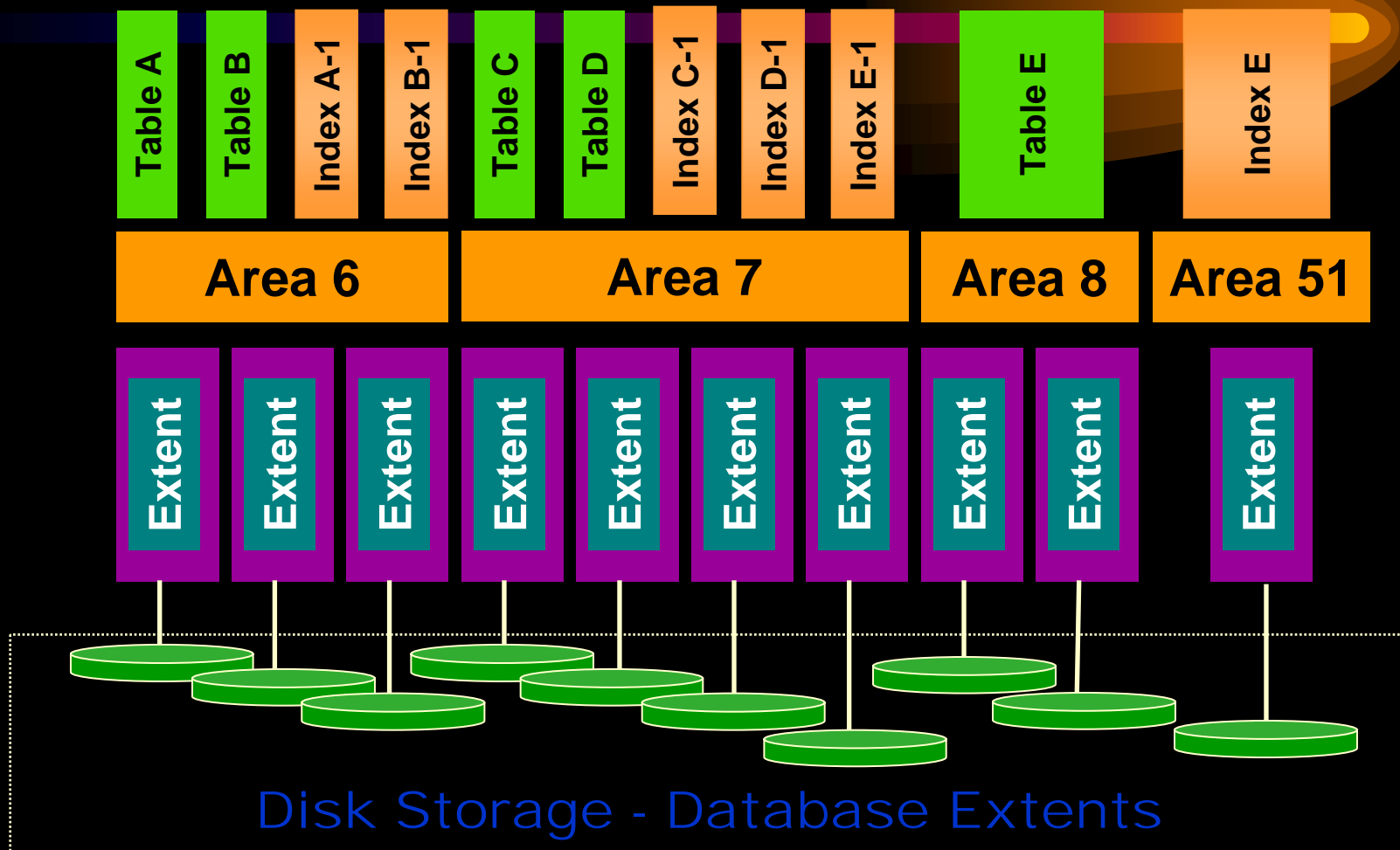
But all was not perfect...

- Each storage area has its own high water mark
 - Promon HWM is now useless
- No more single volume DBs
 - But that's good!
- Database will probably be larger
 - Extra space for every area, not just the DB
- Can't add new extents (or areas) on-line
 - Until 10.0

So what does it look like?

- Control Area (.db File)
 - In V8 (and earlier) contained hardcoded Extent path names
 - In V9 contains `_Area`, `_AreaExtent` Tables; these tables can be queried with the 4GL
- Primary Recovery Area (V8 BI Extents)
 - 2GB BI Size Limit is Gone! (it's just much higher)
- After-Image Areas (V8 AI Extents)
- User Defined Data Areas (no V8 equivalent)
- Schema Area (No V8 equivalent)

No, really... here's what it looks like



What should I know about it?

- The schema area (6) is the default repository
 - That's where everything goes with a conv89
- Layout is described in structure file
- One variable extent per area
- Some utilities are area-aware
 - Prostrct, prorest, procopy
- Proutil mvsch... reset schema

Storage Areas (.df File)

```
ADD TABLE "Order"  
  AREA "od1"  
  DESCRIPTION "Order header Info"  
  VALEXP "1 = 1"  
  DUMP-NAME "order"
```

```
ADD INDEX "Order-Num" ON "Order"  
  AREA "odi1"  
  UNIQUE  
  PRIMARY  
  INDEX-FIELD "Order-num" ASCENDING
```

V9 Structure (.st) Files

```
# Schema Area
# Records/Block:64
# Located in the /db directory
d "schema area",64          /db

# Data Area
# Area Name: data1 Area Number: 10
# Records/Block:128
d "data1":10,128 /db1/data_10.d1  f          100000
d "data1":10,128 /db2/data_10.d2  f          100000
d "data1":10,128 /db3/data_10.d3

# Index Area
#Area Name:index1Area Number: 20
#Records/Block:Not given because intended for index blocks
d "index1":20      /db4/          f          100000
d "index1":20      /db4/

# Variable size BI extent
b /bi/prod.b1

# After Image Extents
a          /ai/data.a1          f          100000
a          /ai/data.a2          f          100000
a          /ai/data.a3          f          100000
```

b .

d "Schema Area":6,64 .

d "TRANSACTION":7,64 . f 128

d "TRANSACTION":7,64 .

d "TRANSACTION_IDX":8,32 . f 128

d "TRANSACTION_IDX":8,32 .

d "STATIC":9,64 . f 128

d "STATIC":9,64 .

d "STATIC_IDX":10,32 . f 128

d "STATIC_IDX":10,32 .

d "HISTORY":11,64 . f 128

d "HISTORY":11,64 .

d "HISTORY_IDX":12,32 . f 128

d "HISTORY_IDX":12,32 .

d "TRHIST":25,64 . f 128

d "TRHIST":25,64 .

d "TRHIST_IDX":26,32 . f 128

d "TRHIST_IDX":26,32 .

Things to note...

- Areas can have multiple objects, but objects can't have multiple areas
- If you can... keep to 19 areas. Why?
- Each area can have its own RPB
 - ST-Gen in DBA Resource Kit can optimize*

Storage Area Layout



- Fools Errand
 - Put index Areas on separate disks from data Areas to distribute the I/O evenly... HA!
 - It's nearly impossible to figure out the average I/O activity of 30+ Areas for tables & indexes
- Recommendation
 - Put the Database Extents (all Data Areas) on a set of Striped Disks (RAID 0 or RAID 10)
 - Keep BI & AI extents on Mirrored Pairs (RAID 1)

Options to Convert to V9

- *proutil conv89*
 - Usually takes less than 5 minutes to run
 - All Data & Indexes are stored in the Schema Area
 - Will NOT take advantage of Storage Areas
 - Moving Tables and Indexes still leaves a Large Schema Area (V9.1D *proutil/mvsch*)
- Dump/Load
 - Will takelonger than conv89
 - Can take full advantage of Areas through easy modifications to the .df file

V9 Utilities - Table Move

- `proutil <db> -C tablemove <table> <area>`
 - Rebuilds the Indexes during the move since Recids are no longer unique to a DB (but are unique to an Area)
 - Off-line as well as on-line
 - On-line: Will Exclusive Lock the entire Table for the duration of the move
 - One Move = One Trx = Bigger BI/AI Files
- Monitor the process with the `_UserStatus VST`; the Table Lock can be monitored with `promon` or the `_Lock VST`

V9 Utilities - Index Move

- proutil <db> -C indexmove <Index def> <area>
 - Index def: <owner>.<table>.<index>
 - Run off-line or on-line
 - On-line:
 - Only read operations will be allowed
 - The table will be SHARE locked
- Benefits: flexibility and availability
- Monitor the process with the _UserStatus VST

V9 Utilities - Index Compact

- proutil <db> -C indexcompact
<owner>.<tablename>.<indexname>
 - Combines partially filled adjacent index blocks
 - Runs off-line or on-line
 - Index Fix + Index Compact = Index Rebuild
- Benefits:
 - Performance - fewer disk i/o's
 - Reduces index size
 - Availability - index not locked
- **DB Corrupting Bug until V9.1D SP06**

Enhancements to Utilities

- Most utilities are now ‘Area Aware’
- Example: *proutil idxbuild* will only scan the blocks in the Area where the table exists
- *proutil dbanalyz/tabanalyz/ixanalyz* can now report individual Areas (this is undocumented)
 - *proutil <db> -C dbanalyz <area name>*

Procopy / probkup

- *procopy & probkup*
 - Target Area Numbers must Match
 - Area Names don't need to Match
 - Target Records per Block must Match
 - *prorest* - the # of Areas must match
 - Uses .st File (if one exists) in Target Directory

Truncate Area



- *proutil truncate area*
 - If no Area is specified, all Areas that have no Storage Objects have the HWM moved to ‘ground zero’ (AKA empty)
 - If an Area name is specified, the data in the Area is removed (after confirmation)
 - In each case there is no impact on the physical extents

V9 Size Limits

- Database size: BIG! (mb > gb > tb > eb)
- Primary Recovery Area (BI) size: 16-32TB
- Maximum Number of Areas: 1,000
- Data Area size: Varies
 - 16TB (1 RPB, 8k block size)
 - 8GB (256 RPB, 1k block size)
 - 64GB (256 RPB, 8k block size)
- Data Area block size: 8,192 bytes
- Extent size: Still 2 gigabytes until V9.1C
 - proutil <db> -C EnableLargeFiles

V9 Size Limits

- Record size: 32,000 bytes
- Records per Area/Table: 2,147,483,647
 - Depends on DB Block Size & RPB Value
- Number of Records per block: 256
- Number of Indexes: 32,000
- Number Sequences: 2,000 (with 8k blocks)
 - Increased from 100
- Number of concurrent users: 10,000
- Number of concurrent transactions: 10,000

But with all that...



- Life still wasn't good.
- Wouldn't it be great if I could easily manage individual tables inside storage areas?

Agenda



- Setting the Stage
- The big bucket
- Many small buckets
- **Enter the sandwich bags**
- Questions?
- Take-Homes

Enter type II storage areas!

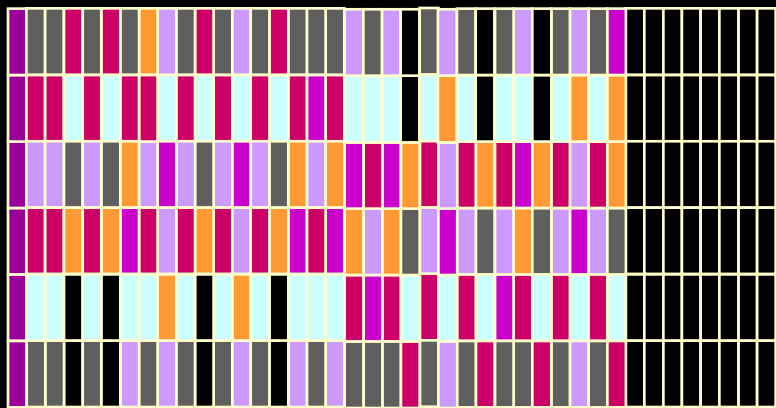
- Now we know the others were type I areas...
- Type II areas don't deal with DB blocks, but with clusters
 - 8, 64, or 512 blocks in cluster

OpenEdge 10.0A

Online Database tools, ASA II Storage

Type I Storage Area

Db extent (variable)

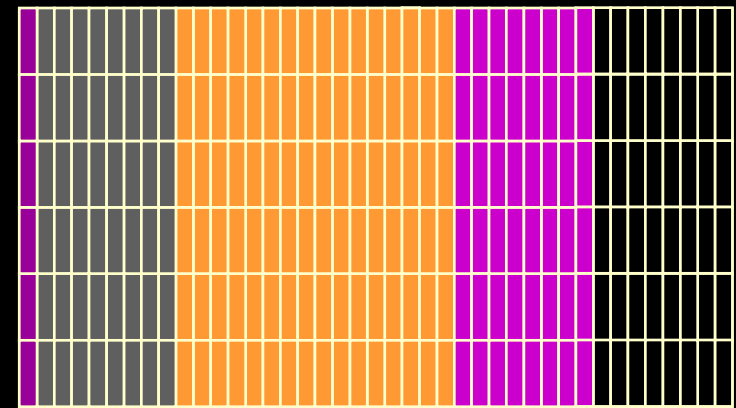


High
Water
Mark

Total
Blocks

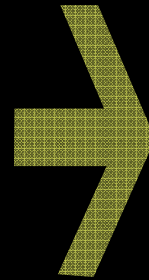
Type II Storage Area

Db extent (variable)

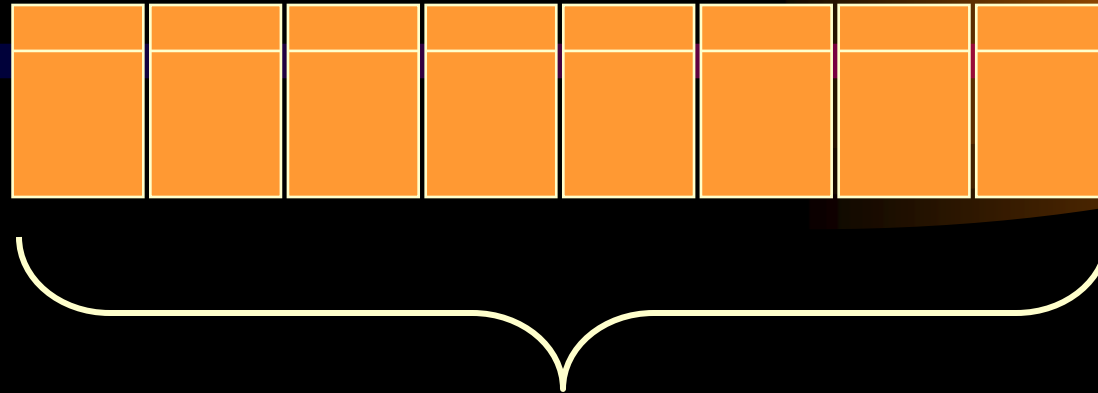


High
Water
Mark

Total
Blocks



Type II Area Block Clusters



Block Cluster:

- ❖ 8, 64, or 512 adjacent blocks
- ❖ Configured in .st file
- ❖ Applied via prostrct
- ❖ Fixed size for area
- ❖ Unit of space allocation for objects
- ❖ Blocks within cluster are “non-social”

Setting Cluster Size

```
b /bi/exampleDB.b1 f 1024000
```

```
b /bi/exampleDB.b2 f 1024000
```

```
b /bi/exampleDB.b3
```

```
#
```

```
d "Schema Area":6,64 /db/exampleDB.d1
```

```
#
```

```
d "Customer Indexes":7,1;8 /db/exampleDB_7.d1 f 512000
```

```
d "Customer Indexes":7,1;8 /db/exampleDB_7.d2
```

```
#
```

```
d "Customer Data":8,128;64 /db/exampleDB_8.d1 f 1024000
```

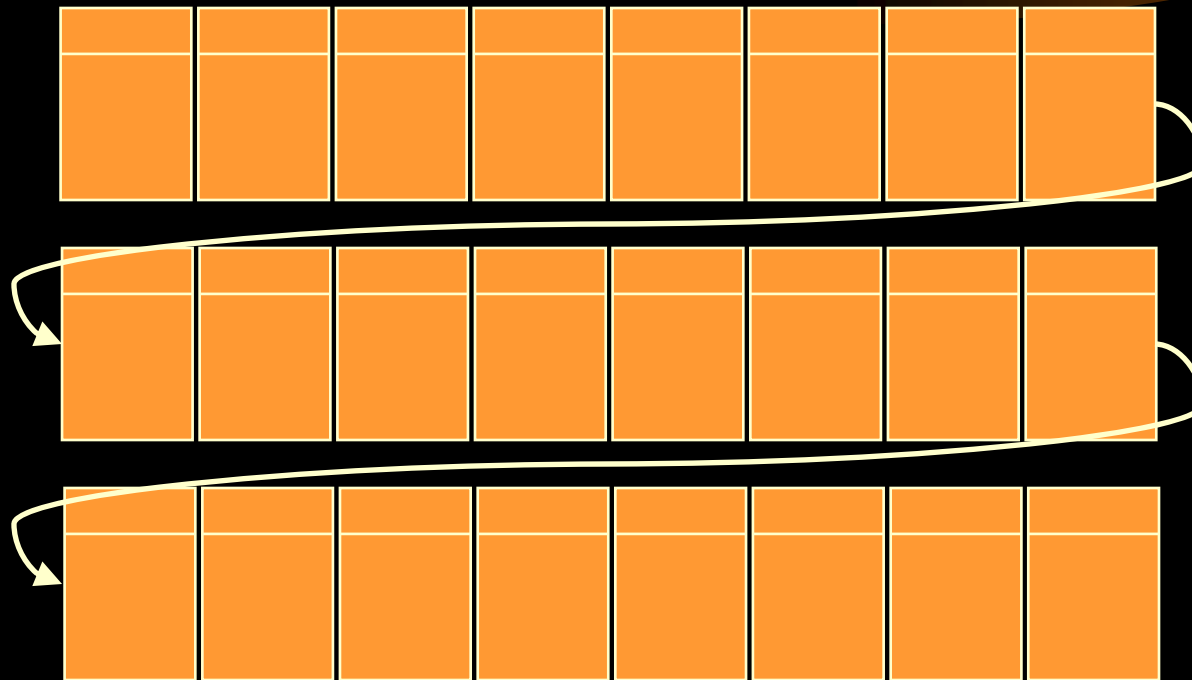
```
d "Customer Data":8,128;64 /db/exampleDB_8.d2
```

Type II Area Objects

Clusters chained together for fast access

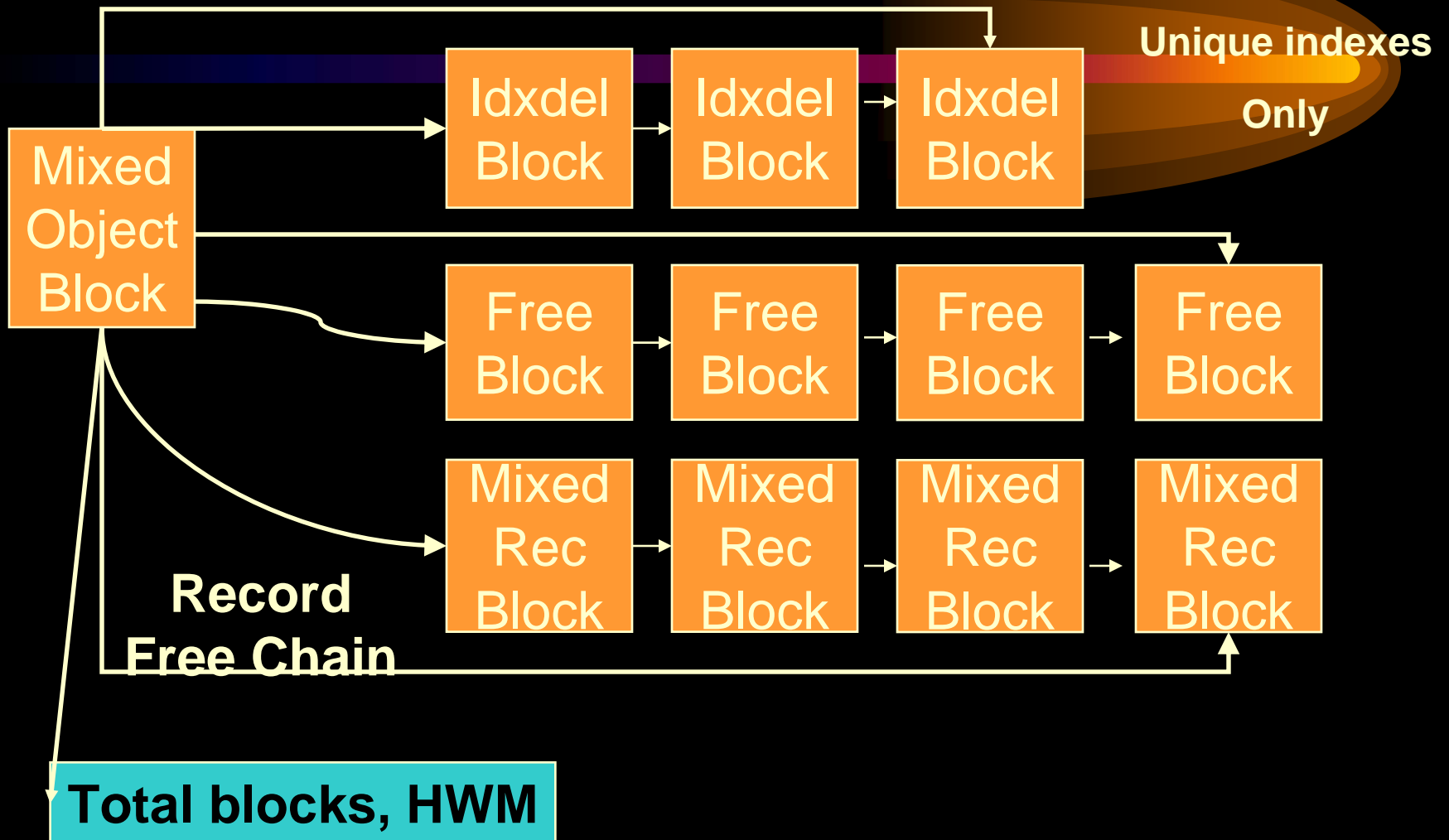
Allows table scan without an index

Allows fast table delete



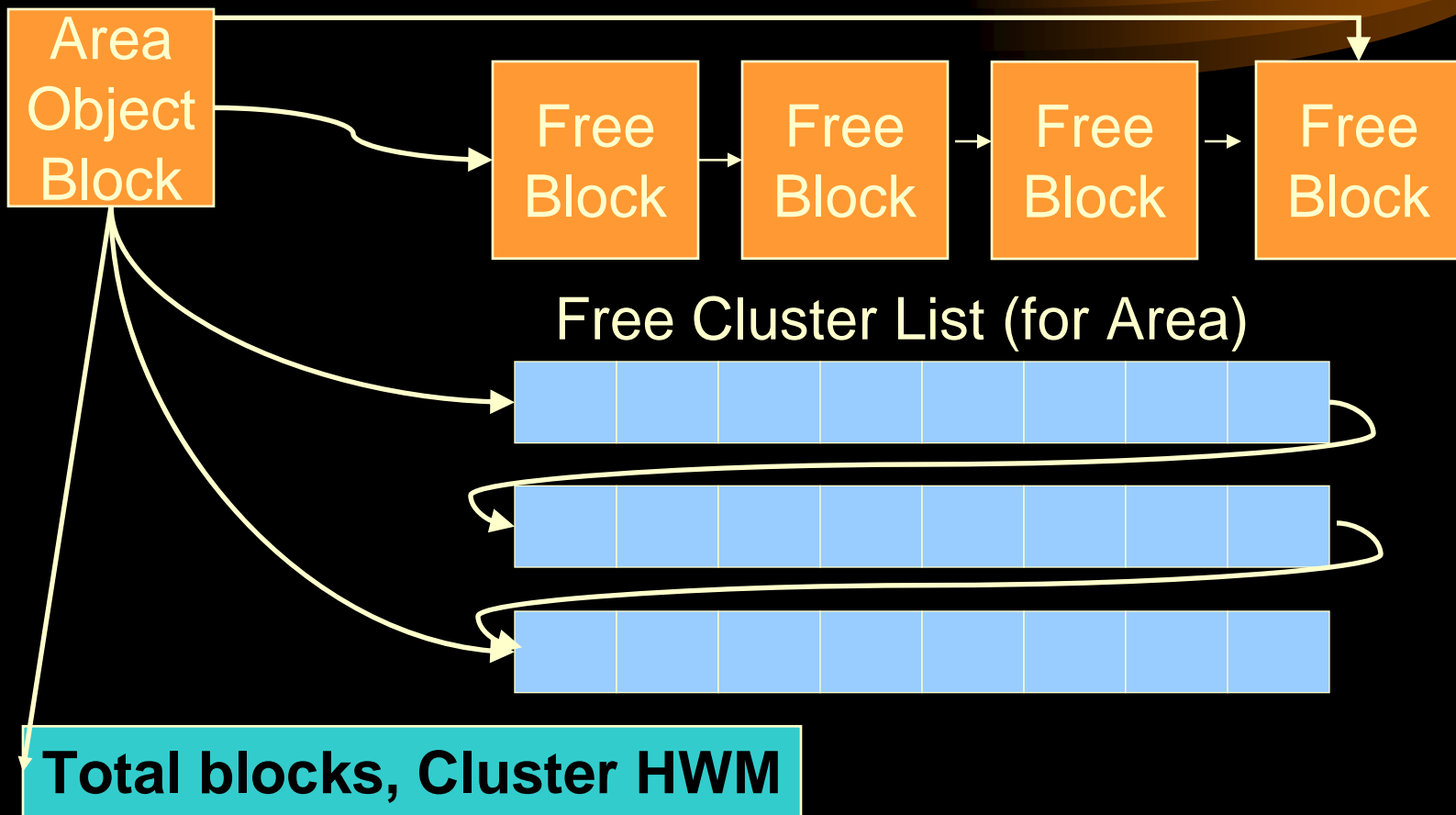
(Objects: tables, indexes, LOBS, area control)

Object Block (Type I Area): Space Allocation Chains

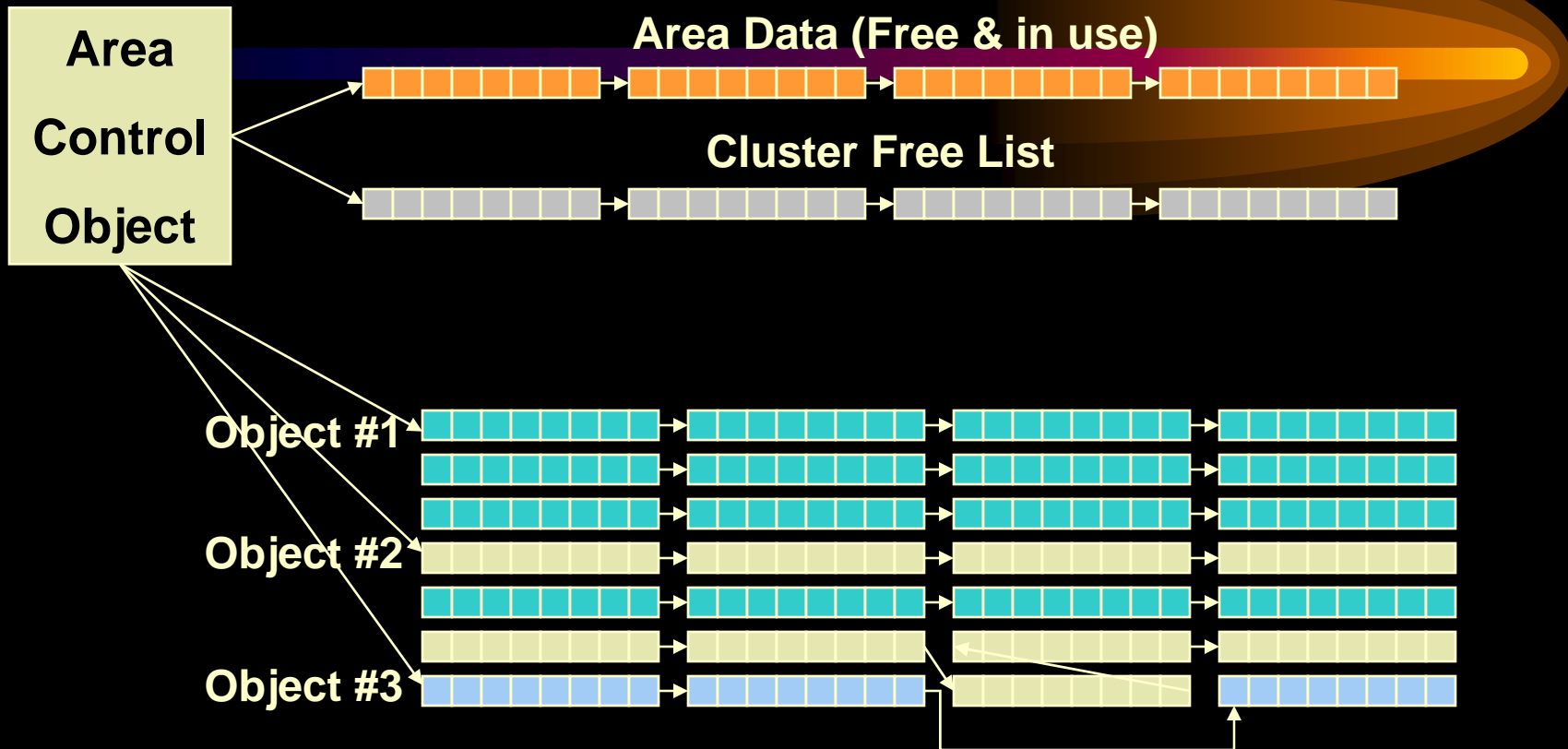


Object Block (Type II Area): Area Free Space Allocation Chains

Area Control Object



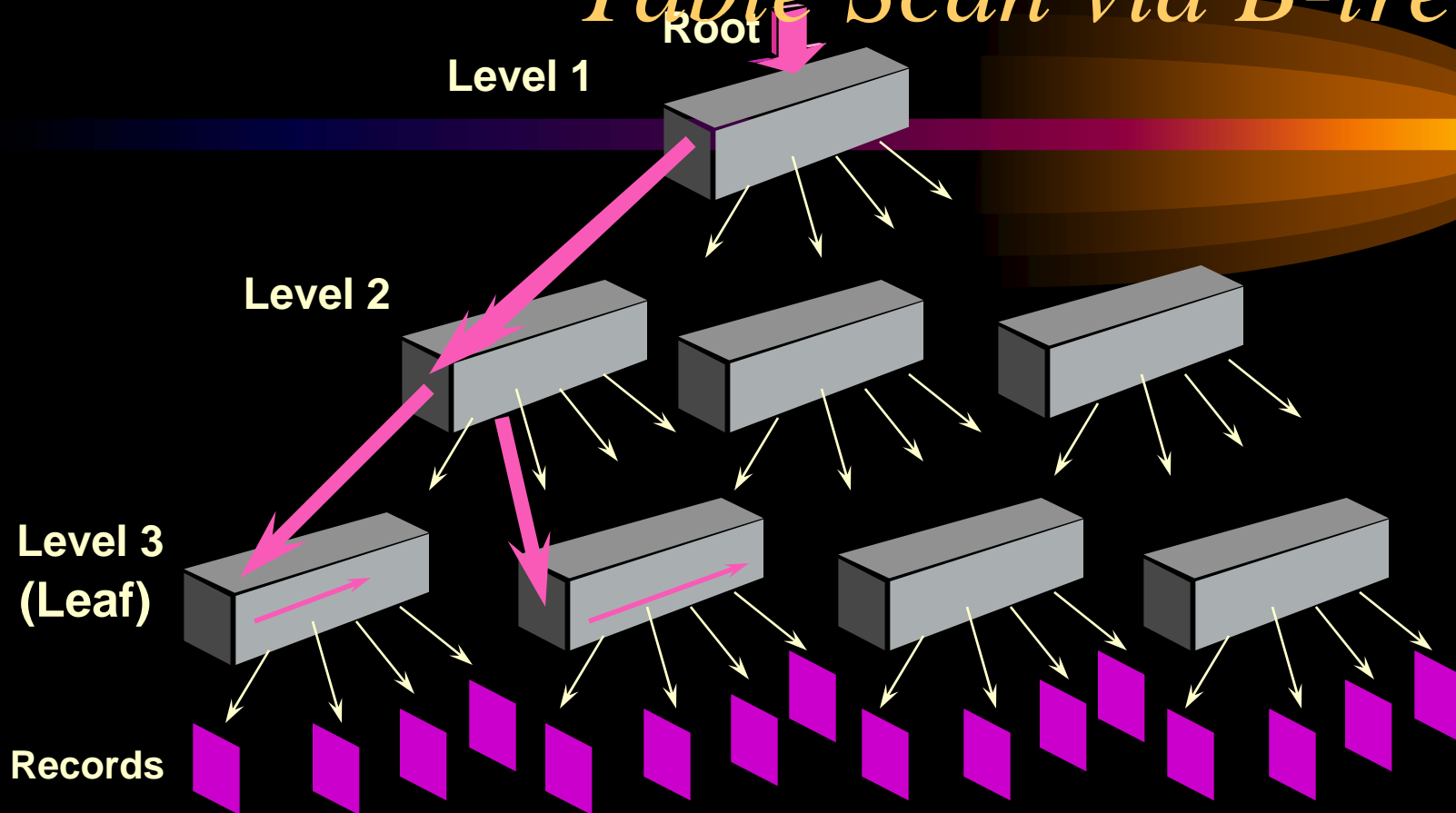
Overall Type II Layout



What Else Is Different w/Type II

- Area HWM increased a cluster at a time
 - More efficient block formatting
 - Concurrent Space Allocation
- Database extend
 - MAX(64, cluster size)
- Other
 - Reduced fragmentation and scatter
 - Allows more advanced tools to be designed

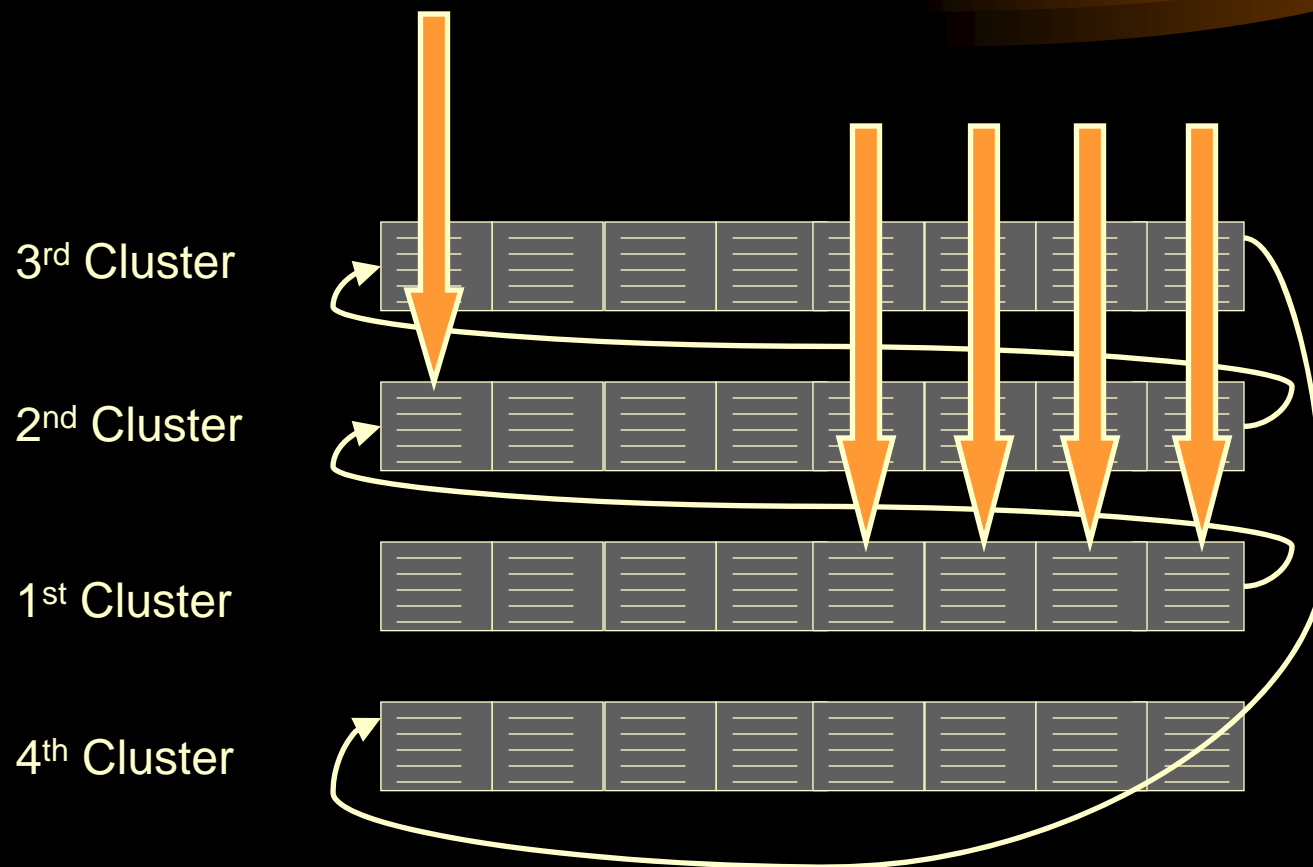
Table Scan via B-tree



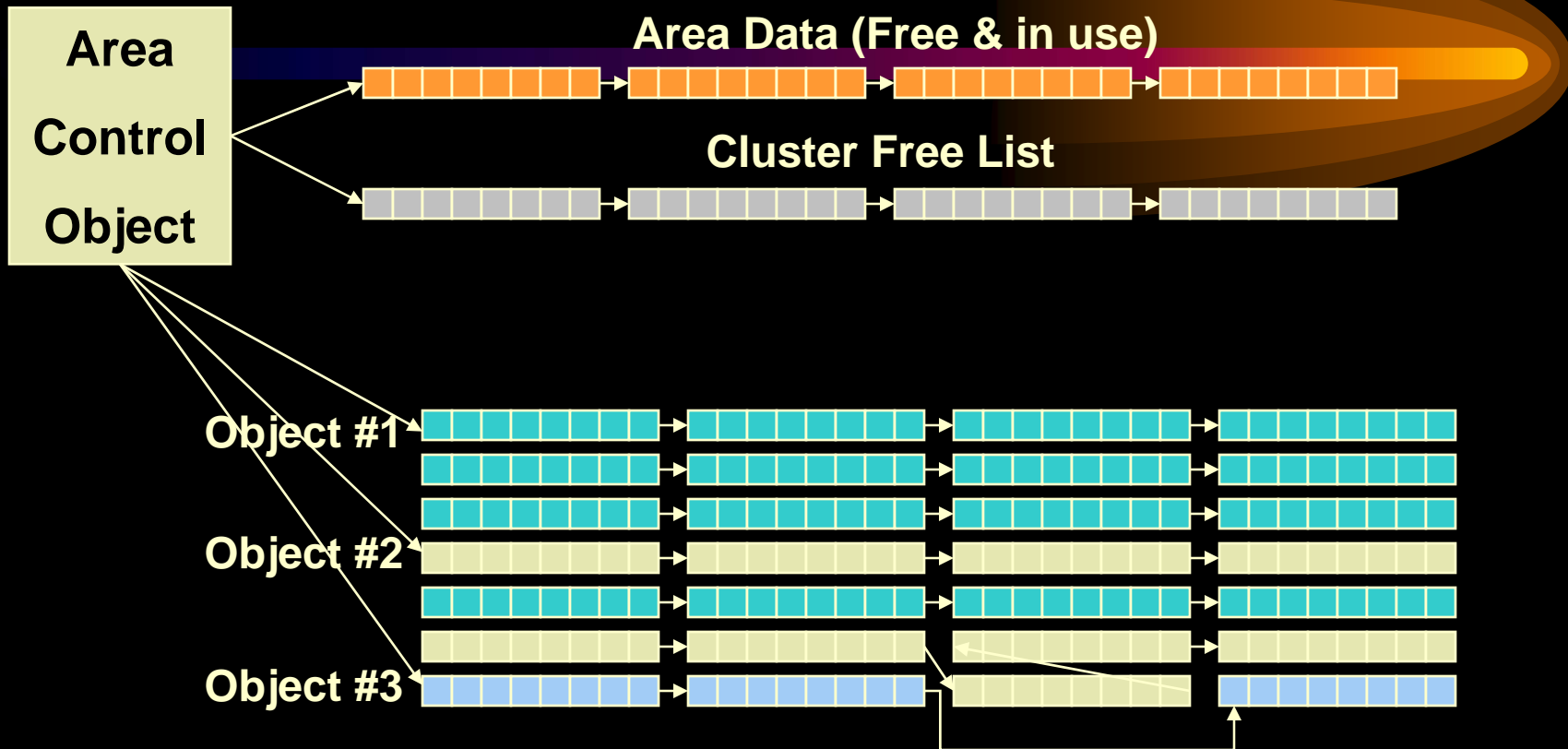
- Leaf entries contain pointer to record
- Cursor maintains info or last key accessed

*Select * from Customer;*

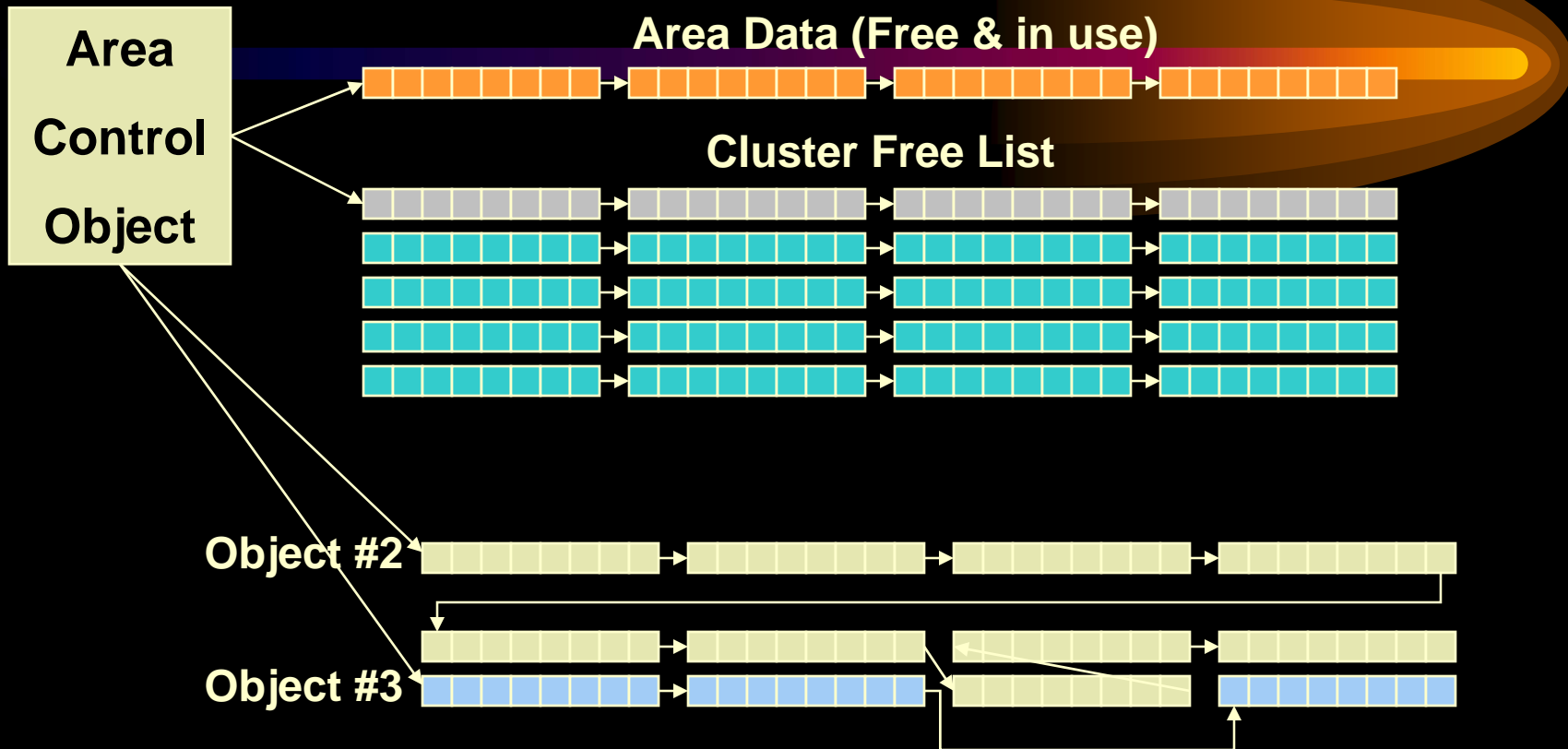
- Cursor maintains info of last record accessed
- I/O Sequential through cluster



Fast Object Delete



Fast Object Delete



OpenEdge 10 Temp tables

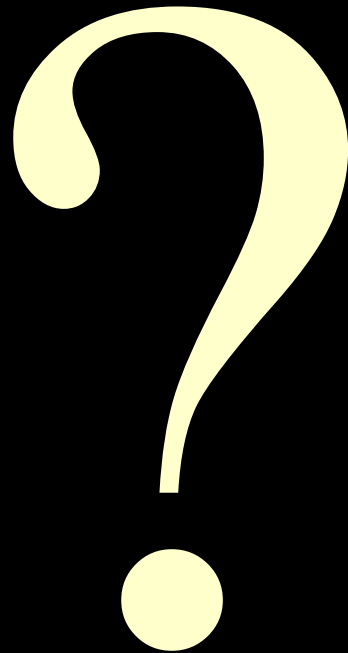
- Released in 10.0b
 - **Fast delete**
 - **Fast delete/create on empty**
- Enhanced in 10.0b02
 - **Avoid delete/create**
 - **Avoid I/O when formatting**
- Hybrid Type I & II Storage Area
 - **Index Objects Type I**
 - **Other Objects Type II**
 - **8 Block Clusters**

Agenda



- Setting the Stage
- The big bucket
- Many small buckets
- Enter the sandwich bags
- **Questions?**
- Take-Homes

Questions Before Wrap-up



Agenda



- Setting the Stage
- The big bucket
- Many small buckets
- Enter the sandwich bags
- Questions?
- **Take-Homes**

For More Information



- Progress documentation
- Progress kbase
- PEG Archives
- Dan Foreman's *V9 and V10 Database Administration Jumpstart* publications
- Dan Foreman's *Database Administration Guide* publication
- Dan Foreman's DBA Resource Kit

Take-Homes



- Storage areas make life easier and easier
- They enable better and better tools
- Like anything, if you set it up wrong, it will actually be worse!

For Further Information



*For a copy of this presentation, leave me
your business card with “Storage” on it*

Scott M. Dulecki

Bravepoint

616/957-3184

sdulecki@bravepoint.com