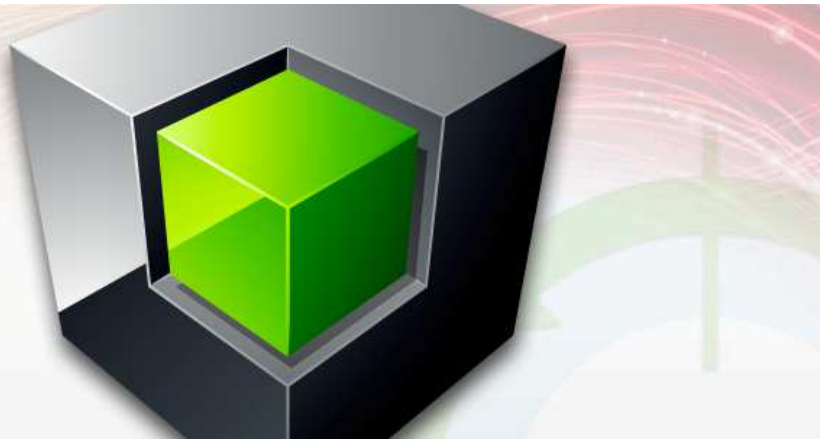
IBM Informix v12.10
- Technical Features

Jan Musil
jan_musil@cz.ibm.com
Informix CEE Technical Sales



Acknowledgements and Disclaimers

Availability. References in this presentation to IBM products, programs, or services do not imply that they will be available in all countries in which IBM operates.

The workshops, sessions and materials have been prepared by IBM or the session speakers and reflect their own views. They are provided for informational purposes only, and are neither intended to, nor shall have the effect of being, legal or other guidance or advice to any participant. While efforts were made to verify the completeness and accuracy of the information contained in this presentation, it is provided AS-IS without warranty of any kind, express or implied. IBM shall not be responsible for any damages arising out of the use of, or otherwise related to, this presentation or any other materials. Nothing contained in this presentation is intended to, nor shall have the effect of, creating any warranties or representations from IBM or its suppliers or licensors, or altering the terms and conditions of the applicable license agreement governing the use of IBM software.

All customer examples described are presented as illustrations of how those customers have used IBM products and the results they may have achieved. Actual environmental costs and performance characteristics may vary by customer. Nothing contained in these materials is intended to, nor shall have the effect of, stating or implying that any activities undertaken by you will result in any specific sales, revenue growth or other results.

Contents

IBM Informix v12.10 Release

- GA on 26.3.2013

Warehouse
Cluster
Schema
Genero
SDS
Connection
PureSystem
SQL
Purge
OLTP
Application
Union
Group
Cloud
Primary
HTML5
Embed
Queues
Dynamic
VTI
Enhancement
Spatial
Support
Flexible
Replication
Storage Manager
Grid
Flexible
Rolling
XML
Accelerator
Simple
Window
Backup
Dashboard
Automatic
Data
Unicode
Performance
Libraries
ANSI
Mobile
Data Mart
OLAP
Partition Refresh
Smart
Data
Trickle Feed
Network
Storage
Compression
SDS
Administration
Server
Ease of Use
Region

Informix v12.10 Themes

- **IN-MEMORY ANALYTICS**
  - Informix Warehouse Accelerator

- **SMART DATA**
  - Big Data and adoption of TimeSeries

- **CLOUD**
  - Hypervisor, Replication and Cluster technology

- **EMBED**
  - Embeddability, SQL / OLAP Compatibility

- **MODERNIZATION**
  - Graphical Administration, Mobile OAT, Genero

Informix v12.10 Themes

- **IN-MEMORY ANALYTICS**
  - Informix Warehouse Accelerator

- **Highlights**
  - Significant SQL enhancement in support of operational analytics on Informix
  - Industry standard SQL/OLAP functions
  - Significant performance improvement in analytics queries
  - Ability to update or refresh partitions in IWA
  - Load and run IWA from a Secondary server
  - Support Union queries
  - Refresh IWA mart with single command
  - OAT support for IWA

Informix v12.10 Themes

- **Theme SMART DATA**
  - Big Data and adoption of TimeSeries

- **Highlights**
  - Easier adoption of TimeSeries
  - Significant performance improvements using virtual View for inserts/deletes & query processing
  - Automatically maintain a window of Time-Series data
  - GUI for developing Time-Series loader and applications
  - Ease of data management
  - Reducing logging for faster data upload
  - Replicate Time Series data (ER and HDR support)
  - Controlled writing to time series containers
  - Flexibility to develop a custom load programs and use of command line utility to load time series data
  - Reduced application development efforts

Informix v12.10 Themes

- **Theme CLOUD**
  - Hypervisor, Replication and Cluster technology

- **Highlights**
  - PureAS offering for Informix
  - Enhanced resiliency and data availability
  - Business continuity ensured
  - Achieve a true data consolidation model
  - Ability to manage and monitor the status of ER Queues
  - Perform data replication on servers with different owners
  - Integration with Storage Provisioning enhances Self-Healing capabilities
  - Ability to break Grid into manageable Regions
  - Query distributions across multiple nodes in a grid
  - Propagate non-database objects across Grid
  - Perform SDS failover when network connection lost
  - Manage Cluster in the event of network outage

Informix v12.10 Themes

- **Theme EMBED**
  - Embeddability, SQL / OLAP Compatibility

- **Highlights**
  - Compression technology has new dimensions
  - New Storage Manager helps embed BAR solutions
  - DDL support for time-cyclic data management
  - Easily configure an embedded server
  - Configure the Server dynamically on the fly
  - Enhanced SQL/SPL, Enriched SQL/OLAP sets
  - Extended support for XML functionality
  - ANSI Joins are now way faster
  - Improved application compatibility
  - Efficiently manage database with Rolling Windows
  - Raises the bar eliminating limitations/restrictions

Informix v12.10 Themes

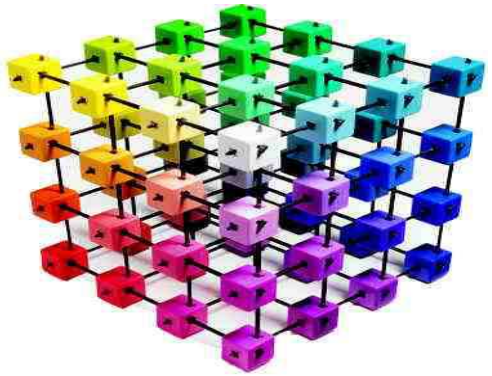
- **Theme Modernization**
  - Graphical Administration, Mobile OAT, Genero

- **Highlights**
  - Makes OAT an integrated administration interface for all Informix operations
  - Enhancements to **plug-ins** help achieve deeper integration across features
  - Graphical administration to most of the new features
  - OAT has a all new Welcome page to greet users and Multi-server Dashboard helps view status summary for a group of servers
  - Administer OAT as users other than “Informix”
  - Server information at finger tips, Available on Smart phones
  - Accommodates the manageability functionalities of ISAO tool
  - Genero v2.41 has new templates for BAM and enhancements done to HTML5 theme
  - Schedule automatic backups using PSM

IBM Informix v12.10
In-Memory Analytics

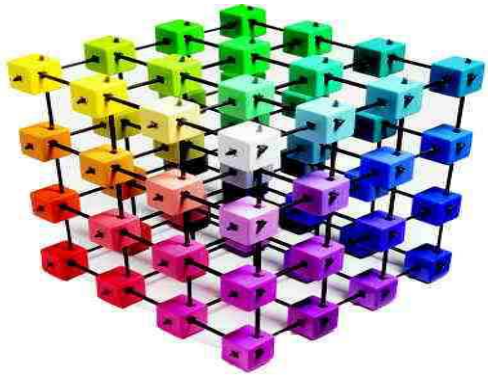
Informix Warehouse Accelerator (IWA)

- **Automatic Partition-Level Refresh**
  - Easy partition-level delta refresh of data marts in IWA
- **Trickle Feed**
  - Continuous incremental loading of IWA data marts
  - For real-time / right-time in-memory analytics
- **Support for Time Series data**
  - For in-memory analytics on data coming from your sensors and meter devices



Informix Warehouse Accelerator (IWA) (cont'd)

- **Additional SQL and OLAP for analytics in IWA**
  - UNION queries and standard OLAP functions
  - More analytic queries & applications will run 100x faster



- **IWA administration using OpenAdmin Tool**
  - Easy IWA administration using graphic interface

Contents

- **OLAP functions (Windowed Aggregates)**
- **Hash Join for ANSI Join improvements**
- **Multiple DISTINCTs**
- **Data Refresh Methods in IWA**
  - Partition Refresh
  - Auto Partition Refresh
  - Trickle Feed
- **TimeSeries to IWA Integration**
- **Grid Support for IWA**
- **Union Support**
- **OAT support for IWA**

SQL OLAP Functions (Windowed Aggregates)

- **ANSI SQL analytics functions**
- **Helps create running totals, aggregates based on custom partition, window and order for each aggregate.**
- **Simplify the expression of complex but commonly used business questions and are efficient without resorting to subqueries or other calculations**
- **Avoids need for complex SQL involving multiple derived tables and joins and hence improves query performance**

The OLAP functions supported by Informix are as follows:

Ranking	RANK(), DENSE_RANK(), DENSERANK(), CUME_DIST(), PERCENT_RANK(), NTILE()
Numbering	ROW_NUMBER(), ROWNUMBER()
Aggregate	RATIO_TO_REPORT, RATIOTOREPORT
First/Last	FIRST, LAST

SQL OLAP Concepts

- **OLAP Window Partitions**
  - A window partition is a set of rows that are grouped together for the purpose of applying an OLAP function.
  - The OLAP function is applied to every row, but is calculated with respect to the rows in the partition.
  - If no partition is specified, the OLAP function is computed over the complete intermediate result set.

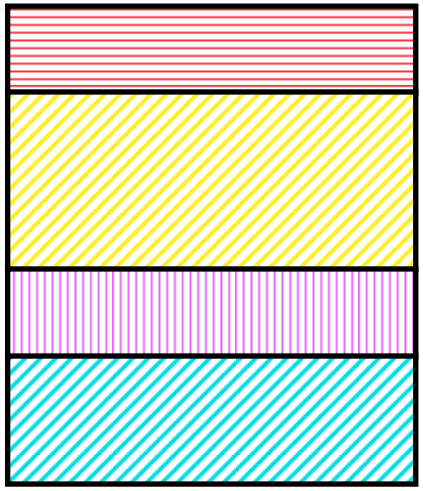
- **The OVER Clause**
  - The OLAP OVER clause differentiates OLAP functions from other analytic or reporting functions
  - The OLAP OVER clause has three distinct capabilities:
    - Defining window partitions (PARTITION BY clause)
    - Ordering rows within partitions (ORDER BY clause)
    - Defining window frames (ROWS/RANGE specification)

SQL/OLAP Analysis Window

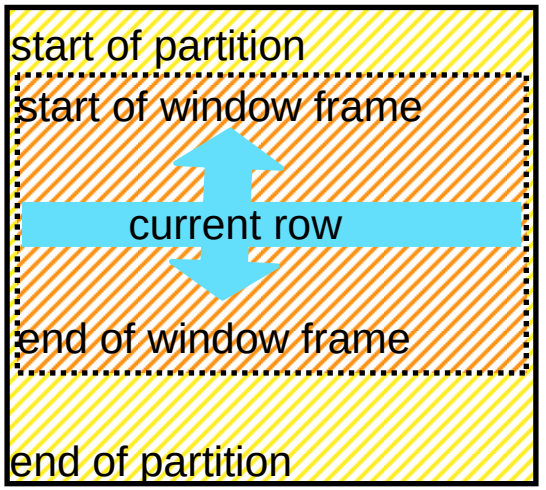
- **The analysis window over which a SQL/OLAP function is analyzed is defined by the OVER clause...**
  - window partition clause
  - window order clause
  - window frame clause

```
sum(x) over (
partition by a, b
order by c, d
rows between 2 preceding
  and 2 following)
```

Multiple Partitions

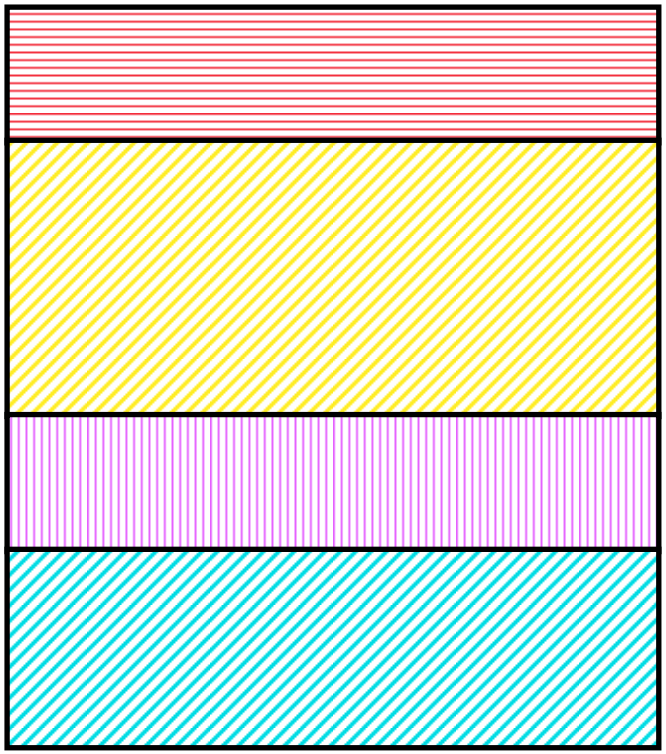


Within Each Partition



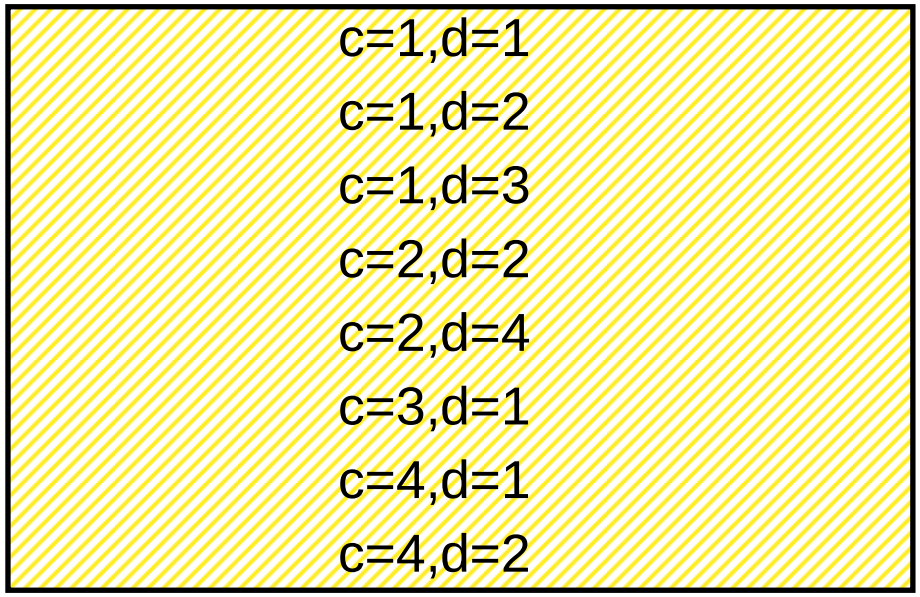
Window Partition Clause

```
sum(x) over (
  partition by a, b
  order by c, d
  rows between 2 preceding and 2 following)
```



Window Order Clause

```
sum(x) over (
  partition by a, b
  order by c, d
  rows between 2 preceding and 2 following)
```



partition a=1, b=2

Window Partition Example

```
Select date, store_name, sum(dollars) as sales_dols,
  rank () over (partition by date order by sales_dols desc) as date_rank
    from period, store, sales
    where period.perkey = sales.perkey
       and store.storekey = sales.storekey
       and state = "CA"
     group by date, store_name
     order by date;
```

DATE	STORE_NAME	SALES_DOL	DATE_RANK
2012-01-02	Beaches Brew	785.55	1
2012-01-02	Roasters, Los Gatos	762.25	2
2012-01-02	Los Gatos Roasting Company	636.25	3
2012-01-02	Cupertino Coffee Supply	634.0	4
2012-01-02	Instant Coffee	457.75	5
2012-01-03	Instant Coffee	713.75	1
2012-01-03	Los Gatos Roasting Company	633.25	2
…..	….	….	..
2012-01-04	Instant Coffee	1031.50	1
2012-01-04	Los Gatos Roasting Company	613.95	2
……	….	….	..

SQL OLAP – Window Frames

- **Row-Based Window Frames**
  - The reference point for all window frames in the current row. The SQL OLAP syntax provides mechanisms for defining a row-based window frame as any number of rows preceding and/or following the current row.
- **In the following example, rows 1 through 5 represent a partition;**
  - Each row becomes the current row as the OLAP window frame slides forward.
  - The frame in this case is defined as between current row and 2 following, so each frame includes a maximum of three rows and a minimum of one row.
  - When the frame reaches end of the partition, only the current row is included
  - Shaded areas indicate which rows are excluded from the frame at each step

1	Current Row	//////////////////////	//////////////////////	//////////////////////	//////////////////////
2	Current Row+1	Current Row	//////////////////////	//////////////////////	//////////////////////
3	Current Row+2	Current Row+1	Current Row	//////////////////////	//////////////////////
4	//////////////////////	Current Row+2	Current Row+1	Current Row	//////////////////////
5	//////////////////////	//////////////////////	Current Row+2	Current Row+1	Current Row

Window Frame Example (By Rows)

- **The sliding calculations produce a moving average with an interval of three rows or fewer, depending on which row is the current one**

```
Select row_number () over () as row,
   dimension,
    measure,
    avg (measure) over (partition by dimension)
         order by measure
         rows between current row and 2 following)
         as OLAP_AVG
 from ….
```

Row	Dimension	Measure	OLAP_AVG
1	A	10	53.3
2	A	50	90.0
3	A	100	240
4	A	120	310
5	A	500	500

Window Frame Example (By Range)

- **The following partial result set demonstrates the concept of a value-based window frame:**
  - The frame consists of rows that:
  - Have the same year as the current row
  - Have the same year as the current row *minus 1*

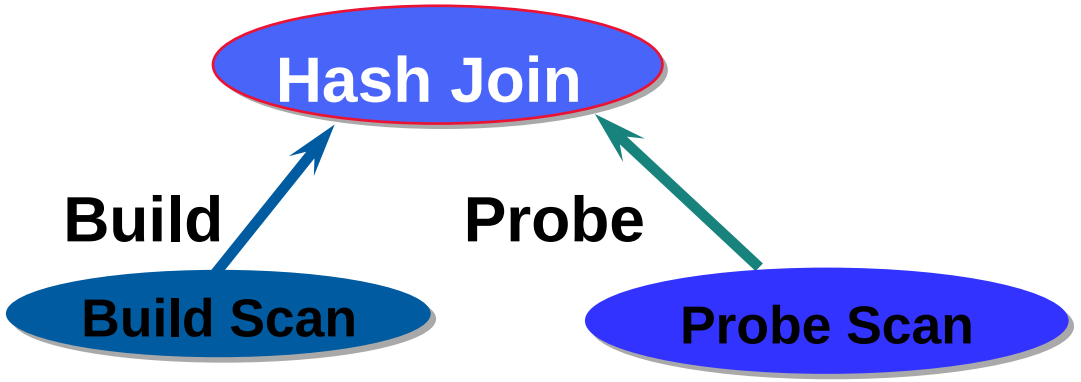
- **The SQL for this query would look like this:**

```
select row_number() over () as row, dimension, year, measure,
        avg (measure) over (partition by dimension)
                order by year asc
        range between current row and 1 preceding as olap_avg
from …
```

Row	Dimension	Year	Measure	Olap_Avg
1	A	1999	10000	10000
2	A	2001	5000	3000
3	A	2001	1000	3000
4	A	2002	12000	5250
5	A	2002	3000	5250

Hash Joins for BI Queries

- Hash join requires equality join filters.
- In the build phase, a hash table is built by applying a hash function on join keys from table with lower cardinality.
- In the probe phase, the other table is then scanned, and joined using the hash table.
- More efficient than nested loop join when large number of rows have to joined, typical in BI queries.



Hash Join for ANSI Joins – Page 1

- **ANSI standard SQL specifies four types of JOIN: INNER, OUTER, LEFT, and RIGHT.**

- **Prior to this release:**
  - Hash joins were considered only for inner joins, “informix outer” joins and not for ANSI joins
  - We would try to rewrite the Left Outer Join (LOJ) as inner join when we could

- **Cognos and other BI tools generate left outer joins extensively.**

- **Now supports:**
  - Hash joins for left outer joins, with all the view folding
  - Improved cost estimation of hash join access path

Hash Join for ANSI Joins – Page 2

ANSI (or Explicit) syntax

```
SELECT *
FROM employee
INNER JOIN department
ON employee.DepartmentID = department.DepartmentID;
```

Informix (or Implicit) syntax

```
SELECT *
FROM employee, department
WHERE employee.DepartmentID = department.DepartmentID;
```

Multiple aggregates with DISTINCTs and DISTINCT with Case Expression

- **Multiple Aggregates with DISTINCTs**

```
SELECT region, sum(DISTINCT cid),
               avg(DISTINCT salesdt)
FROM    sales_tab
GROUP BY region
ORDER BY region;
```

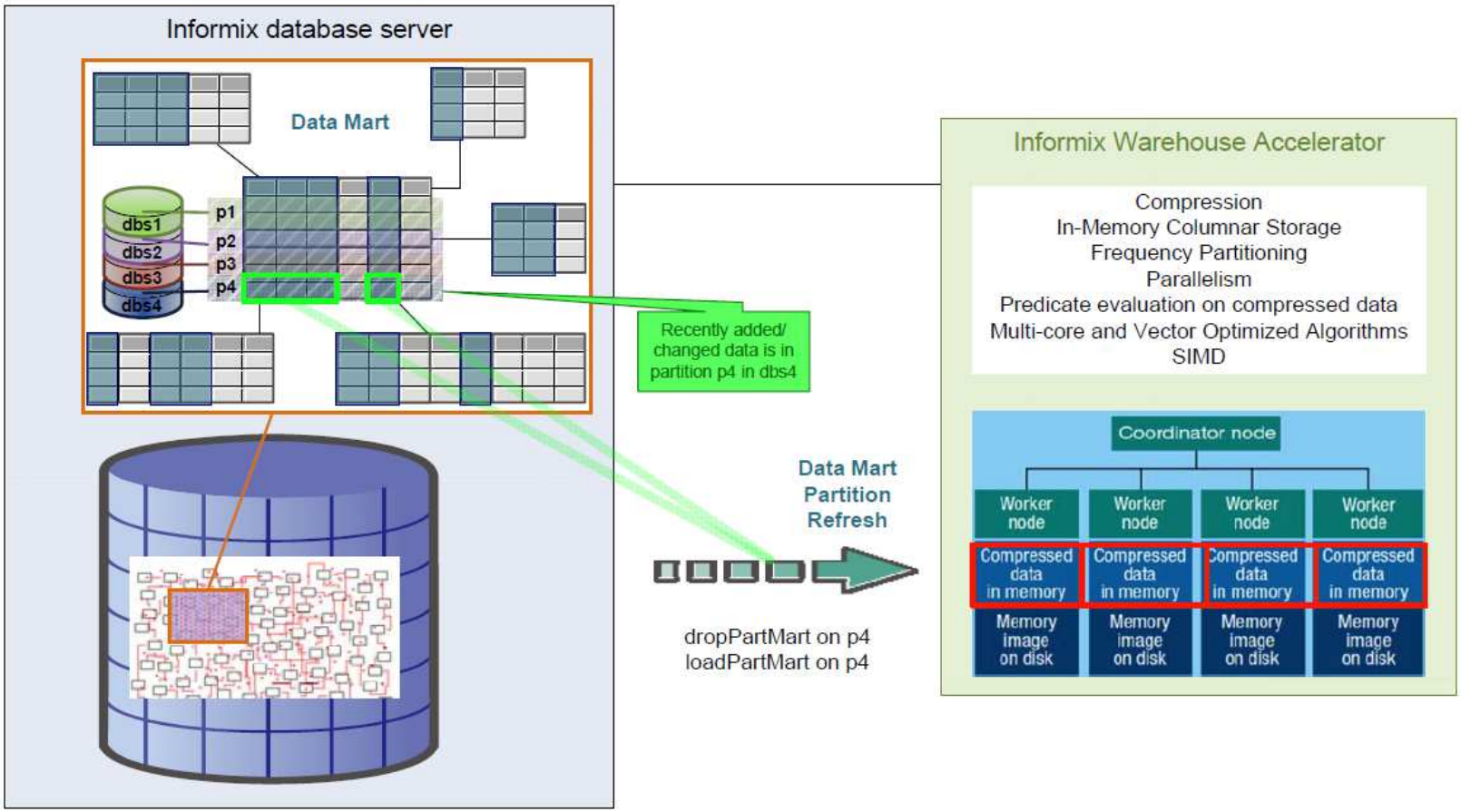
- **Support for DISTINCT with CASE expression**

```
SELECT  sum(T983271.set_avgday_sales_rtl_amt) as c2,
                count(distinct case
                            when T983271.not_set_cnt > 0
                            then T983271.store_sk_id end ) as c3
FROM features_upc_tab T983271;
```

Data Refresh Methods in IWA

Methods	Frequency of Deployment	Considerations
Complete Load	Nightly or a small number of times per day	For initial load and for periodic complete refresh of mart
Automatic or manual Partition Update	Hourly, nightly or on-demand	For partitioned FACT table For non-partitioned or partitioned dimension tables
Trickle Feed	Continuous	For Inserts (Append) to FACT table and I/U/D for dimension tables

Partition Refresh in IWA



Partition Refresh - Example

Example: Time-cyclic based roll-out and roll-in of data on salesfact. Detach an old partition and attach a partition

- **Drop the partition p1 from data mart salesmart in the accelerator SLSACC. Detach the old partition p1 from dmuser.salesfact table to a standalone table salesfact_p1.**

```
EXECUTE FUNCTION ifx_dropPartMart(
        'SLSACC',
        'salesmart',
        'dmuser',
        'salesfact',
        'p1');
  ALTER FRAGMENT ON TABLE salesfact
  DETACH p1 salesfact_p1;
```

Partition Refresh – Example (cont'd)

Example: Time-cyclic based roll-out and roll-in of data on salesfact. Detach an old partition and attach a partition

- **Now, attach a table p4 that contains recent data, as a new partition p4 into the Informix table dmuser.salesfact and then load this fragment into your data mart salesmart in the accelerator SLSACC.**

```
CREATE TABLE p4 (   sk1_cust    INT,
                    sk2_store   INT,
                     id         INT,
                     val        DECIMAL (9,2),
                    CHECK       (ID=4));
ALTER FRAGMENT ON TABLE salesfact
ATTACH p4 AS (id = 4) AFTER p3;


EXECUTE FUNCTION ifx_loadPartMart(
 'SLSACC', 'salesmart', 'dmuser', 'salesfact', 'p4');
```

Automatic Partition Refresh

- **Current Partition Refresh (as of 11.70xC5) requires that the user manually perform the following:**
  - Identify which partitions are to be refreshed
  - Identify which partitions are to be dropped
  - Identify which partitions are to be added

- **Then for each partition affected, drop the partition and load the partition**

- **Ideally, keeping a data mart up-to-date (and minimizing total reload), one would want to:**
  - Do the initial load (entire mart)
  - Schedule [on a regular basis] a single refresh mart by partition
  - On a regular basis [e.g. weekly basis], do a full load

Automatic Partition Update (cont'd)

```
ifx_refreshMart (<accelerator name>, <mart name>,
                 [locking mode], threshold]);
```

where

lockmode = NONE, TABLE, or MART (similar to *loadMart*) and data mart is available for acceleration when *lockmode* is NONE or TABLE

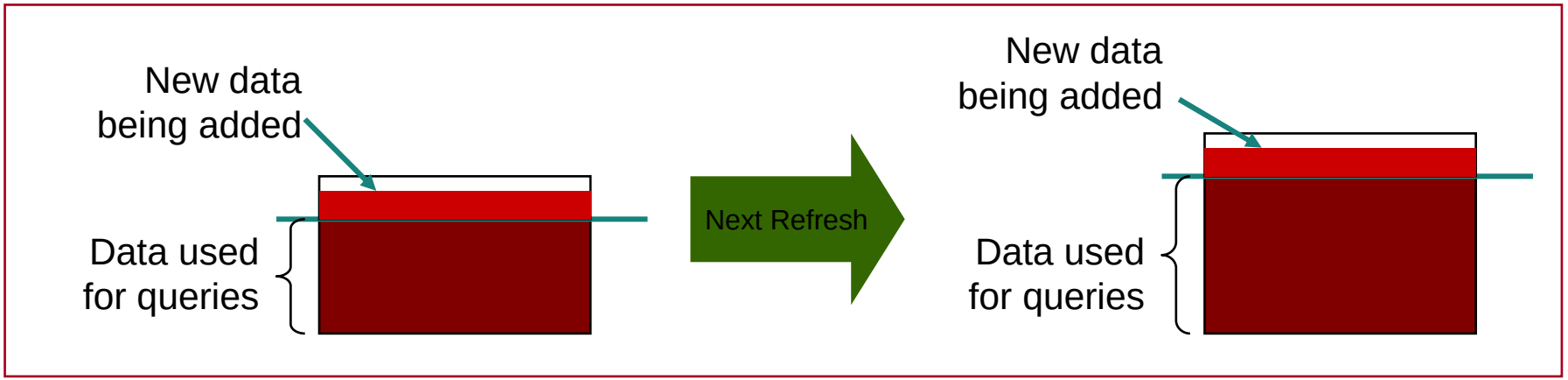
Threshold is a value for amount of changes before the refresh is initiated. A '0.0' value means that refresh is done for any change

Data differential = total number of data change operations / total number of rows

(Note: this is not changed rows / total rows)

Trickle Feed

- **Trickle feed techniques allow continuous update of the data.**
- **With Real-time Data Warehousing, it allows for continuous update to the database**
- **Different implementations by different vendors over the years, e.g. Red Brick, Oracle, but concept is the same**

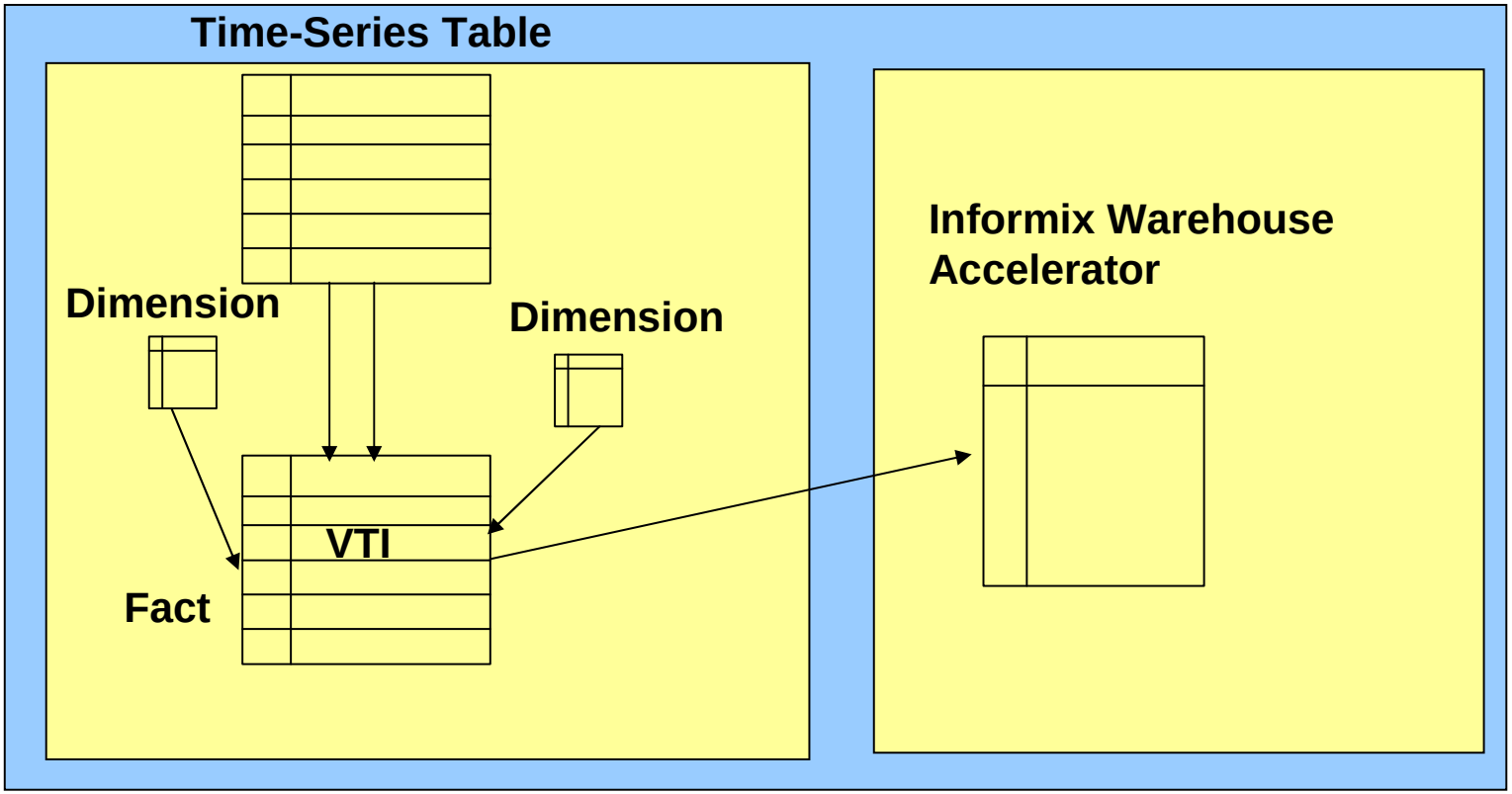


Trickle Feed in Informix/IWA

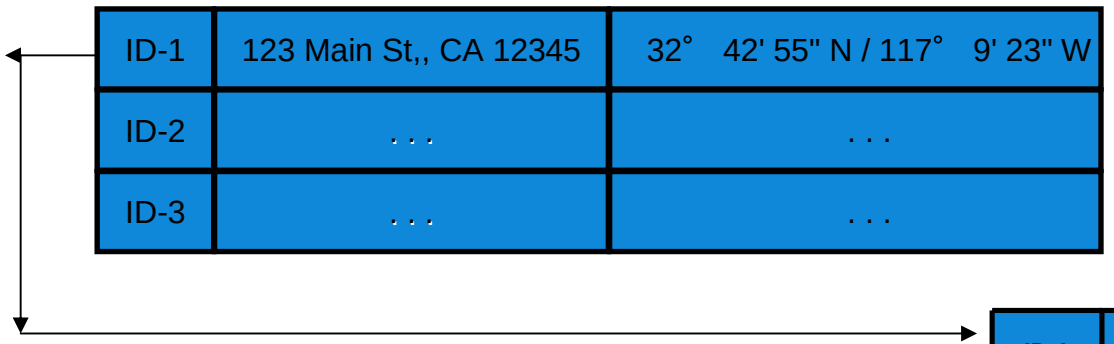
- **Operation: Update to Informix is continuously fed to IWA**
  - FACT table inserts (Append)
    - Trigger on FACT table to capture row and save in external file
  - Dimension tables
    - Supports Insert/Update/Delete
  - DBScheduler to push data to IWA
    - For Inserts on FACT table
    - Partition Refresh on dimension tables
- **Setup: Execute stored procedure once to set up Trickle Feed**
  - Stored Procedure ifx_setupTrickleFeed
    - Parameters for
      - Accelerator name, Datamart name,
      - time interval ***n*** before flushing buffer
  - Set up via Stored Procedure or OAT
- **Consideration:**
  - IWA dictionary unchanged with Trickle Feed
  - Partition Update operations cannot be executed while Trickle Feed is turned on

Time-Series to IWA Integration

- **Native Time-Series data can be loaded directly into IWA**
  - without materialization (on disk)
- **Provides for generalized business analytics on time-series data across different dimensions (not just time dimension)**
- **Queries may join other dimension tables to the Fact (VTI) table**



Storage Needed to Store Time-Series Data (Using Standard Relational)



ID-1	123 Main St,, CA 12345	32° 42' 55" N / 117° 9' 23" W
ID-2	. . .	. . .
ID-3	. . .	. . .

DeliveryPoint Table

ID-1	2011-01-01 08:30:00	0.45
ID-1	2011-01-01 08:45:00	0.32
ID-1	2011-01-01 09:00:00	0.34
ID-1	2011-01-01 09:45:00	0.40
ID-1	2011-01-01 10:00:00	0.39
ID-1	2011-01-01 10:15:00	0.41

MeterData Table

- **ID in each record (8 bytes)**
- **Timestamp on each record (12 bytes)**
- **Index on ID and timestamp (20 bytes)**

Useful data: **8 bytes**
Overhead: **40 bytes**

Storage Needed to Store Time-Series Data (TimeSeries)

ID-1	123 Main St,, CA 12345	32° 42' 55" N / 117° 9' 23" W	(0.45), (0.32), (0.34), (0.40), (0.39), (0.41)
ID-2	. . .	. . .	
ID-3	. . .	. . .	

DeliveryPoint Table

TimeSeries

- **No ID duplication**
- **No Timestamp**
- **No Index on ID and timestamp**

Useful data: **8 bytes for each record**
Overhead: **Negligeable**

TimeSeries commonly uses 2 to 3 times less space than standard relational

Virtual Table Interface Makes Time Series Appear Relational

TimeSeries Table

Smart_meter

mtr_id (int)	Series timeseries(mtr_data)
1	[(Mon, v1, ...)(Tue,v1…)]
2	[(Mon, v1, ...)(Tue,v1…)]
3	[(Mon, v1, ...)(Tue,v1…)]
4	[(Mon, v1, ...)(Tue,v1…)]
5	[(Mon, v1, ...)(Tue,v1…)]
6	[(Mon, v1, ...)(Tue,v1…)]
7	[(Mon, v1, ...)(Tue,v1…)]
8	[(Mon, v1, ...)(Tue,v1…)]

TimeSeries Virtual Table

SM_vt

mtr_id	date	col_1	col_2	
1	Mon	Value 1	Value 2	...
1	Tue	Value 1	Value 2	...
1	Wed	Value 1	Value 2	...
...	...	...	...	...
3	Mon	Value 1	Value 2	...
3	Tue	Value 1	Value 2	...
3	Wed	Value 1	Value 2	...
...	...	...	...	...

Execute procedure tscreatevirtualtable ('SM_vt', 'Smart_meter');

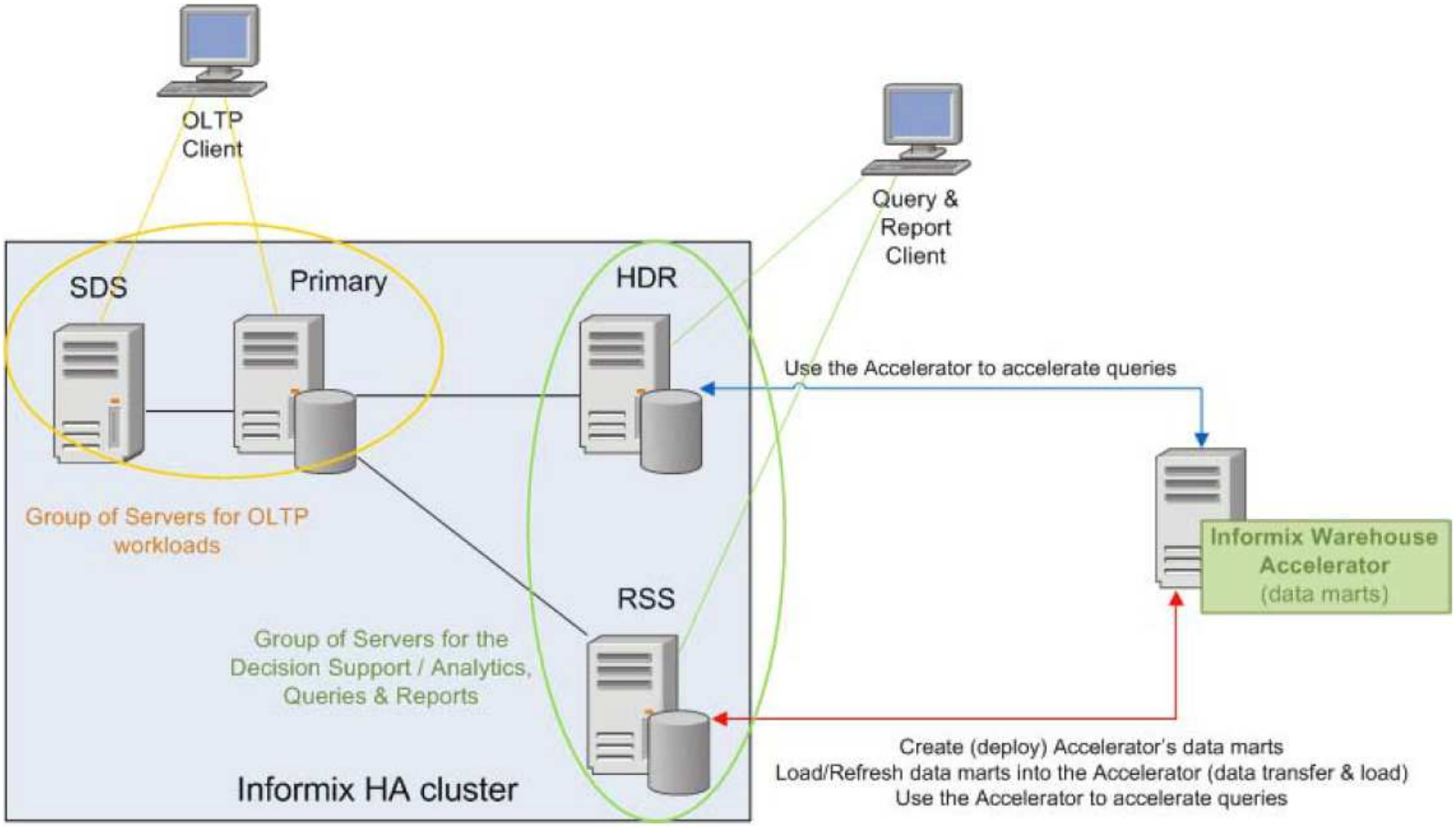
Steps for Loading Time-Series data to IWA

- **Declare the VTI table to contain time-series**
  - Example

```
create table my_vti_table(
       meterid      int,
       ts           datetime year to fraction(3),
       v1           float,
       v2           float)
USING ts_vtam   (
       basetabname='ts_meter_readings',
       tsfirstcolidx='2',
       tscolcnt='4',
       tsscanmode='0',
       tscolname='meter_data',
       tselemtype='meter_type'  );
```

- **Include the VTI table (in this example, "my_vti_table") as part of the IWA data mart, specifying the VTI table as the FACT table**
- **Perform IWA loading using standard IWA load, i.e. ifx_load_mart()**

Using IWA with Secondary Servers



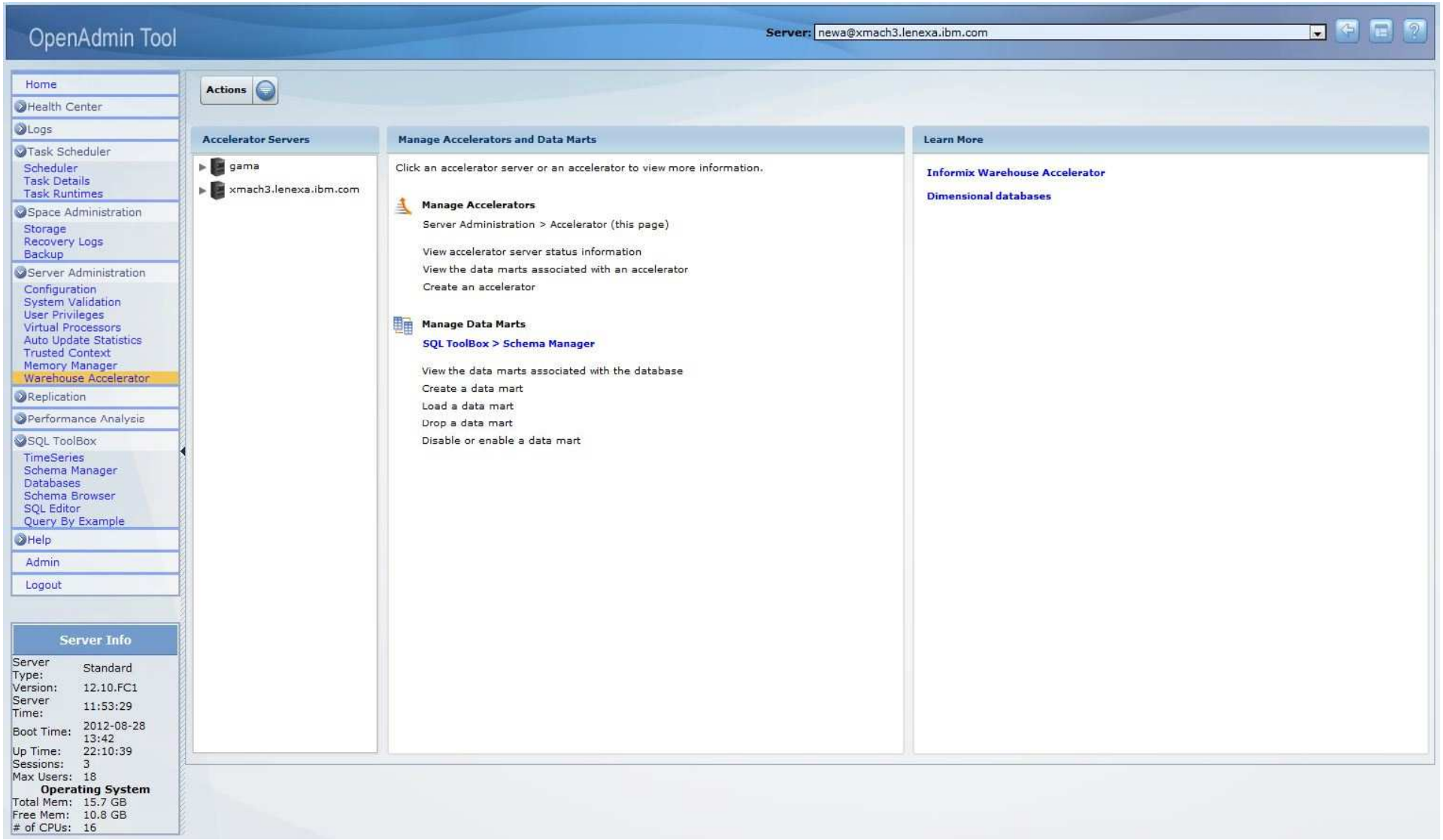
Support Union Queries for IWA

```
Select first 100 i_item_id,
        avg(ss_quantity) agg1,
        avg(ss_list_price) agg2,
        avg(ss_coupon_amt) agg3,
        avg(ss_sales_price) agg4
from store_sales,
customer_demographics, date_dim,
item, promotion
where  ss_sold_date_sk = d_date_sk
and
        ss_item_sk = i_item_sk and
        ss_cdemo_sk = cd_demo_sk and
        ss_promo_sk = p_promo_sk and
        cd_gender = 'F' and
        cd_marital_status = 'M' and
        cd_education_status =
'College' and
        (p_channel_email = 'N' or
         p_channel_event = 'N') and
         d_year = 2001
group by i_item_id
```

UNION ALL

```
Select first 100 i_item_id,
        avg(ws_quantity) agg1,
        avg(ws_list_price) agg2,
        avg(ws_coupon_amt) agg3,
        avg(ws_sales_price) agg4
from web_sales,
customer_demographics, date_dim,
item, promotion
where  ws_sold_date_sk = d_date_sk
and
        ws_item_sk = i_item_sk and
        ws_cdemo_sk = cd_demo_sk and
        ws_promo_sk = p_promo_sk and
        cd_gender = 'F' and
        cd_marital_status = 'M' and
        cd_education_status =
'College' and
        (p_channel_email = 'N' or
         p_channel_event = 'N') and
         d_year = 2001
group by i_item_id
```

OAT Support on IWA



Customer Quotes (EVP Program)

As a provider of ERP solutions to medium-sized businesses for over 25 years and as an IBM Informix Partner, Deister has certified that Informix 12.10 is a giant leap on features and performance. ***New standard Datawarehouse SQL syntax allows to create complex SQL timebased computations, increasing overall performance of processes*** *and simplifying programming and software maintenance.*

- Vicente Salvador, DEISTER, S.A.

IDS 12.10 : "vNext have great new features we start to include in our Database Managed Service. The expanded compression functionality is easy to use and have a great lasting effect for the whole system. Based on the ***new 12.10 warehouse feature we extend leolo's XPS to IDS database and application migration tools to a 99% automatic migration process."***

- Henri Cujass, CTO leolo IT and Media Consulting GmbH

IBM Informix v12.10
SMART DATA

Smart Data (TimeSeries and Spatial)

- **Enhance Usability of TimeSeries Capability**
  - Graphical tool (eclipse plug in) simplifies complexity
  - Enhances ease of use and improved productivity

- **Enhanced SQL interface to the TimeSeries data**
  - Easier to manipulate TimeSeries through standard SQL interface.
  - Reduced learning curve to adapt TimeSeries and speed up implementation

- **TimeSeries rolling windows**
  - Greater capacity for each TimeSeries element
  - Efficient management and removal of range of values

Smart Data (TimeSeries and Spatial), cont'd

- **Support for TimeSeries data within Flexible Grid**
  - Enable TimeSeries data to be replicated across a Grid
- **SPATIAL: Release with ESRI SDE 10.1 Libraries**
  - Enables easy and cost effective spatial applications
  - Applications can manipulate both temporal and geospatial data
- **Overall performance improvements**
  - Concurrent writing to containers, Reduced logging

Agenda

- Overview
- Time Series data in Rolling Window Containers
- Load data faster by reducing logging
- Replicate Time Series data (ER and HDR support)
- Control writing to time series containers
- Write a custom program to load time series data faster
- Load time series data from an external database
- User command line to load time series data faster

Overview

- **This presentation includes technical information about new features and changes in existing TimeSeries functions after 11.70**

- **The changes includes focus in the are of**
  - Ease of use
  - Performance improvement
  - Replication
  - Ease of data management
  - Ease of porting

Time Series data in rolling window containers

- **Store data in partition by date interval**
- **Automatically delete old/obsolete data**
- **Create rolling window container -** TSContainerCreate
  - Specify Maximum size for active and dormant window
  - Enable Automatic delete of old partition
  - Specify multiple dbspaces to store partitions
- **Manage rolling window container –** TSContainerManage
  - Change window size
  - Attach or detach partitions
  - Destroy partitions
  - Enable/disable automatic deletion of partition
  - Change extent size of partition

Rolling window containers…. (cont'd)

- **Estimate amount of storage required**
  - space = [*primary_key* + *index_entry* + (*time_series_columns* * *elements*)] * (*table_rows*) + *B-tree_size*
- **Estimate container partition size**
  - *Space = (container_name_length + dbspace_name_length + 48) * (active_windowsize + dormant_windowsize) * 2*
- **Estimate window partition size**
  - Approximate number of partitions in each dbspace = CEIL((active_windowsize + dormant_windowsize) / number_dbspaces) + 1

Rolling window containers…. (cont'd)

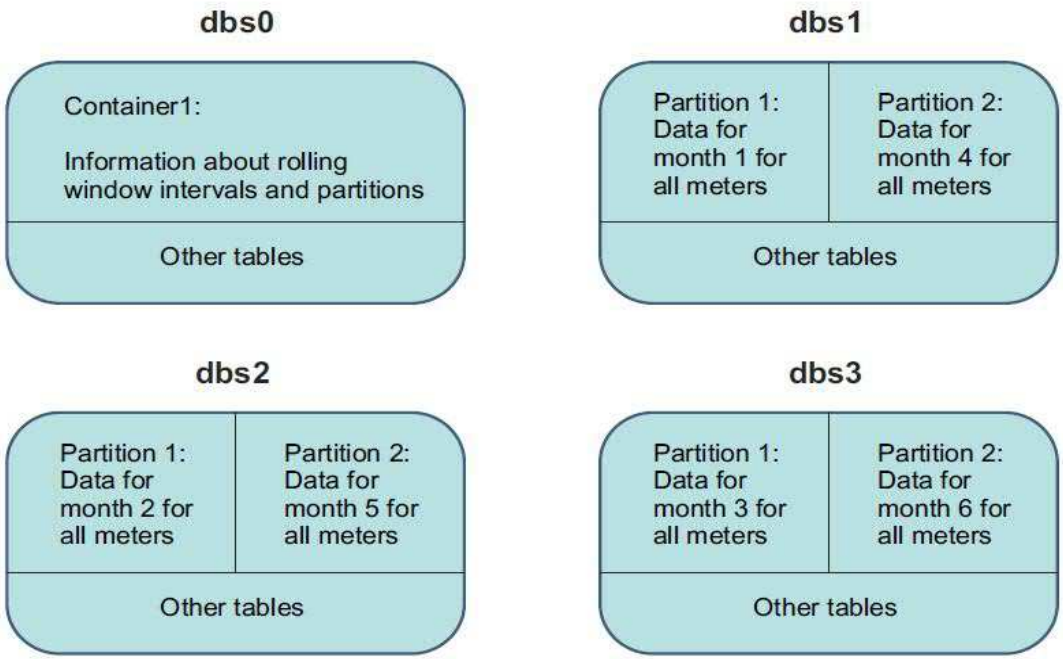
- **Create container with rolling window**

```
TSContainerCreate(container_name varchar(128,1),
   dbspace_name varchar(128,1),
   ts_type varchar(128,1),
   container_size integer,
   container_grow integer,
   window_origin datetime year to fraction(5),
   window_interval lvarchar default 'month',
   active_windowsize integer default 0,
   dormant_windowsize integer default 0,
   window_spaces lvarchar(4096) default null,
   window_control integer default 0,
   rwi_firstextsize integer default 16,
   rwi_nextextsize integer default 16
   );
```

Rolling window containers…. (cont'd)

- **Create container with rolling window**

```
execute procedure TSContainerCreate(
'readings_container', 'containerdbs', 'rt_raw_intvl',
25600, 12800, '2011-01-01 00:00:00.00000'::datetime year
to fraction(5),'month', 4, 10, 'dbs0, dbs1, dbs2, dbs3,
dbs4', 1, 16, 8);
```



Rolling window containers…. (cont'd)

- **Manage containers**
  - Move active window to dormant window

```
execute function TSContainerManage(
"readings_container", "detach active partitions before
2012-01-08");
```

  - Move dormant window to active window

```
execute function TSContainerManage(
"readings_container", "attach dormant partitions after
2012-01-06");
```

  - Increase active window size to 10

```
execute function TSContainerManage(
"readings_container", "set active window size to 10");
```

  - Destroy partition in dormant window

```
execute function TSContainerManage(
"readings_container", "destroy dormant partitions
before 2012-01-08");
```

Rolling window containers…. (cont'd)

- View information of rolling window containers
- When you create a rolling window container, a row is inserted in the **TSContainerTable** and the **TSContainerWindowTable** table.
- As partitions are added for time series data, rows are added to the **`TSContainerUsageActiveWindowVTI`** and the **`TSContainerUsageDormantWindowVTI`** tables
  - The **`TSContainerUsageActiveWindowVTI`** contains information about the partitions in the active window.
  - The **`TSContainerUsageDormantWindowVTI`** contains information about the partitions in the dormant window.

- **One can manage time series through Open Admin Tool (OAT) as well**

Load data faster by reducing logging

- **Every insert statement generates two log records each one for:**
  - Inserted element and
  - Page header
- **Page header logging can be done for each transaction instead by setting TSOPEN_REDUCED_LOG flag to 256 or TS_VTI_REDUCED_LOG flag to 256**
- **TSOPEN_REDUCED_LOG can be used only with functions**
  - BulkLoad
  - InsElem
  - PutElem
  - PutElemNoDups
  - PutNthElem
  - PutTimeSeries
  - TSL_Flush

Load data faster by reducing logging… (cont'd)

- **TS_VTI_REDUCED_LOG can be used with virtual table**
  - Create virtual table called **daily_stocks_no_ts**, based on **daily_stocks** with reduced logging:

```
execute procedure TSCreateVirtualTab(
'daily_stocks_no_ts', 'daily_stocks',256);
```

- **Restrictions**
  - Insert must run with transaction
  - The transaction can include other functions that uses this flag
  - The transaction can not include other functions or SQLs that do not use this flag
  - The elements are not visible in dirty read mode until after the transaction commits

Replicate time series data

- **Time series data can be replicated with:**
  - High Availability Data Replication (HDR)
  - Enterprise Replication (ER)
- **Time series data can not be replicated with**
  - Remote Stand-alone Secondary (RSS)
  - Shared Disk Secondary (SDS)
  - HDR secondary server that allow updates
- **Can not replicate time series data with Change Data Capture API**

Replicate time series data with HDR

- **Time series data can be replicated with read-only HDR secondary server**
- **No pre-requisites other than regular HDR setup requirements**
- **No restrictions on replicating time series data**

Note: Because some time series calendar and container information is kept in memory, stop replication before you drop and then re-create your calendar or container definitions with the same names but different definitions

Server Preparation for ER

- **All participant servers should be on Informix 12.10 or later**
- **ER for TimeSeries only works on containers created in 12.10**
- **Perform following tasks on all participant server**
  - Set CDR_TSINSTANCEID on all participant servers for different values onconfig.std value
  - If not set, replication of TimeSeries columns is not allowed
  - Range of values
    - 0 - Default. Disable the replication of TimeSeries columns
    - 1 to 32768 - The number that is used to modify the time series instance identifiers
  - Takes effect for new TimeSeries
    - After you edit your onconfig file and restart the database server.
    - When you reset the value dynamically in your onconfig file by running the onmode -wf command.
    - When you reset the value for a session by running the onmode -wm command.

Server Preparation for ER (cont'd)

- Create container with same name on all servers
  - Containers create automatically or with rolling window can not be used
- Create same time series calendars on all servers
- Create time series table on all servers
- Initiate time series instances on all servers

Rules for defining replicates in ER

- **The replicate must be a mastered replicate.**
- **The Projection list in the participant definition must include all columns in the table.**
- **The WHERE clause in the participant definition cannot include a** TimeSeries **column.**
- **Cannot define a participant as send-only.**
- **The conflict resolution rule must be always-apply.**
- **All the containers must be created in 12.10 and above version**

Rules for defining replicates in ER (cont'd)

- **The replication key cannot include an opaque data type.**
- **Cannot enable conversion to and from UTF-8 (Unicode) when you replicate data between servers that use different code sets.**
- **Cannot use the --autocreate option to create tables that have** TimeSeries **columns.**
- **Cannot generate ATS or RIS files in XML format. ATS and RIS files must be in text format.**

Restrictions - ER

- **Flexible grid do not support TimeSeries columns**
- **Cannot run the ifx_grid_connect() procedure and then run a CREATE TABLE or ALTER TABLE statement that includes a** TimeSeries **column**
- **Cannot run the cdr change replicateset command with the --add option to add a replicate that includes a** TimeSeries **column to a grid**
- **Cannot use the following commands on replicates that include** TimeSeries **columns**
  - `cdr alter`
  - `cdr remaster`
  - `cdr start sec2er`
  - `cdr swap shadow`

Restrictions - ER (cont'd)

- **You cannot use the following options when you check or repair inconsistencies on a replicate that includes a** TimeSeries **column**
  - The *--deletewins* option in the **cdr check replicate** or **cdr check replicateset** command
  - The *--extratargetrows=merge* option in the **cdr sync replicate, cdr sync replicateset, cdr check replicate**, or **cdr check replicateset** command
  - The *--since* option in the **cdr check replicate** or **cdr check replicateset** command
  - The *--timestamp* option in the **cdr check replicate** or **cdr check replicateset** command
  - The *--where* option with a **TimeSeries** column in the WHERE clause in the **cdr check replicate** command

Control Writing to time series containers

- **The TSContainerLock procedure controls whether multiple sessions can write to a container at one time.**

- **Syntax**

```
TSContainerLock(container_name varchar(128),flag integer);
```

  - *container_name*
    - The name of the container. Must be an existing container name.
  - *flag* Controls whether multiple sessions can write to the container:
    - 0 = Multiple sessions can write to the container at the same time.
    - 1 = Only one session at a time can write to the container.
- **Default flag value = 0**
- **The following statement restricts the number of sessions that can write to the container named** ctn_sm0 **to 1:**

```
EXECUTE PROCEDURE TSContainerLock('ctn_sm0',1);
```

- **Data is loaded faster if only one session writes to container**

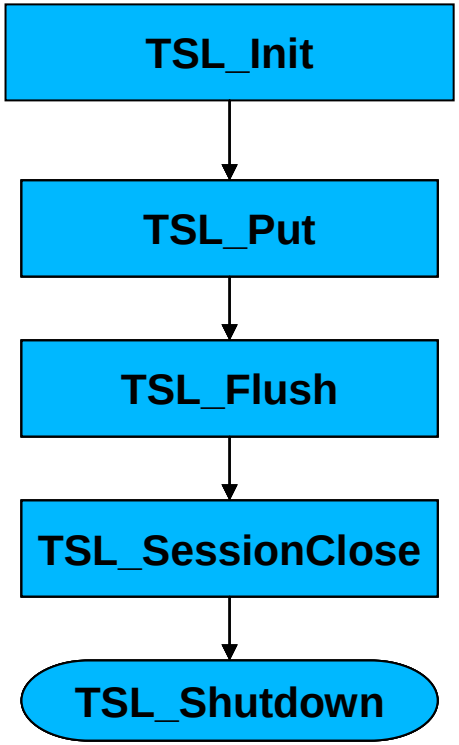
Write customer program to load data faster

- **Loader API for faster data uploaded included in the product**
- **Loader APIs can directly be used in SQLs in application**
- **Pre-requisites**
  - The name of table and TimeSeries column should be in lower case
  - Primary key is defined for each row
  - If primary key is CHAR(*n*), each value should be *n* bytes long OR user VARCHAR(*n*)
  - Data to be loaded should be compatible to TimeSeries data type
  - Data must be compatible to Informix data types

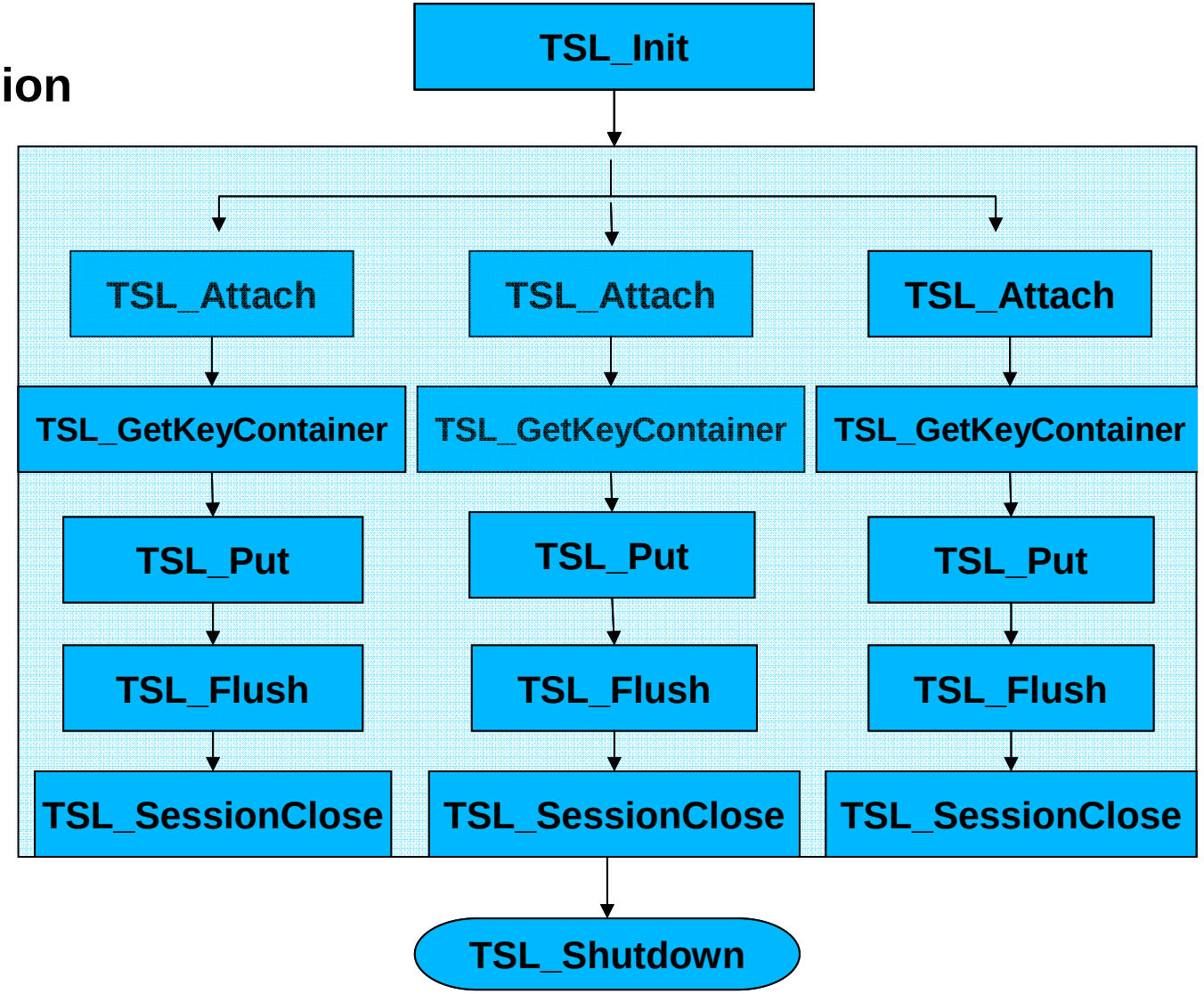
Introduction to Loader APIs

- TSL_Init - Initialize a global context and open a database session
- TSL_Put, TSL_PutRow, or TLS_PutSQL - Copy data from a file or input stream into the database server. Need to run many times.
- TSL_Attach – Open additional database sessions
- TSL_GetKeyContainer – Determine, how to distribute data among database sessions
- TSL_Flush - Save data to disk by running the function. You must run this function after 65536 records are loaded.
- TSL_SetLogMode - change the logging mode
- TSL_GetLogMessage - Monitor the progress of loaded and saved data
- TSL_SessionClose - Close the database session
- TSL_Shutdown - Remove the global context and shut down the loader

Single database session



Multiple database session



Loader Program Example

```
EXECUTE PROCEDURE ifx_allow_newline ('t');

EXECUTE FUNCTION TSL_Init ('ts_data','raw_reads', 3,4, NULL,
  '%Y-%m-%d %H:%M:%S', '/tmp/rejects.log',NULL);

EXECUTE FUNCTION TSL_Put ('ts_data|raw_reads',
'472735432100011|KWH|P|2010-11-10 00:00:00.00000|0.092|
472735432100011|KWH|P|2010-11-10 00:15:00.00000|0.084|
472735432100011|KWH|P|2010-11-10 00:45:00.00000|0.085|
');

EXECUTE FUNCTION TSL_Put ('ts_data|raw_reads',
'472735432109095|KWH|P|2010-11-10 00:00:00.00000|0.026|
472735432109095|KWH|P|2010-11-10 00:15:00.00000|0.035|
472735432109095|KWH|P|2010-11-10 00:45:00.00000|0.092|
');

begin;

EXECUTE FUNCTION TSL_Flush ('ts_data|raw_reads');

commit;

EXECUTE FUNCTION TSL_SessionClose ('ts_data|raw_reads');

EXECUTE PROCEDURE TSL_Shutdown ('ts_data|raw_reads');
```

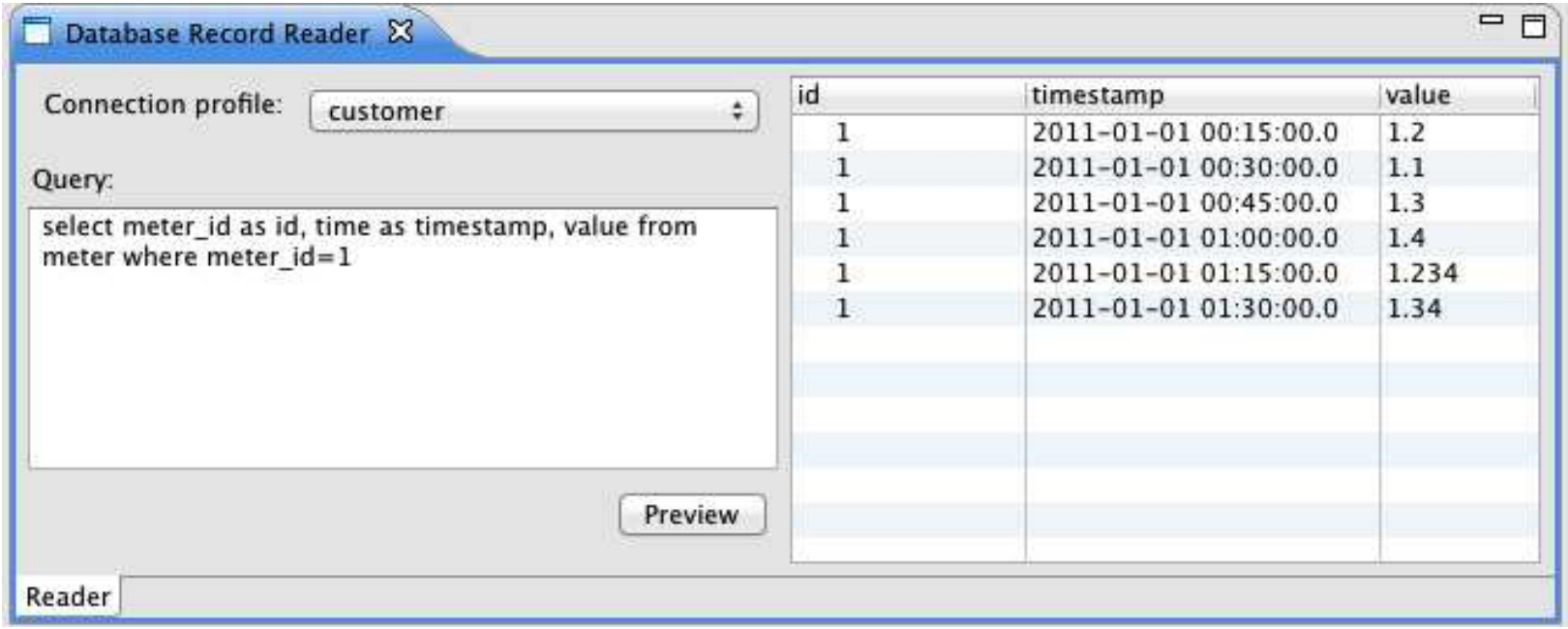
Load time series data from an external database

- **Eclipse-based Tools for Working with TimeSeries Data**
  - Enables rapidly loading TimeSeries data
  - Custom file formats are easily created using graphical interface
  - Works with Optim Data Studio

- **New Features**
  - Loading from another database through JDBC
    - Enables extract-load without intermediary files
  - Non-GUI tooling execution
    - Enables automation in production environments

Loading TimeSeries Data from a Database

- **Select the connection profile for the source database**
  - Defined using New Database Connection Profile
- **Specify a query that results in the data to be loaded**
- **A preview of the record structure that results from the query can be seen by hitting the “Preview” button**



Loading Data from a Non-Graphical Environment

- **Example command-line below**
- **As the load proceeds, statistics are regularly printed to the console**
- **When the load exits, cumulative statistics are printed**

```
./eclipse -application com.ibm.informix.timeseries.loader \
          -recordReader=customer1.udrf \
          -table=loadjob1_table.tbl \
          -map=loadjob1_map.fcmap \
          -connection=loadjob1_conn.xml \
          -loadSettings=loadjob1_prop1.tslp
```

Summary

- Time Series data in Rolling Window Containers for ease of data management and better performance
- Reducing logging for faster data upload
- Replicate Time Series data (ER and HDR support) for consolidation at central location and high availability
- Control writing to time series containers by putting appropriate lock on container for faster data upload
- Write a custom program to load time series data faster for ease of development and fast data upload
- A data studio plug-in to load time series data from an external database eases porting of relation information to time series
- User command line to load time series data faster reduces application development efforts drastically

IBM Informix v12.10
CLOUD

Cloud (Replication and High Availability)

- **Reduced learning curve with ready to use**
  - PureAS offering for Informix
- **Enhanced resiliency, data availability and business continuity**
- **Systematize Replication domain**
  - Achieve a true data consolidation model
  - Ability to manage and monitor the status of ER Queues
  - Perform data replication on servers with different owners
  - Integration with Storage Provisioning enhances Self-Healing capabilities

Cloud (Replication and High Availability)

- **Improve Network failover support**
  - Perform SDS failover when network connection lost
  - Manage Cluster in the event of network outage

- **Orchestrate Grid operations**
  - Ability to break Grid into manageable Regions
  - Propagate non-database objects across Grid
  - Query distributions across multiple nodes in a grid

Announcing IBM Informix Hypervisor Edition

For the IBM PureApplication™ System

- **Add-on System Pattern enabled for easy integration with IBM's PureApplication System:**
  - Built-in expertise to address complex business and operational tasks automatically
  - Integration by design to tune data infrastructure for optimal performance and efficiency
  - Simplified experience for the enterprise ecosystem

IBM Informix Hypervisor for RHEL/AIX

Expert at: *Optimally deploying and running Informix database servers for rapid time-to-value*

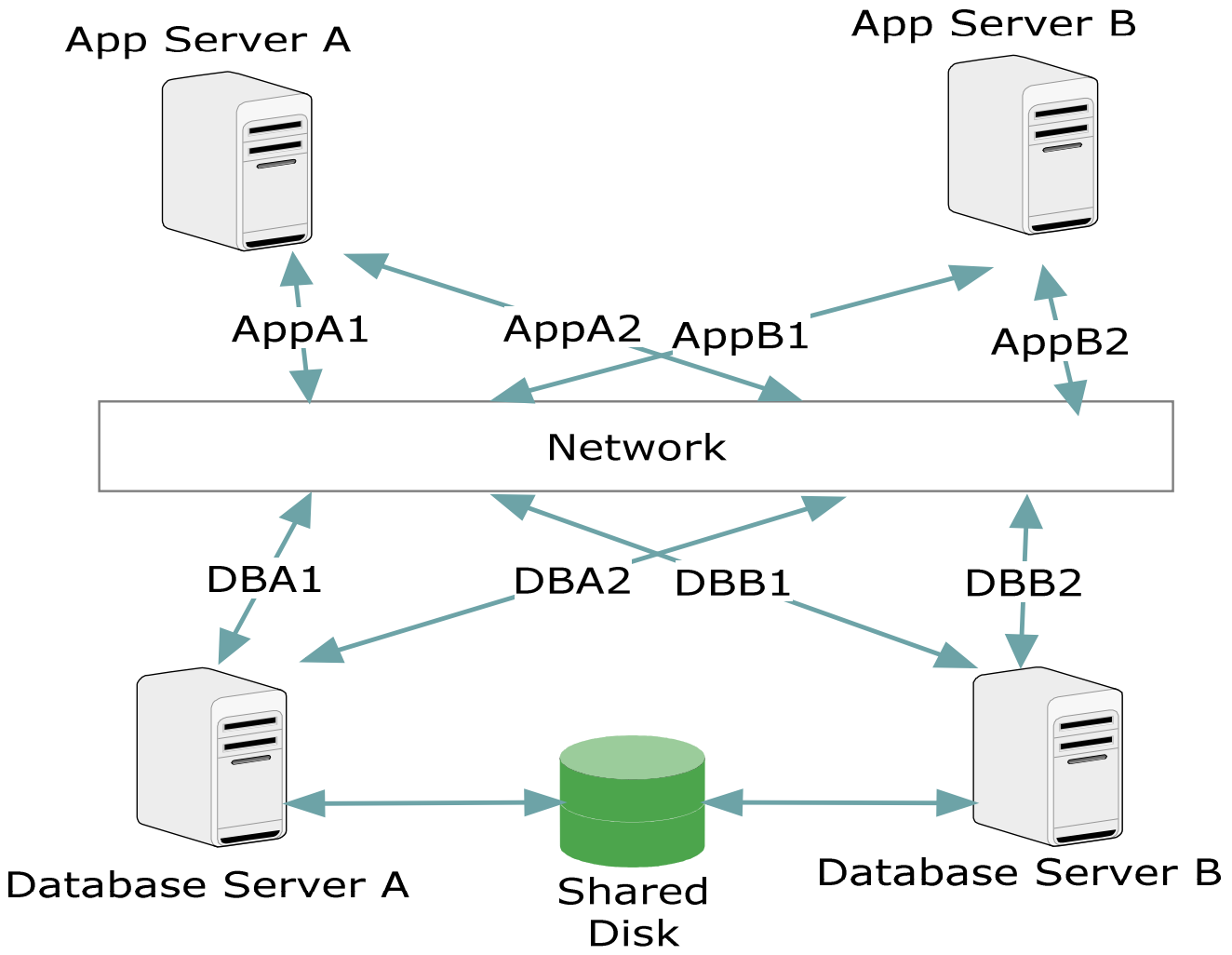
New

- Built using the Informix Ultimate Edition database server
- Enabled and optimized for PureApplication System
- Simplified management with a single console
- Delivers both Hypervisor Image and System Pattern
- Designed for the private cloud with flexibility and simplicity taking advantage of Informix clustering and Flexible Grid technology

Server side support for failover

- Evaluate the impact of a failover based on all network monitor priorities.
- Allow a failover if no higher priority network will lose cluster access.
- Disallow a failover if any higher priority network will lose cluster access.
- Allow an alternate means of communication for cluster with SDS servers whenever the TCP/IP communication is disrupted or unavailable.

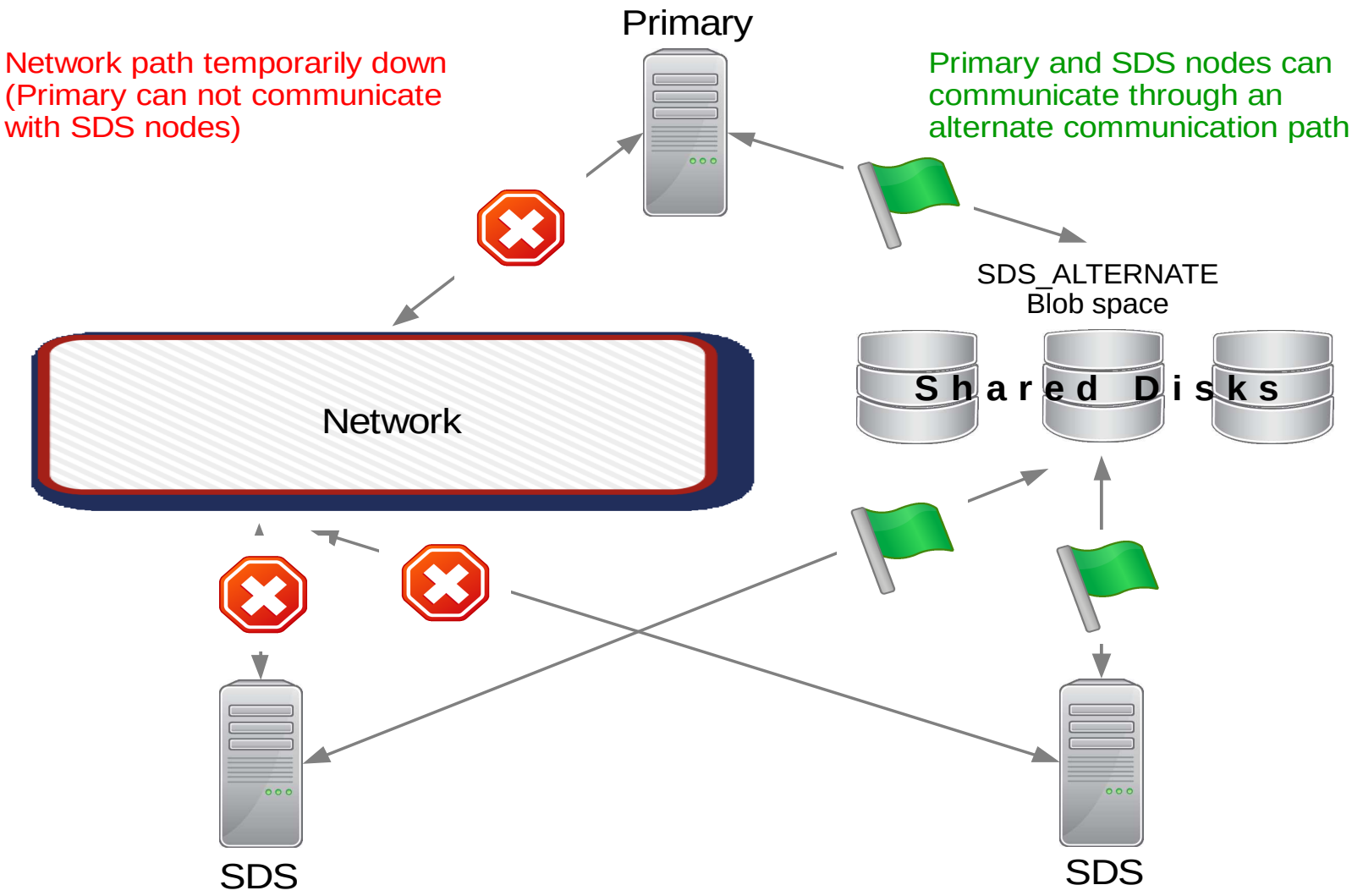
Server side support for failover



SDS Alternate Communication

- SDS_ALTERNATE is set to a dedicated BLOB space name.
- The BLOB space must be created and the logical logs advance to the next log file before initializing SDS nodes.
- When the TCP/IP communication is unavailable, the primary and SDS nodes will communicate via SDS_ALTERNATE.
- When an SDS node is about to failover and become the primary server, but the TCP/IP communication is unavailable, the SDS_ALTERNATE is used to inform the original primary server to perform a shutdown.

SDS Alternate Communication

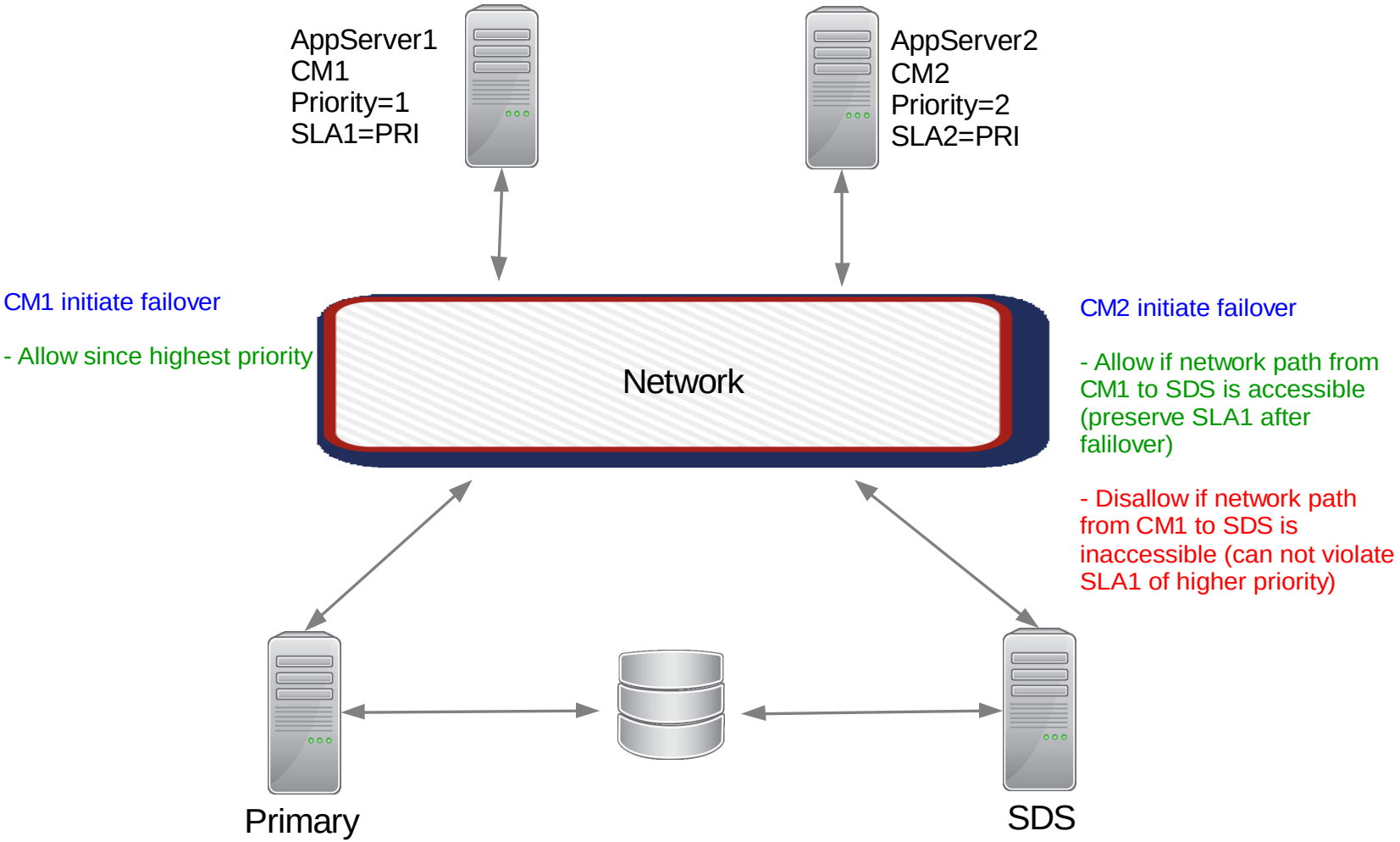


Connection Manager Network Monitor with Server Failover Enhancements

- **Connection Manager (CM) enhancements:**
  - Configure CM to monitor network communications between application servers and database servers.
  - Each application server must run on its own computer and have a Connection Manager instance configured.
  - If the communication between the application server to the primary server is down, a server failover can be initiated.

- **Configuration**
  - LOCAL_IP is a list of network IP addresses to be monitored by a CM
  - PRIORITY is the network monitor priority defined as part of the Failover Configuration (FOC)

Connection Manager Priority Failover



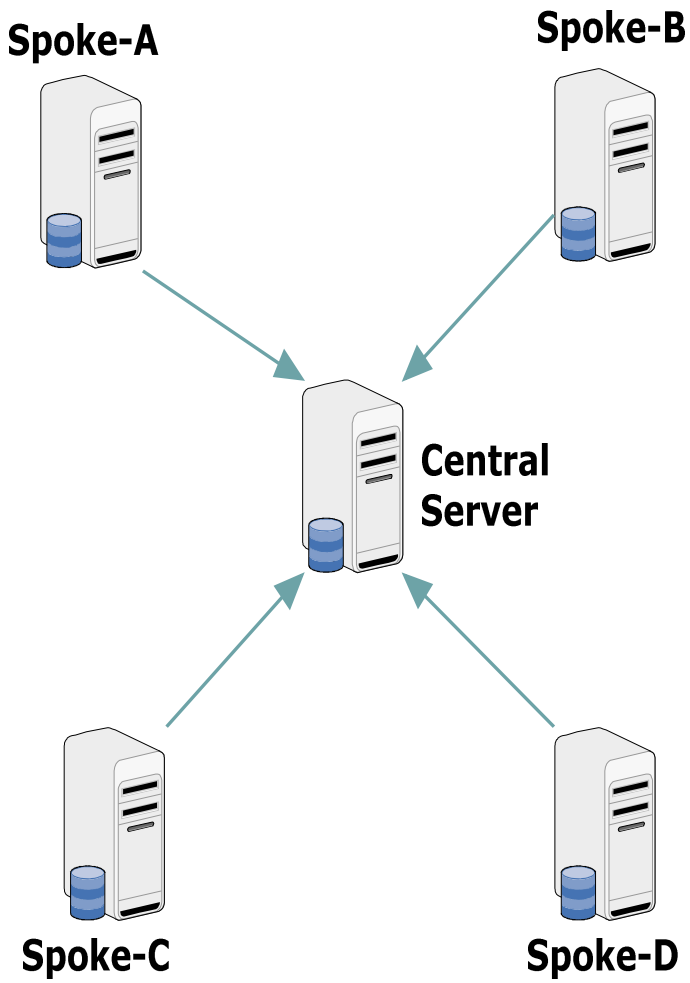
Simplified set up of a data consolidation system

- **Current Setup**
  - Supports ‘Receive Only’ target in the ER setup
  - User must define individual replicates for each spoke to hub table that is being replicated

- **Limitations**
  - Hard or Difficult to create a true consolidation model
    - As primary member will be the source node for replication and thus cannot be ‘Receive Only’, as it needs to receive and apply the data changes made
  - Causes Performance issues
  - Need to define numerous replicates to achieve ‘Receive Only’ model
  - Each replicate requires a lot of memory allocation
  - Causes an undo complexity in the definition of replicates

Simplified set up of a data consolidation system

- Adds ‘Send Only’ a new type of member to the replicate definition
- Forwards the replicated changes to the non-SENDONLY nodes
- Reduces the effort on replicate definition significantly as all nodes can be configured in a single replicate
- Significantly reduces administration efforts and storage requirements
- The only real error condition arises if there exists a row on one of the spoke servers that does not exist on the non-spoke servers.



86

Simplified set up of a data consolidation system

- **Example 1 – Create single SENDONLY node**

```
cdr define repl -c g_er1 -C always -f y price_book \
“S stores@g_er1:informix.my_prices"
     "select * from my_prices where prod_code > 500" \
“stores@g_er2:informix.my_prices" "select * from my_prices" \
“stores@g_er3:informix.my_prices" "select * from my_prices"
```

  - In this case g_er1 has a S participant qualifier indicating the replicate can only send data changes from this replicate, it can not receive updates
  - g_er2 and g_er3 don’t have a participant qualifier indicating the replicate is R/W in these instances

- **Example 2 – Create Multiple SENDONLY nodes**

```
cdr define repl -u -c g_er1 –C timestamp -f y district_claims_processing \
stores@g_er1:informix.district_claims_processing
         "select * from district_claims_processing" \
“S stores@g_er2:informix. claims_processing"
         "select * from claims_processing" \
“S stores@g_er3:informix. claims_processing"
         "select * from claims_processing”
```

  - In this case g_er2 and g_er3 instances are sending their activity up to g_er1

Enterprise Replication Queue Monitoring

- **As a DBA**
  - I have just issued a replicate definition on my ER network
  - I now want to perform initial synchronization

- **However**
  - I can't start the synchronization unless the definition is propagated to all servers on my ER network
  - How do I get to know the status of replication?
  - Has the propagation completed by now?
  - How long should I wait, before I execute my next command?
  - What happens if I issue a query without the knowledge on the previous query status / results?

Enterprise Replication Queue Monitoring

- **Use '*cdr check queue*' command to monitor ER queues**

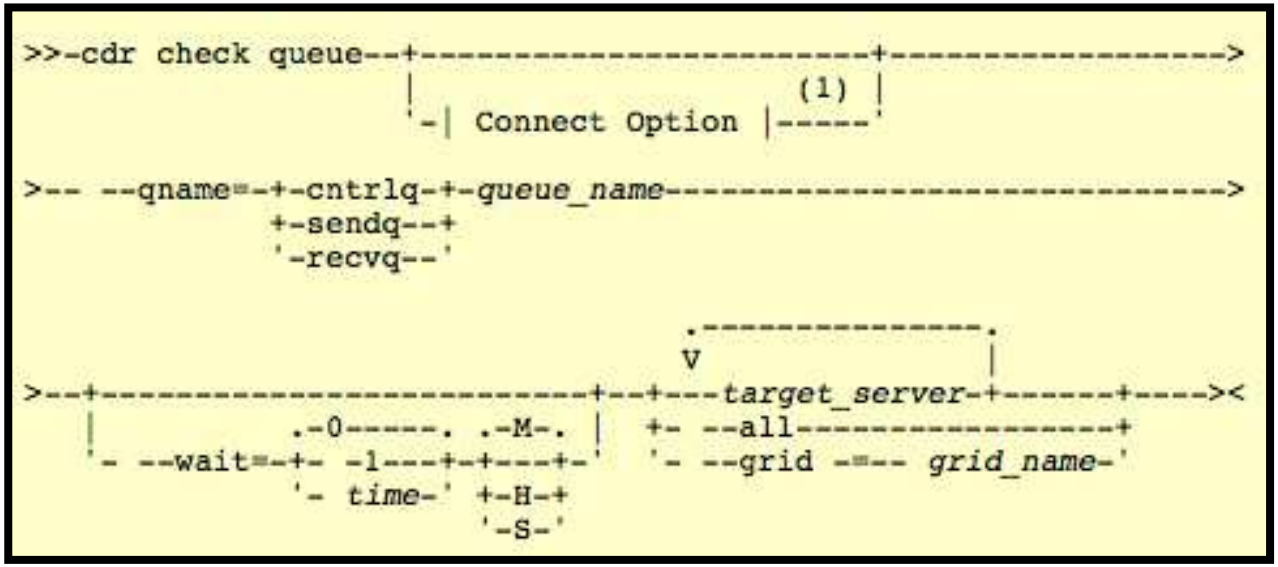
- **Supported ER queues: Control, Send, and Receive queues**
  - For **Control** and **Receive** queues
    - The command waits for all messages in the queue up to the most recently queued message is deleted from the queue.
  - For the **Send** queue
    - The command waits for all of the committed transactions that were active at the time the program was run to be queued into send queue, and these transactions are deleted from the queue

- **Wait for the queues to complete as of time now**
  - Specify a timeout period for the *cdr check queue* command. Define time (in minutes) to allow the ER queues to be drained out or drain up to a certain time.
  - If the time expires before the ER queues are complete, the command exits with error code 21.

- **You can monitor all servers in parallel unless a parent/child relationship (using root and non-root server topology) exists between servers**

Enterprise Replication Queue Monitoring, cont'd

- **Syntax: cdr check queue command**



– 'wait' values (default time value is in minutes)
  - '0' - return a result immediately, does not wait for queues to clear
  - '-1' - wait until all queue(s) are cleared before returning
  - 'N' - Positive integer value of time in seconds, minutes, hours to wait before returning a result (One of more queues could still be full at this time)

– A successful execution will return a 0 (zero) code and failed execution, (queue(s) are still full) can return one of the following codes: 5, 17, 21, 48, 62, 94, 99, 100, 196, 222

Enterprise Replication Queue Monitoring, cont'd

- **Example 1:**
  - Check the send queue on inst_1, wait up to 10 minutes to complete
    - **Result**: Queue cleared in that time frame and command returned successfully

```
cdr check queue –q sendq –w 10m g_inst_1

Checking sendq queue status for server g_inst_1 ...
sendq queue status for g_inst_1 as of Wed Jan 9 13:03:19 2013: COMPLETE
```

- **Example 2**
  - Check the receive queue on all instances, wait up to 5 seconds to complete
    - Result: Queues did NOT clear in that time frame and command returned an error

```
cdr check queue –q recvq –w 5s –a

Checking recvq queue status for server g_inst_1 ...
recvq queue status for g_inst_1 as of Wed Jan 9 13:08:19 2013: COMPLETE
Checking recvq queue status for server g_inst_2 ...
recvq queue status for g_inst_2 as of Wed Jan 9 13:08:19 2013: COMPLETE
Checking recvq queue status for server g_inst_3 ...
recvq queue status for g_inst_3 as of Wed Jan 9 13:08:19 2013: INCOMPLETE
        Operation timed out.
command failed -- Command timed out. (21)
```

Enterprise Replication and Automatic Storage Provisioning

- **11.70 Automatic Storage Provisioning feature ensures that space is available**

  Review: what does the feature do?

  – Add device, directory, etc. to the server's storage pool
  – Allocate chunks and spaces from the pool
  – Configure the minimum free space threshold
  – The server periodically monitors for low space and extend a chunk or expand a space appropriately

- **Failure to create the spaces and configure the $ONCONFIG was one of the biggest stumbling blocks for new ER/Grid administrators**

- **With Informix v12.10, it now works when defining Enterprise Replication!**

92

Enterprise Replication and Automatic Storage Provisioning

- **ER requires a certain amount of disk space to run**
  - CDR_QDATA_SBSPACE
  - CDR_DBSPACE
- **Prior to this feature –**
  - Setting the parameters was a must before executing ***cdr define server***
- **With this feature:**
  - if server has a storage pool with sufficient space and the ONCONFIG settings are unset then ***cdr define server*** will:
    - Create needed space from storage pool
    - Update ONCONFIG settings
    - Proceed to define the new ER node
- **How much space must storage pool have?**
  - 500 MB for CDR_QDATA_SBSPACE, 100 MB for CDR_DBSPACE comprised of 100 MB minimum sized chunk(s)

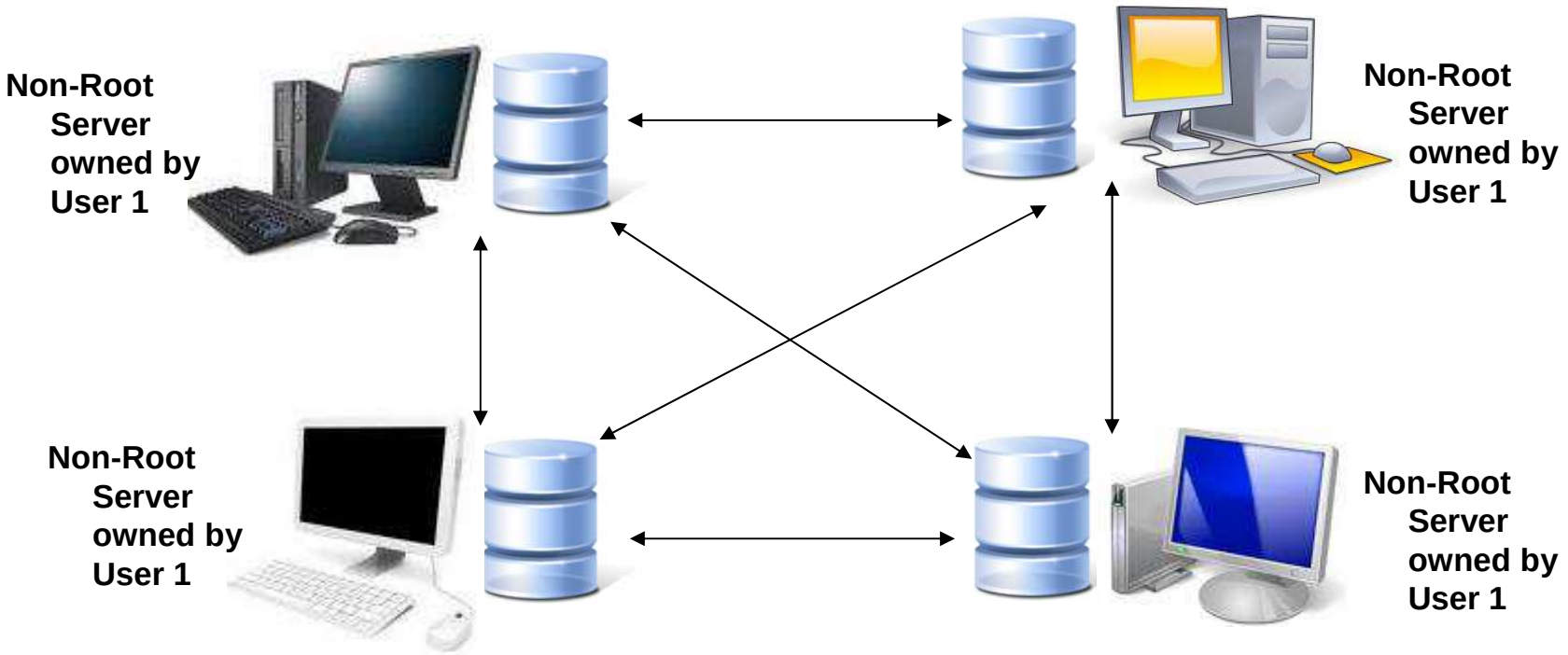
93

Data replication is supported among servers that have different owners

- **Prior to Informix v12.10, Enterprise Replication required database servers to connect as user Informix**
- **You can now replicate data among database servers that**
  - Have non-root installations
  - Do not have a user informix account
- **To configure and manage Enterprise Replication, you must have one of the following roles or privileges:**
  - Be the owner of a non-root server
  - Have the Database Server Administrator (DBSA) privilege
  - Be user informix (UNIX) or a be a member of the Informix-Admin group (Windows)
- **To initiate Data replication, you need to**
  - Create a file that lists trusted hosts for remote authentication
  - Set the REMOTE_SERVER_CFG configuration parameter to that file
  - Ensure all servers in the replication domain have the same owner

Data replication is supported among servers that have different owners

- All the non-root servers have User 1 as owner in common



Replicates are mastered by default

- **If you do not specify a master server, the master replicate is based on the first participant**
- **You must specify at least one participant when you create a master replicate**
- **A master replicate:**
  - Uses saved dictionary information on attributes of replicated columns
  - Verifies participants conformity to the specified schema
  - Disables participants who do not conform to the master definition
- **Create a classic replicate, if you do not want to verify the schema**
- **All database servers that have master replicates must be able to establish a direct connection with the master replicate database server.**

Support for light-append operations

- **Enterprise Replication now captures rows that are added through light-append operations, such as the express-load operation of the Informix High-Performance Loader (HPL)**
- **Initial synchronization feature allows you to easily bring a new table up-to-date with replication**
- **No need to suspend any servers that are replicating data while you add the new replicate and synchronize it**
- **Set up your replication environment to load the datasets:**
  - Choosing any of the load /unload utilities
  - Ensuring the use of same utility to perform load/unload operations
- **Block the replication to use load and unload tools on tables that are already being replicated**
- **Unlogged changes to a table, such as data added by a light append operation are now replicated to other tables**

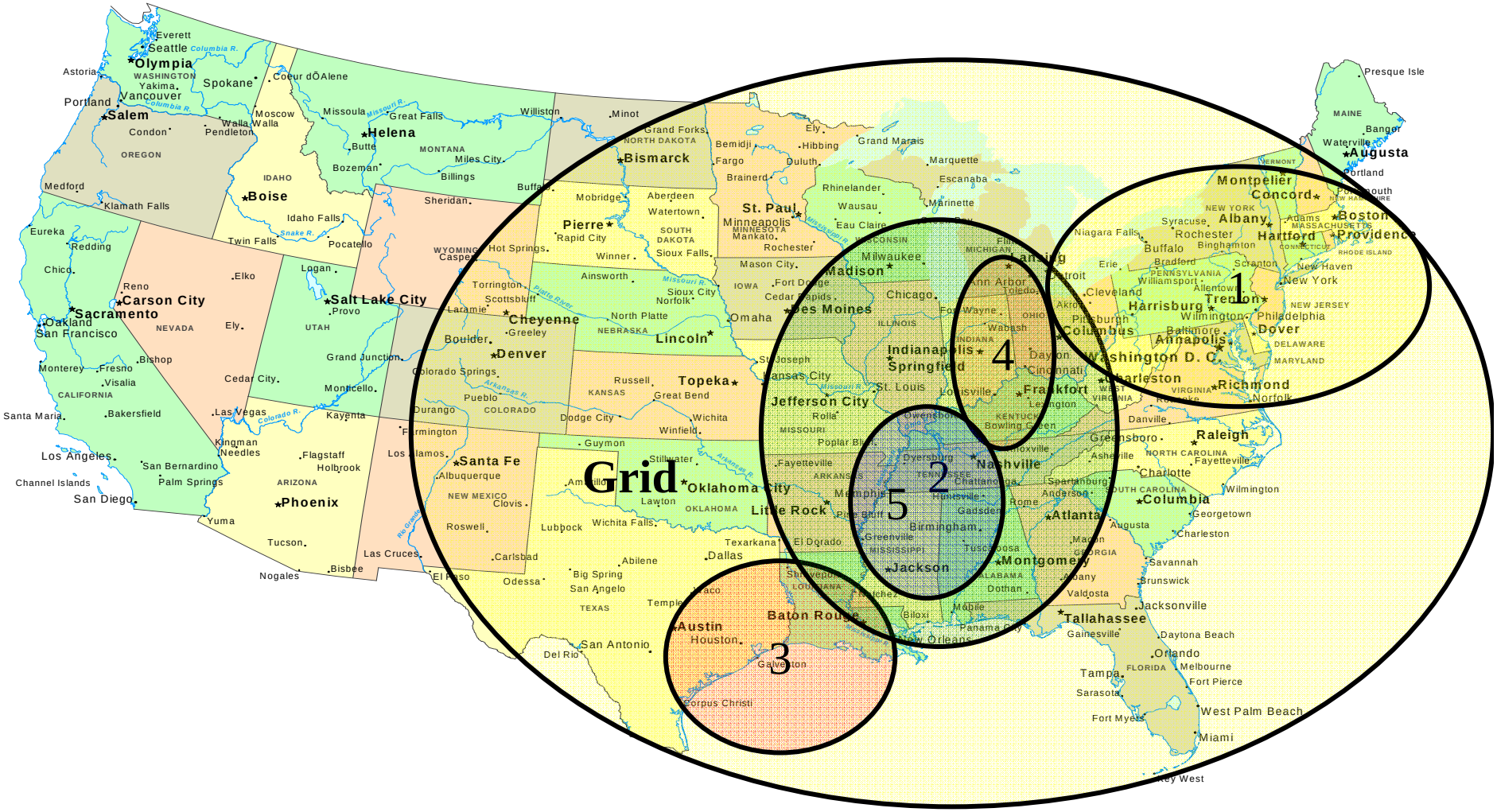
Multi-Node Grid Queries

- **Flexible Grid – Current setup**
  - Is a group of independent servers
  - Schema changes performed on one node could be chosen to be propagated to other nodes within the grid
  - Automatically replicate data between the nodes within the grid

- **Flexible Grid - Enhancements in v12.10**
  - Introduces a new concept called **Region**
  - A Region is a subset of nodes within the **Grid**
  - Based on the business requirements, you have the:
    - Flexibility to define any number of regions of choice
    - Choice on grouping of nodes against each individual region
  - Node can be made available in multiple regions
  - No restriction on the number of regions within a grid
  - Multiple Regions are allowed to overlap against each other
  - **Multi-Node Query** helps add a level of virtualization between the query and the sources of the result set.

Multi-Node Grid Queries

- **Grid and its Regions**



Multi-Node Grid Queries

- **Syntax: Define a Region**

The `cdr define region` command is used to define regions

```
cdr define region [--grid | -g ]=grid_name
   region_name [list_of_instances]
```

region_name must be unique across all clusters
list_of_instances is a whitespace separated list of server group names

- **Example: Define a Region**

```
cdr define region –g=my_grid  region_1
   g_inst_1 g_inst_2;

cdr define region --grid=my_grid region_2
   g_inst_3 g_inst_4;
```

  - Instances can be members of more than one region if needed
  - Regions can contain a subset of instances of another region

Multi-Node Grid Queries

- **You can also:**
  - **Ignore unavailable Grid nodes** - Avoid failure of a grid query if one or more nodes are unavailable by running the SET ENVIRONMENT GRID_NODE_SKIP ON statement of SQL before issuing the grid query. The identity of a skipped grid server can be returned by executing the new ifx_gridquery_skipped_nodes( ) built-in function.
  - **Defer DDL Propagation** - Defer the DDL propagation to a later time by queuing up the DDL changes, as part of the grid operation. That way, you can make all of the DDL changes on one node of the grid, test the application with the new DDL changes and then roll out the DDL changes
  - **Work with ALTER** - When a ALTER command is executed, the gridtable metadata is flagged to indicate that a alter operation is in progress. A cdr command “cdr remaster gridtable” can be used to re-verify that the alter has been successfully propagated through out the grid.

Multi-Node Grid Queries

- **Syntax – Ignore unavailable nodes**

```
set environment grid_node_skip [ default | off | on ]
```

  - Where
    `default` | `off` – the default setting. Query is aborted and an error returned
    on – rows are returned from the nodes available at the time the query was executed

- **Syntax – Defer DDL propagation**
  - The `ifx_grid_connect()` function now supports the following options

```
execute procedure ifx_grid_connect('gridname', 'tagname',
er_enabled)
```

  - Where the er_enabled flag can be
    - 0 – ER replicates for DDL operations are not created
    - 1 – ER replicates for DDL operations are created
    - 2 – ER replicates for DDL operations are not created and any ER / Grid errors are suppressed so the session may connect to the cluster
    - 3 – ER replicates for DDL operations are created and any ER / Grid errors are suppressed so the session may connect to the cluster
    - 4 – DDL operations are deferred, ER replicates are not created
    - 5 – DDL operations are deferred, ER replicates are created

Multi-Node Grid Queries

- Any SELECT statement that explicitly or implicitly includes the GRID clause is called a grid query
- The result of a grid query are qualifying rows from a logical UNION or UNION ALL of each table in the FROM clause across tables with the same names and the same schema in every grid server.
- This union can include tables across all nodes in the grid, or across a subset of those grid nodes, called a region.
- The GRID clause is not valid unless the session is connected to a database within an existing grid.

Multi-Node Grid Queries

- **SET ENVIRONMENT SELECT_GRID**
  - This statement can specify a grid or region as the default scope of subsequent grid queries that return the union of unique qualifying rows.
  - The GRID clause can omit the grid or region name for grid queries that return UNION results for the specified default nodes.
- **SET ENVIRONMENT SELECT_GRID_ALL**
  - This statement can specify a grid or region as the default scope of subsequent grid queries that return the union of all qualifying rows, including duplicates.
  - The GRID clause can omit the grid or region name for grid queries that return UNION ALL results for the specified default nodes.

- **Examples**

```
SET ENVIRONMENT SELECT_GRID 'region_03'
SELECT * FROM tab1;
SELECT * FROM tab2;

SELECT * FROM tab1 GRID 'region_03';
SELECT * FROM tab2 GRID 'region_03';
```

```
SET ENVIRONMENT SELECT_GRID_ALL 'region_04'
SELECT * FROM tab1;
SELECT * FROM tab2;

SELECT * FROM tab1 GRID ALL 'region_04';
SELECT * FROM tab2 GRID ALL 'region_04';
```

Easily propagate external files through a grid

- **11.70 Flexible Grid feature gave you the ability to replicate non-traditional things e.g. DDL, the execution of a command**
- **Question: What about replicating the scripts, UDR libraries, etc. that accompany my application?**
- **Answer:** ifx_grid_copy()**. Allows the transfer of an external file to all nodes within the grid.**
  - For example: programs/binaries, scripts, configuration files
- **Makes it easy to replicate non-database objects in conjunction with with associated DDL changes**




Easily propagate external files through a grid

- **GRIDCOPY_DIR onconfig setting**
  - the base directory *from which* and *to which* copies are performed
  - Permitted values: $INFORMIXDIR or a subdirectory thereof
  - E.g. ifx_grid_copy(‘myGrid’, ‘app1/script1.py’)
    - copies $INFORMIXDIR/app1/script1.py
    - too all other nodes in grid: <node’s GRIDCOPY_DIR>/app1 directory
    - preserving the group and owner values

- **What happens if “app1” doesn’t exist on a target? It is created**

- **What happens if a node doesn’t have group or owner? Copy fails**

106

Easily propagate external files through a grid

- **Syntax - ifx_grid_copy() function**

```
execute function ifx_grid_copy ('grid_name',
'path_and_source_file'
[,'path_and_target_file'] )
```

  - The parent directory for the source and target is assumed to be the value of GRIDCOPY_DIR
    - If a target location is not specified, the file will be copied to the same location as on the source
    - If the target location directory does not exist, it will be created by the copy operation
  - The file name can be changed as part of the copy process

- **Example – ifx_grid_copy() function**
  - If source GRIDCOPY_DIR = my_stuff and target GRIDCOPY_DIR = custom_scripts

```
execute function ifx_grid_copy ('my_test_grid',
'my_new_shell.sh')
```

  - The operation will copy $INFORMIXDIR/my_stuff/my_new_shell.sh to $INFORMIXDIR/custom_scripts/my_new_shell.sh on the target

Customer Quotes (EVP Program)

IBM Informix's robust & fast in-situ migration from older versions to latest release allows us to upgrade client's existing production databases with confidence and minimal interruption. Our ***clients can continue to enjoy the stability & performance*** *of Informix they expect, but have immediate access to the latest features of the new release.*

- Mark Rees, Chief Technology Officer, Century Software

IBM Informix v12.10
EMBED

Embeddability (SQL, SPL and OLAP)

- **Compression technology has new features**
  - Compress B-tree indexes, Partition Blobs and schedule automatic compression
- **Simplified Server Configuration**
  - Easily configure an embedded server
  - Configure the Server dynamically on the fly
- **New Storage Manager helps embed BAR solutions**
  - Easy to use Primary Storage Manager

Embeddability (SQL, SPL and OLAP)

- **Enhanced SQL/SPL, Enriched SQL/OLAP sets**
  - DDL support for time-cyclic data management
  - ANSI Joins are now way faster
  - Efficiently manage database with Rolling Windows
  - Raises the bar, eliminating limitations/restrictions
- **Boosting application compatibility**
  - Extended support for XML functionality
  - Support for Unicode 6.0

SELECT … INTO <permanent_table>

- You can now create persistent tables in Informix that can be accessed across sessions using the Result Set of a SELECT Statement.
- Combines the functionality of CREATE TABLE and INSERT into a single statement.
- Supports all the Storage options of CREATE TABLE on Informix
- Allows you to create new schema from existing tables and populate it with only a subset of data for testing purposes.
- Supported in Flexible Grid environment

SELECT … INTO <permanent_table>, cont'd

- **Functionality supported through two different syntax enhancements:**
  - `SELECT col1, col2 FROM sourcetab INTO newtab;`
  - `CREATE TABLE newtab AS SELECT col1, col2 FROM sourcetab;`
- **Display Label (or Column Alias) used in SELECT Clause would be used as Column Name of newly created table.**
- **All expressions other than simple column expressions must have a display label (or column alias)**
- **Any Informix supported SELECT syntax to create a projection list is valid in the SELECT part of the syntax.**

SELECT … INTO <permanent_table>, cont'd

- The type of the column in the new table would be the same as the type of the corresponding column in the source table.
- For a non-trivial expression used in the projection list, the column type would correspond to the return type of the expression.
- Will not inherit any constraints, primary keys, Label Based Access Control (LBAC) properties, encryption or any other special properties of the columns that are part of the projection list of the SELECT clause.
- Any constraint will need to be introduced with ALTER TABLE.
- Cannot be part of sub-query

SELECT … INTO <permanent_table>, Example

- **Specify Storage Options for a Result Table**
  - In the following example the target table “permtab” would be fragmented and split across the dbspaces “dbs1” & “dbs2” :

```
SELECT col1::FLOAT fcol1, col2  FROM tab1
INTO permtab
FRAGMENT BY EXPRESSION
       fcol1 < 300 IN dbs1,
       fcol1 >=300 IN dbs2;


CREATE TABLE permtab (fcol1, col2)
FRAGMENT BY EXPRESSION
       fcol1 < 300 IN dbs1,
       fcol1 >=300 IN dbs2
       AS
SELECT col1::FLOAT fcol1, col2
FROM tab1;
```

CASE in Stored Procedure Language (SPL)

- You can now take a different branch of execution based on the value of an expression or value of a SPL variable.
- Similar to IF-THEN-ELIF-ELSE-ENDIF statement in SPL routines.
- You can use the CASE statement to create a set of conditional branches within an SPL routine.
- The WHEN and the ELSE clauses are optional, but you must include at least one of them.
- Not to be confused with the CASE Expressions of SQL.
- The statement block that follows the THEN or ELSE keywords can include any SQL statement or SPL statement that is valid in a statement block of an SPL routine.
- Similar to the CASE statement in XPS.

CASE in Stored Procedure Language (SPL)

- **SPL with a CASE statement**
  - An Example

```
CREATE PROCEDURE case_proc( )
RETURNING CHAR(1);
DEFINE grade CHAR(1);
LET grade = 'D';
CASE grade
  WHEN 'A' THEN LET grade = 'a';
  WHEN 'B' THEN LET grade = 'b';
  WHEN 'C' THEN LET grade = 'c';
  WHEN NULL THEN LET grade = 'z';
  ELSE LET grade = 'd';
END CASE;
RETURN grade;
END PROCEDURE;
```

Dynamic Configuration Tuning : External Changes

- More parameters can be modified on the fly
- Parameter behavior is more intuitive and consistent
- More information about parameters is available through new `onstat -g cfg` command
- Configuration snapshot can be exported to a file
- Configuration files can be imported to a running server
- New config-related SQL admin API commands
- Any environment variable can be embedded in any parameter value
- Deprecated Parameters Removed

Dynamic Configuration Tuning : Internal Changes

- **Configuration parameter code and internal structures completely redesigned with dynamic tuning in mind**
- **Easier now for Informix engineers to make new and existing parameters dynamically tunable**
- **Near-Term Goal:**
  - All new parameters should be dynamically tunable
  - Add as many existing parameters to the list as possible
- **Long-Term Goal:**
  - Built-in analytics + DCT --> Self-Tuning Server

Auto Compression

- **Creates a compression dictionary for the data rows of a table automatically when it has enough number of rows.**
- **In earlier release, compression could occur only after data was loaded.**
- **Benefits of automatic compression:**
  - Informix compresses data rows as the data is loaded.
  - Informix sets compression as a property of the table, so any new fragments added also get compressed automatically.
  - You use current SQL admin API commands to set a table or fragment for auto compression.
  - You have an SQL interface to create a compressed table.

Auto Compression (cont'd)

- A minimum of 2000 rows are needed for compression dictionary to be created.
- You can start compression with the same SQL admin API commands that you currently use.
- You can also compress a table, using SQL syntax when you create the table.
- Every compressed partition blob column has its own compression dictionary.
- Each compressed (non-fragmented) table or table fragment has its own compression dictionary for in-row data.
- A dictionary consumes ~75K – 100K per fragment
- Thus compressing tiny tables is not recommended

Auto Compression (cont'd)

- **All dictionaries for tables/fragments in a given dbspace are stored in a special hidden dictionary table in that dbspace**
- **The sysmaster database shows information on all dictionary tables:**
  - syscompdicts_full table: Includes binary dictionary; access restricted to user "informix"
  - syscompdicts view: Globally accessible; omits binary dictionary for security
- **When a table load uses light append, a dictionary will be automatically created. Also, the rows that were loaded before the dictionary was created will be compressed.**
- **During a normal insert, the new rows inserted will be compressed. Data rows already in the table (before the compress) will not be compressed.**

Auto Compression (cont'd)

- **Using Admin API**
  - Execute function task("**table compress**", "my_table", "my_database");
  - Execute function task("**fragment compress**", "my_fragid");
  - Even if there are not enough rows to sample and create a compression dictionary, the SQL admin commands will succeed indicating "Auto compression is set".
- **Using SQL**
  - Create table my_table (my_col ….) **compressed**;
  - After the table is created, as data is loaded into the table, a compression dictionary will be created when 2000 rows are inserted.

Note: Auto compression is not supported for default indexes and blobs.

Index Compression

- **Compress and repack an existing index.**
- **Create a new index that will be built as a compressed index**
- **Only detached B-tree indexes can be compressed**
- **Benefits of Index compression:**
  - Informix saves disk space by keeping indexes compressed.
  - Informix provides I/O savings by reading in compressed index pages to the buffer pool.
- **Only the key values at the leaf level of the index are compressed.**
- **A minimum of 2000 unique keys are needed for a compression dictionary to be created.**
- **The Rowids following the key are not compressed.**

Index Compression (cont'd)

- **You can start compression with the same SQL admin API commands that you currently use.**
- **You can also compress an index, using SQL syntax when you create the index.**
- **Using Admin API**

```
Execute function task("index compress", "my_idx",
  "my_database");
```

- **Using SQL**

```
Create index my_idx on my_tab(my_col ….) in dbs_idx
  compressed
```

- **To compress an existing index**

```
Execute function task("index compress", "j", "testdb");
```

- **To repack and shrink an existing index**

```
Execute function task("index repack shrink", "j",
  "testdb");
```

BLOB Compression

- In previous version, all the data that resides within a row was compressed. All the data stored outside the row was left uncompressed.
- With this feature we are enabling the compression of data stored in *Partition Blobs*.
- *Partition Blobs* are simple large objects (mostly text and byte data types) in which the data is stored outside the row, but in same partition of the same dbspace as the row data.
- After the blob data is repacked, there will be lot more FREE pages towards the tail of the partition. These pages can be returned back to the dbspace.

BLOB Compression (cont'd)

- **HDR - Tables/blobs will be compressed on secondary server only if they are compressed on primary server**
- **ER - Compression status of tables/blobs is independent between source and target, as specified by the user**
- **CDC - Compression of targets is a function of what the target database supports and what the user specified.**
- **To compress just the partition blobs of the table**

```
execute function task(
"table compress blobs",
"table_name",
"database_name",
"owner_name"
);
```

BLOB Compression (cont'd)

- **Data that Still Cannot Be Compressed**
  - Data stored in Blobspace Blobs
  - System catalog tables
  - Temp tables
  - Partition tables
  - Dictionary tables
  - Tables in the following databases:
    - *sysuser*
    - *sysmaster*
    - *sysutils*
    - *syscdr*
    - *syscdcv1*
  - TimeSeries tables

SQL Statement Size >64K

- The SQL Statement size of 64K characters is removed from the latest version of Informix.
- The new size is now limited by the amount of memory available to hold large statements.
- This would avoid the need to write complicated logic to break up very large SQL statements to multiple statements to get around the 64K limit
- This simplifies handling of Informix as repository for large automated application like for ERP.

Rolling Window Tables

- Extension of interval fragmentation strategies (called range fragmentation or date-range fragmentation), which create new fragments when they are required.
- Control the size of tables that have an interval table fragmentation strategy by automatically removing the oldest fragments when a table contains more than a maximum number of fragments or is greater than a maximum total size.
- Embedded applications have a need to manage limited amount of space automatically
- OEM's have in the past written thousands of lines of SPL to limit the amount of space taken by tables

Rolling Window Tables (cont'd)

- Offering the ability to control table space usage declaratively simplifies applications.
- Tables with a primary key having referential constraints to it (primary key with no references is fine)
- Only indices following the same fragmentation strategy as the table are allowed (to allow real time fragment detach)
- A new UDR (syspurge()) used to scan the system catalogs of the database to find rolling window tables that have exceeded limits
- Syspurge() returns the number of fragments dropped / discarded

Rolling Window Tables

Syntax

- **Interval fragmentation (in both CREATE TABLE and ALTER FRAGMENT ...INIT) would support following syntax :**

```
    FRAGMENT BY RANGE (<column list>)
    INTERVAL (<value>)
    [ [ROLLING(<integer value> FRAGMENTS)]
    [LIMIT TO <integer value> <SIZEUNIT>]
    [DETACH|DISCARD]]
    STORE IN (<dbspace list> |
<function_to_return_dbspacename()>) ;

 SIZEUNIT: [K | KB | KiB | M | MB | MiB | G | GB | GiB
|T | TB | TiB ]
```

Rolling Window Tables (cont'd)

- **Limit Thresholds:**
  - Maximum number of fragments [ROLLING … FRAGMENTS]
    - When fragments exceed the set value (that is when a new one is created), the one holding the lowest (oldest) set of values will be detached or discarded.
  - Maximum total size [LIMIT TO …]
    - When table exceeds the size limit, fragments holding the lowest (oldest) value will be detached/discarded until space used is below the limit.

- **Control the policy for old fragments**
  - DISCARD - eliminate the fragment for good
  - DETACH - preserve the data by detaching the fragment in a new table

Rolling Window Tables

Other Syntax extentions :

- **ALTER FRAGMENT... MODIFY INTERVAL augmented with :**

```
 [ROLLING(<integer value> FRAGMENTS)]
[LIMIT TO <integer value> <SIZEUNIT>]
[DETACH|DISCARD]
```

- **ALTER FRAGMENT … MODIFY DROP ALL ROLLING**
  - **Removes the rolling window policy altogether**
- **ALTER FRAGMENT … MODIFY INTERVAL DISABLE**
  - Disables rolling window policies without dropping them
- **ALTER FRAGMENT … MODIFY INTERVAL ENABLE**
  - Reinstates the current rolling window policy, if any is defined

Rolling Window Tables

Syntax Example :

```
create table orders
        (order_num serial(1001),
        order_date date,
        customer_num integer not null, ... )
   partition by range(order_date) interval(1 units month)
   rolling (6 fragments) limit to 20 gb detach
   store in (mydbs, mydbs2)
   partition prv_partition values < date('01-01-2010') in mydbs;
```

Rolling Window Tables (cont'd)

Restrictions

- **Cannot be a table with a primary key having referential constraints to it (primary key with no references is fine)**
- **Only indices following the same fragmentation strategy as the table are allowed (to allow real time fragment detach)**

Set Operations (Intersect and Minus)

- **Extension to the existing UNION/UNION ALL SET**
- **Intersect returns distinct rows which are returned by both queries.**
  - SELECT col11 from tab1 **intersect** SELECT col12 from tab2;
- **Minus/Except will return distinct rows from the left query which are not found in the right side query**
  - SELECT col11 from tab1 **minus** SELECT col12 from tab2;
- **NULL friendly, means when comparing NULL to NULL they are considered equal.**

Set Operations (Intersect and Minus) (cont'd)

- Like in the case of UNION/UNION ALL SET, both query block should have exact same number of columns and comparable data types
- Projection should not have BYTE or TEXT
- Order By should be at the end.
- Precedence would be from left to right, unless they are grouped using parentheses.

New Mathematical Functions

- New mathematical functions have been added that are designed to be fully compliant with analogous math functions of the competition.
- Can be used where any executable function is currently used provided the input is a valid value in the set of expected numeric inputs.

New Mathematical Functions (cont'd)

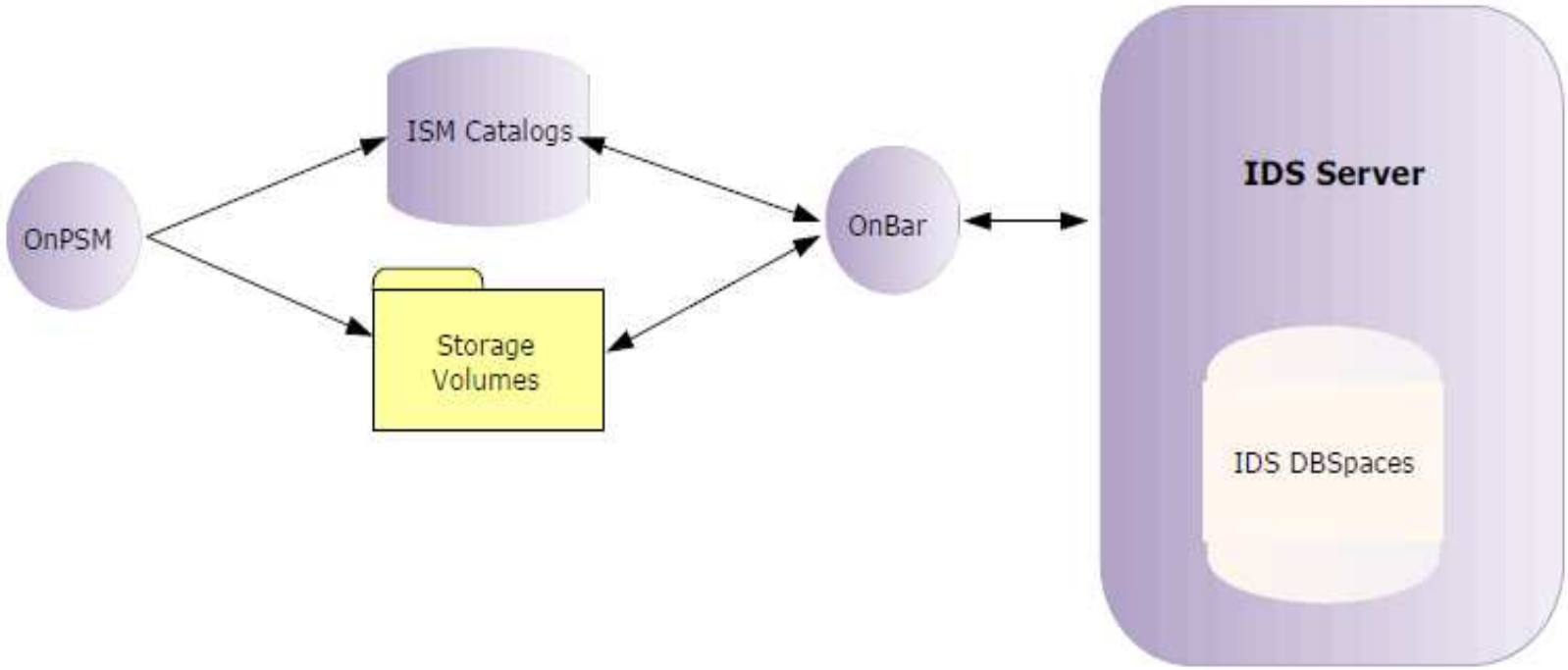
- **SIGN(num) – Returns the sign of the input numeric value (returns integer): -1 for n < 0, 0 for n = 0, and 1 for n > 0**
  - COSH(num) – Returns hyperbolic cosine (returns float)
  - SINH(num) – Returns hyperbolic sine (returns float)
  - TANH(num) – Returns hyperbolic tangent (returns float
  - ACOSH(num) – Returns arc hyperbolic cosine (returns float)
  - ASINH(num) – Returns arc hyperbolic sine (returns float)
  - ATANH(num) – Returns arc hyperbolic tangent (returns float)
  - LN(num) – Alias for existing LOGN(num) function (returns float)

What is PSM?

- PSM is the Informix Primary Storage Manager, available with the latest release of Informix Database Server
- Was born originally in XPS 8.11 as a small storage manager to be used with OnBar to reduce the impact of customer having to pay for additional software
- PSM is built around the concepts of ease of use, maintenance, allowing customers to perform faster backups and restores
- The storage manager and all of its actions are controlled by a single binary called “onpsm”
- It has a higher level of integration with OnBar so that the DBA has to work less to make them work together

Primary Storage Manager (PSM)

- New storage manager to perform the Backup and Restore operations
- Simplified administration, ease of use and maintenance
- Quickly and Easily embeds BAR solutions



Primary Storage Manager (PSM)

- **PSM Advantages**
  - No need to modify or create the $INFORMIXDIR/etc/sm_versions file.
  - In the simplest scenario it only requires one configuration parameter to point to the backup/restore device (PSM_DEVICE).
  - No need to perform any commands to make it work (Besides running ON-Bar).
  - Complete parallelism (At the dbspace level)
  - Buffer transfer size between Informix, PSM and ON-Bar is not limited to 64 KB anymore like with most Storage Mangers (Including ISM)

- **PSM Components**
  - XBSA library: The XBSA shared library is loaded by On-Bar on demand during the backup or restore operation.
  - The onpsm utlity: Front end tool that allows us to manage the aspects of the storage manager, ie report objects, create/delete devices, pools, etc .
  - The catalog: The catalog is a series of C-ISAM tables, stored under $INFORMIXDIR/etc/psm. They manage information related to the objects that PSM backs up, information on pools, devices, volumes, etc.

Embeddability Enhancements

- Enterprise class embed database gets even better!

Informix 12.10

Enterprise Class Embed Database

Dynamic configuration

Dynamically configurable parameters
Switch auto-tuning On/Off EZStart

Smarter storage

Rolling Window Tables
Auto compress while creating table, index
Compression of Partition Blob data
New Storage Manager

Performance

Improved I/O due to compression
Logging/Scaling improvements
OLAP functions for operational analytics
Overall Engine Performance
Start server without environment

Application portability

Enhanced SQL/SPL,SQL/OLAP sets
DDL support for time-cyclic data management
ANSI Joins are now way faster
Efficiently manage database with Rolling Windows
Enablement Tooling

Primary Storage Manager

Simpler to configure
Efficient backup/restore
Unlimited parallelism

Ease of Use and Administration

Mobile OAT
Auto configuration
Automatic backup
Automatic compression
Dynamic tuning

Customer Quotes (EVP Program)

"My Informix DBA team says the ***Informix Primary Storage Manager (PSM)*** *is super simple to setup and use and that* ***onsmsync*** *is really cool as it allows you to manage your saved archives and logical logs in an extremely easy manner.* ***Both of those items saves us cycles when managing our Informix environment.****"*

- Tony Salerno, Database Practice Manager & Virtual-DBA Service Manager, Xtivia Inc.

Users will be able to experience the much-improved management and usability in the new version of Informix. Kind of ***configuration parameter that can be set dynamically*** *increase is very encouraging. In addition,* ***the Primary Storage Manager*** *library setting, policy definitions, and point-in-time recovery process* ***is very simple,*** *so you will be able to* ***apply directly to the production system.*** *Features such as the* ***enhanced OLAP functions*** *and* ***SELECT INTO*** *clause,* ***simplify the programming procedure****.*

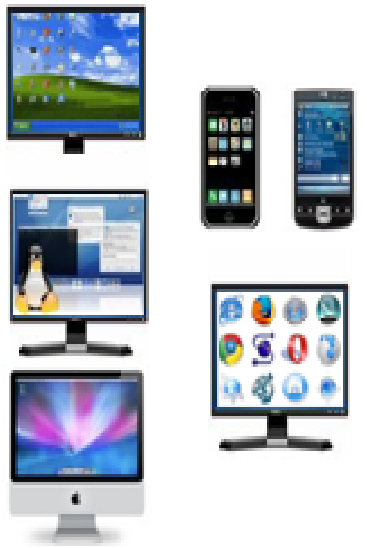
- SangGyu, Jeong, Assistant Manager-IT Service Dept, INFRASOFT CO., LTD

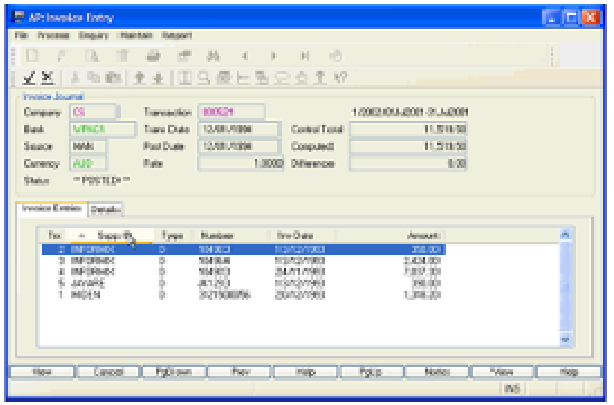
IBM Informix v12.10
EASE OF USE AND ADMINISTRATION

Ease of Use and Administration

- **OAT as an integrated administration interface for all Informix operations**
  - Enhancements to plug-ins help achieve deeper integration across features
  - OAT has a all new Welcome page to greet users and Multi-server Dashboard helps view status summary for a group of servers

- **Information at finger tips**
  - App available for Smart phones

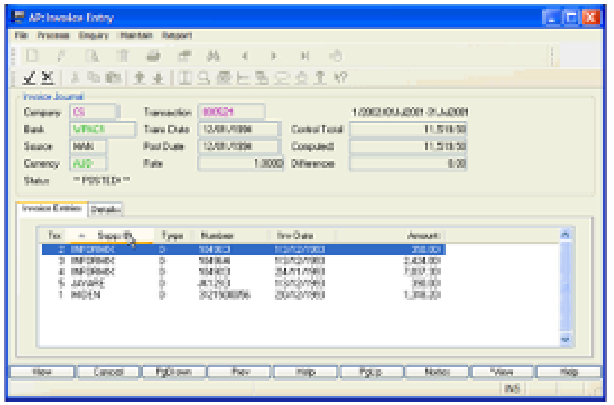




Ease of Use and Administration

- **Eased administration efforts**
  - Automatic backup using PSM
  - Administer OAT as users other than Informix
  - Achieve Storage Optimization
  - Manage SQL Rolling Window
  - Accommodates the manageability functionalities of ISAO tool

- **Improvements to Genero v2.41**
  - Has new templates for BAM
  - Enhancements done to HTML5 theme



Installation

Overview

- On Linux, the Apache and PHP software bundled with OAT have been upgraded to the latest versions.
  - These upgrades provide increased stability and enhanced security to your OAT web server environment

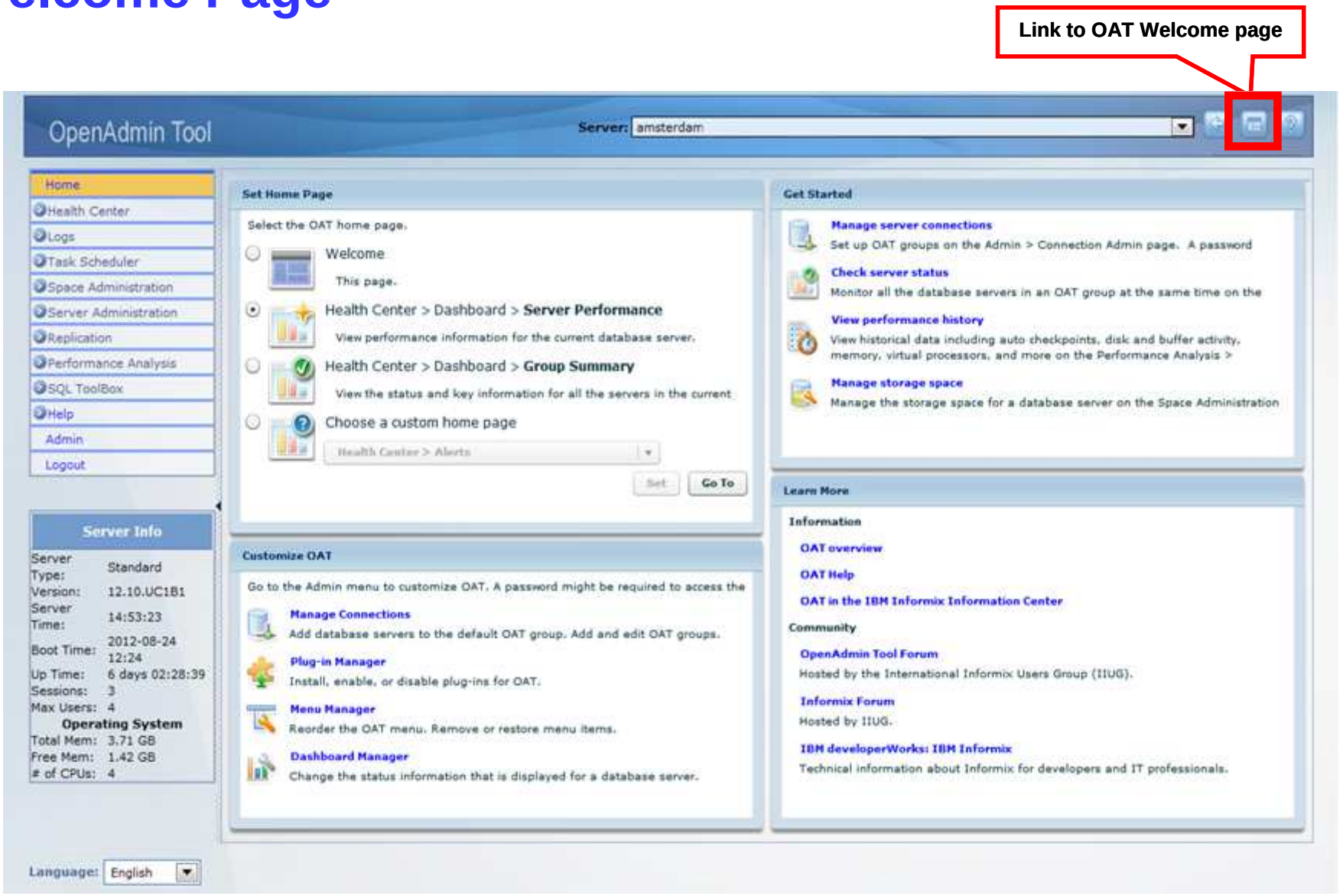
Technical Details

- Product versions shipped with OAT on Linux:
  - Apache 2.4.2
  - PHP 5.4.4
  - PDO_INFORMIX 1.2.7
- Because of the upgrades, prerequisite libraries required on Linux have changed.
  - For details, see the OAT Release Notes (or the notes section of this slide)
- The Linux upgrades have been tested and certified on the following platforms:
  - RHEL 5 x86, RHEL 5 x86_64, RHEL 6 x86, RHEL 6 x86_64, SUSE 11 x86, SUSE 11 x86_64

Welcome Page

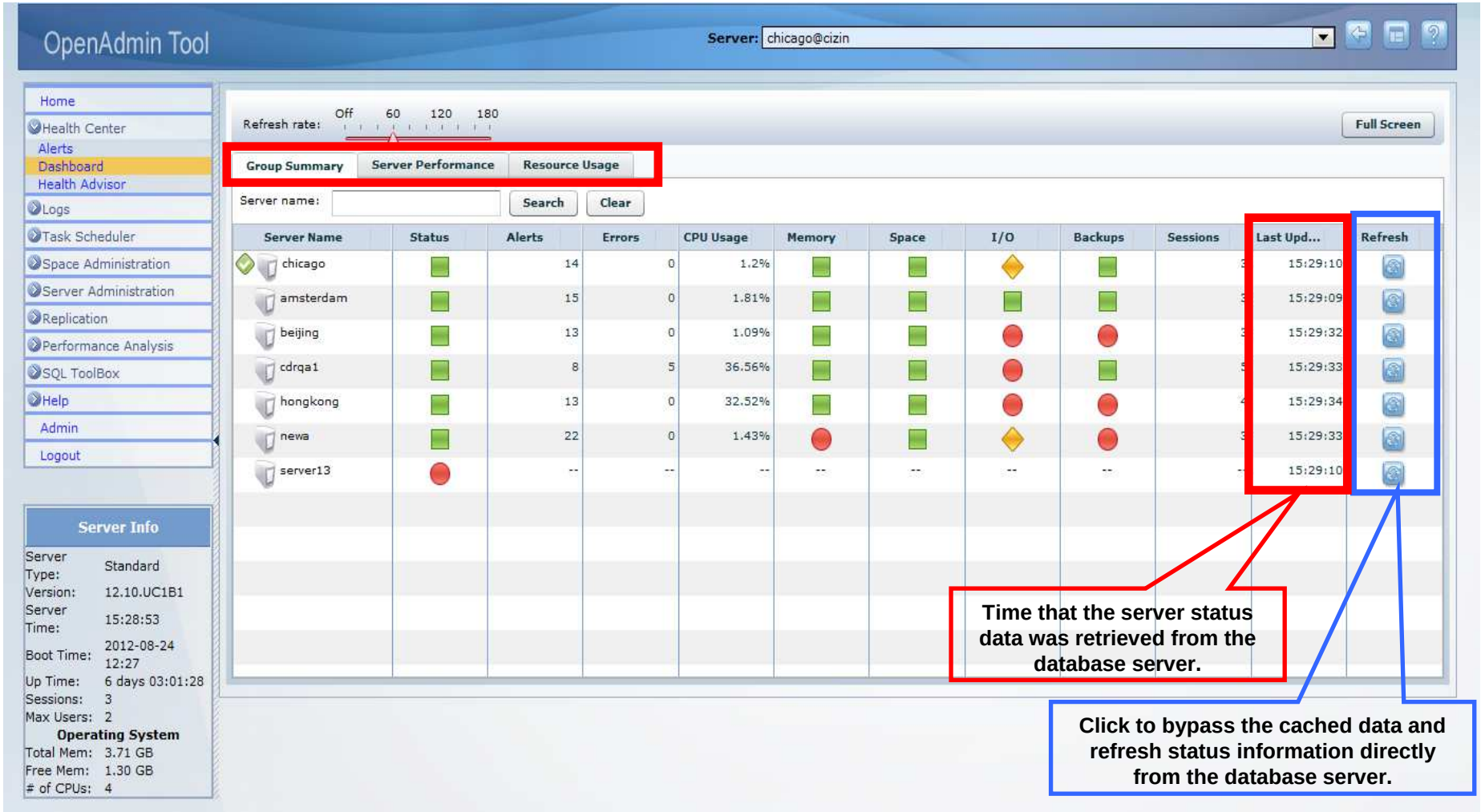
- **Has enhanced Look and Feel interface**
- **Provides a better first-time experience for new OAT users**
- **Enables user to set a Home page of choice**
  - Welcome page
  - Dashboard > Group Summary
  - Dashboard > Server Performance
  - Or any OAT page of you choice
- **Provides resources and links for**
  - Getting Started
  - Customizing OAT
    - Managing Menu, Plug-In's and Connections
    - Including customizing your OAT home page
  - And learning more about OAT

Welcome Page



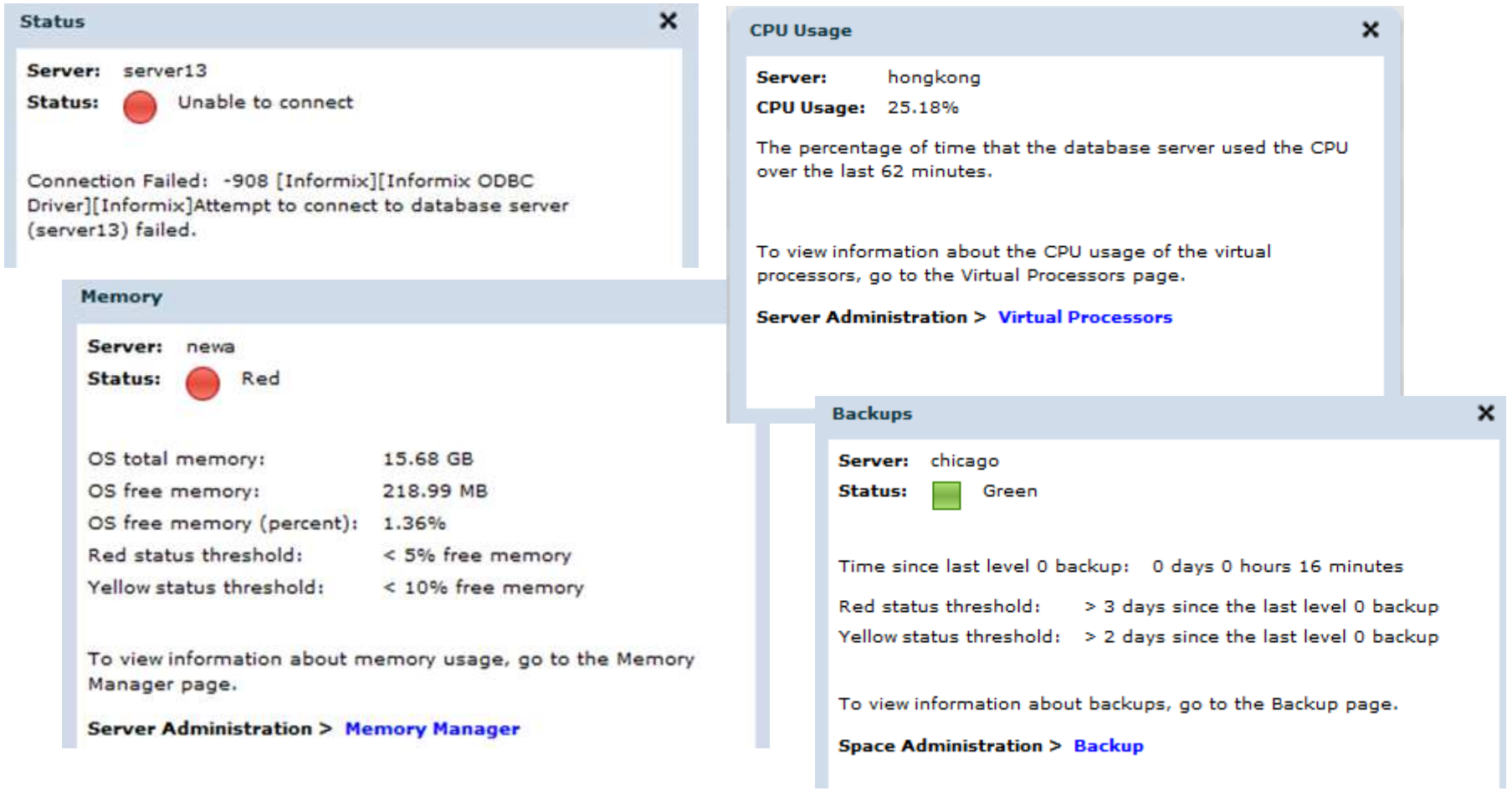
Dashboard > Group Summary

- **Click on any cell on the Group Summary page for a pop-up with more details and links to the related OAT pages**

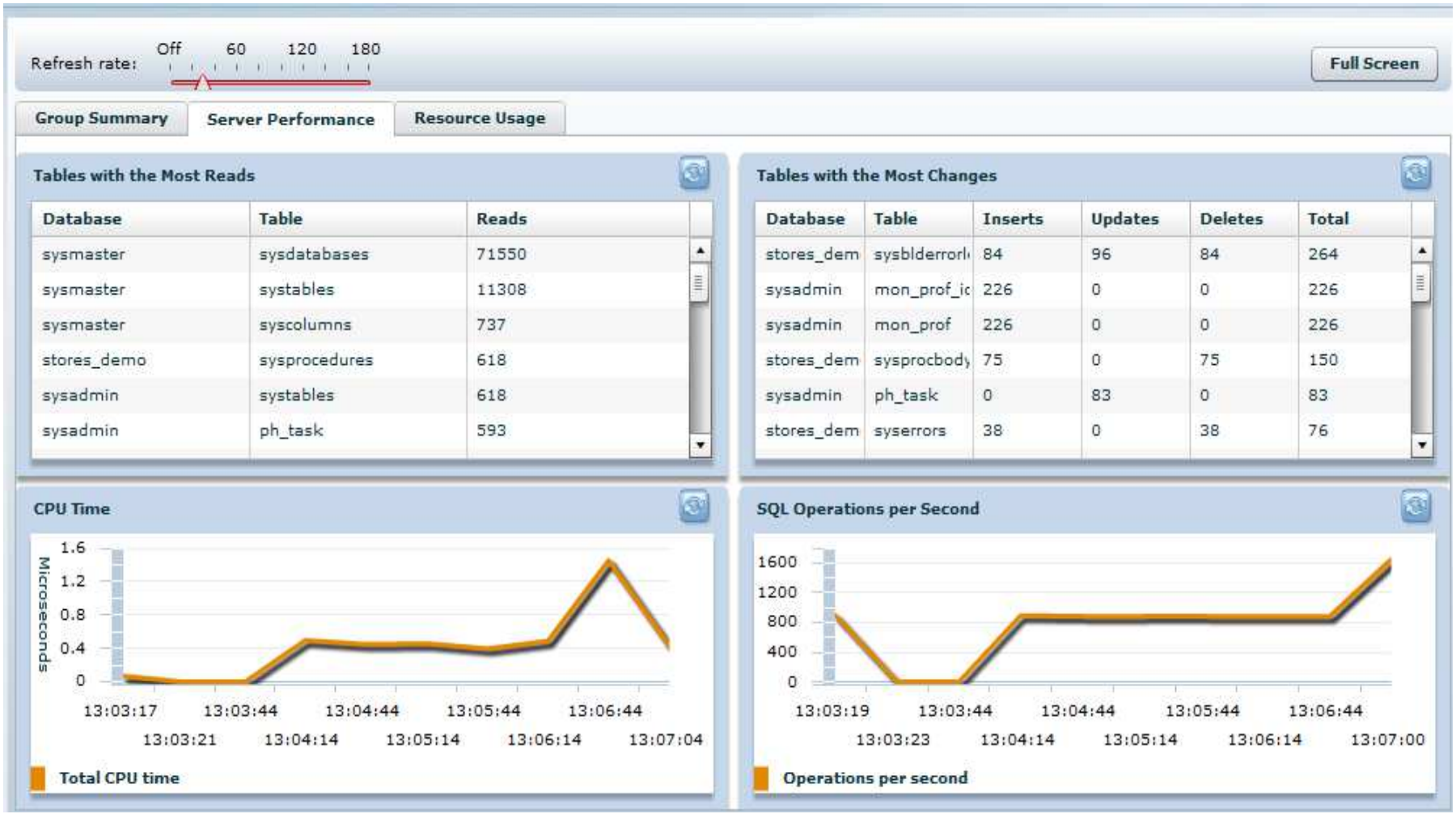


Dashboard > Group Summary

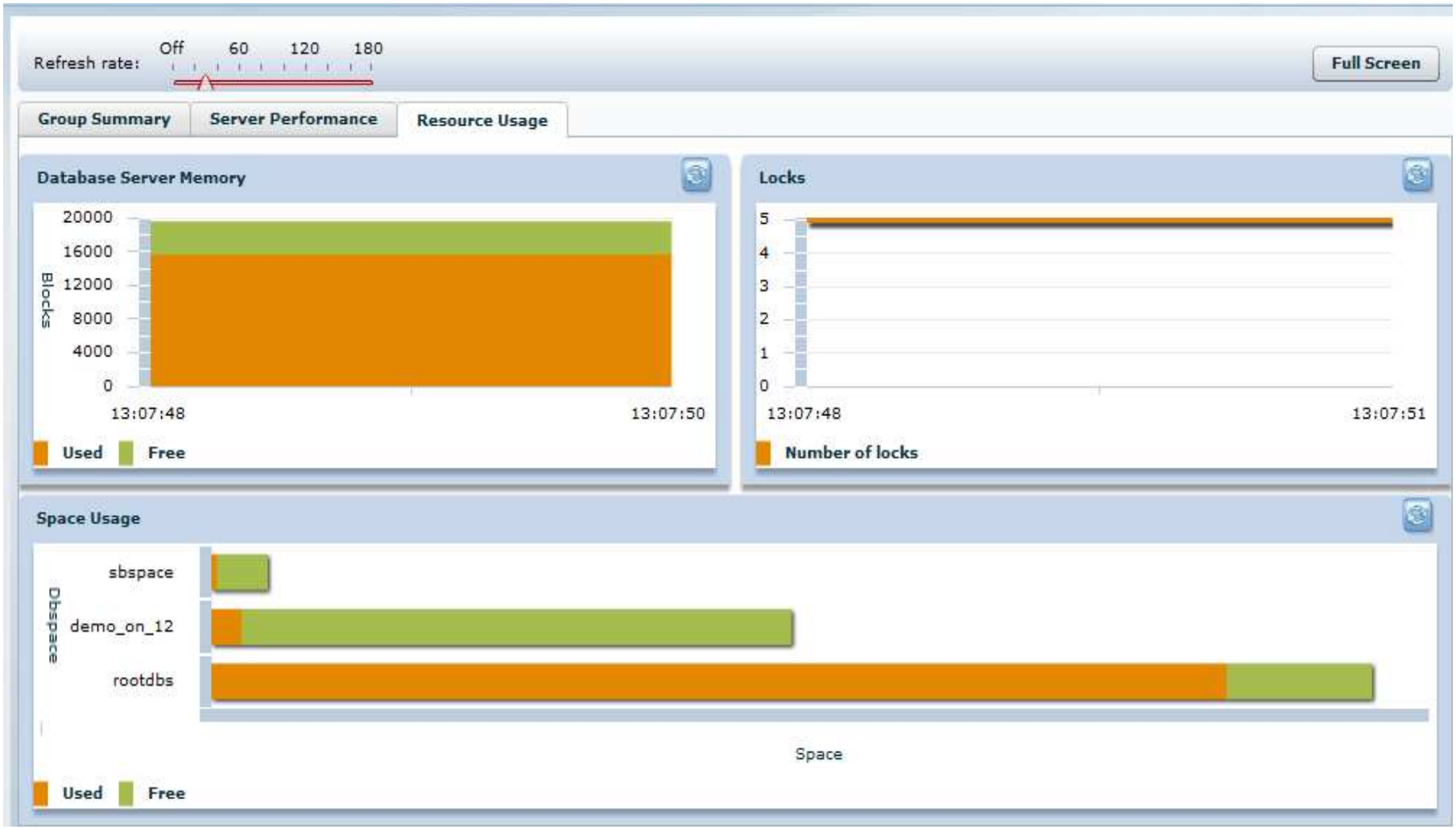
- **Click on any cell on the Group Summary page for a pop-up with more details and links to the related OAT pages.**



Dashboard > Server Performance

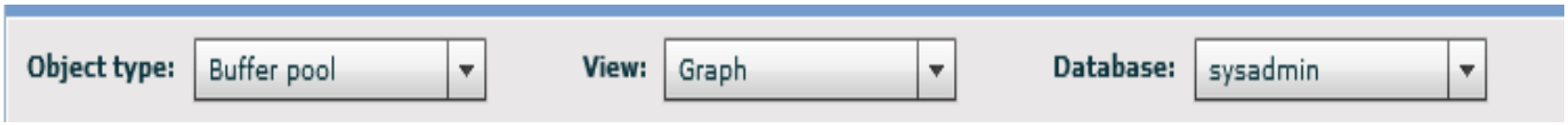


Dashboard > Resource Usage

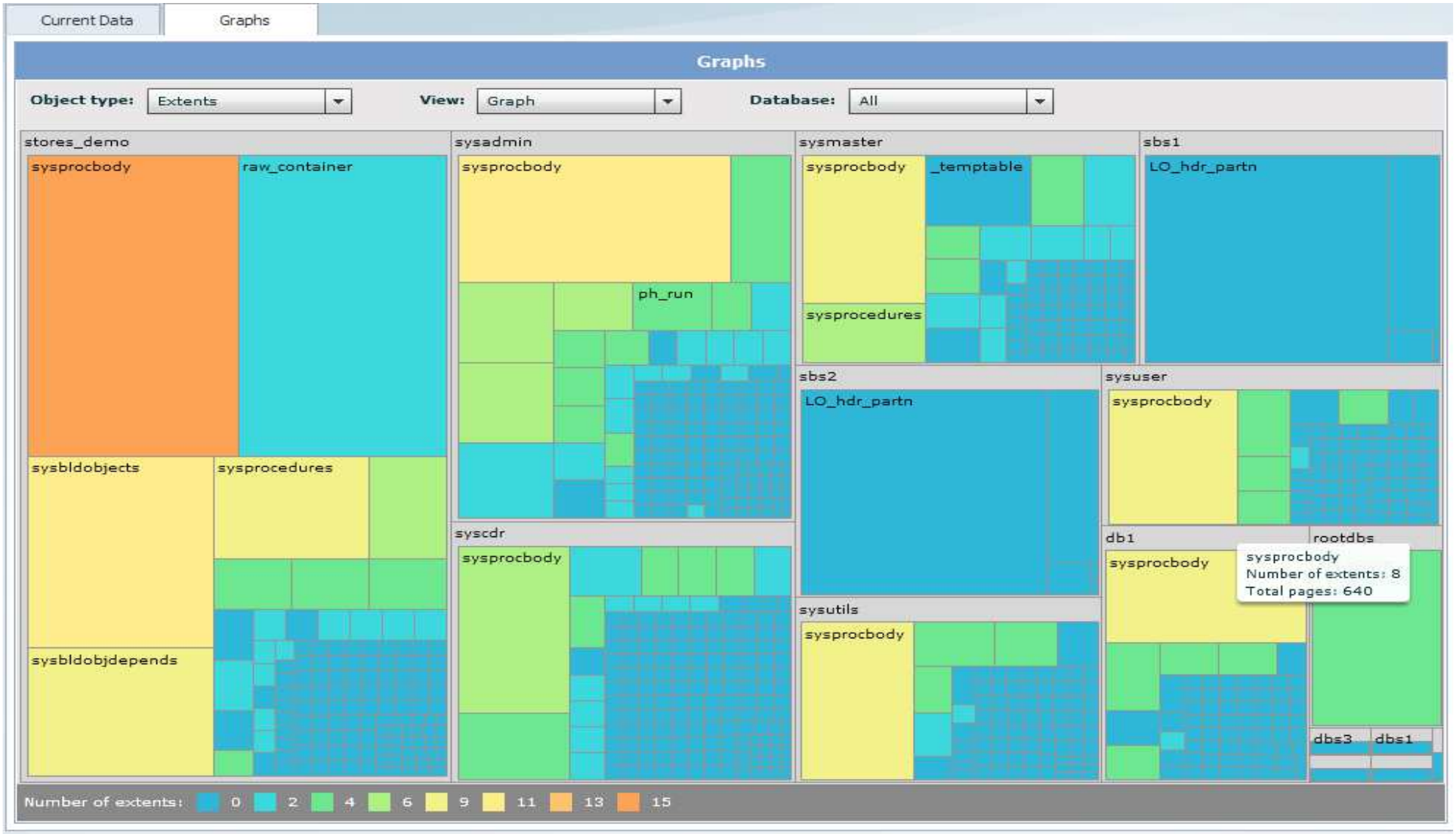


OAT Heat Map: System Reports > Graphs

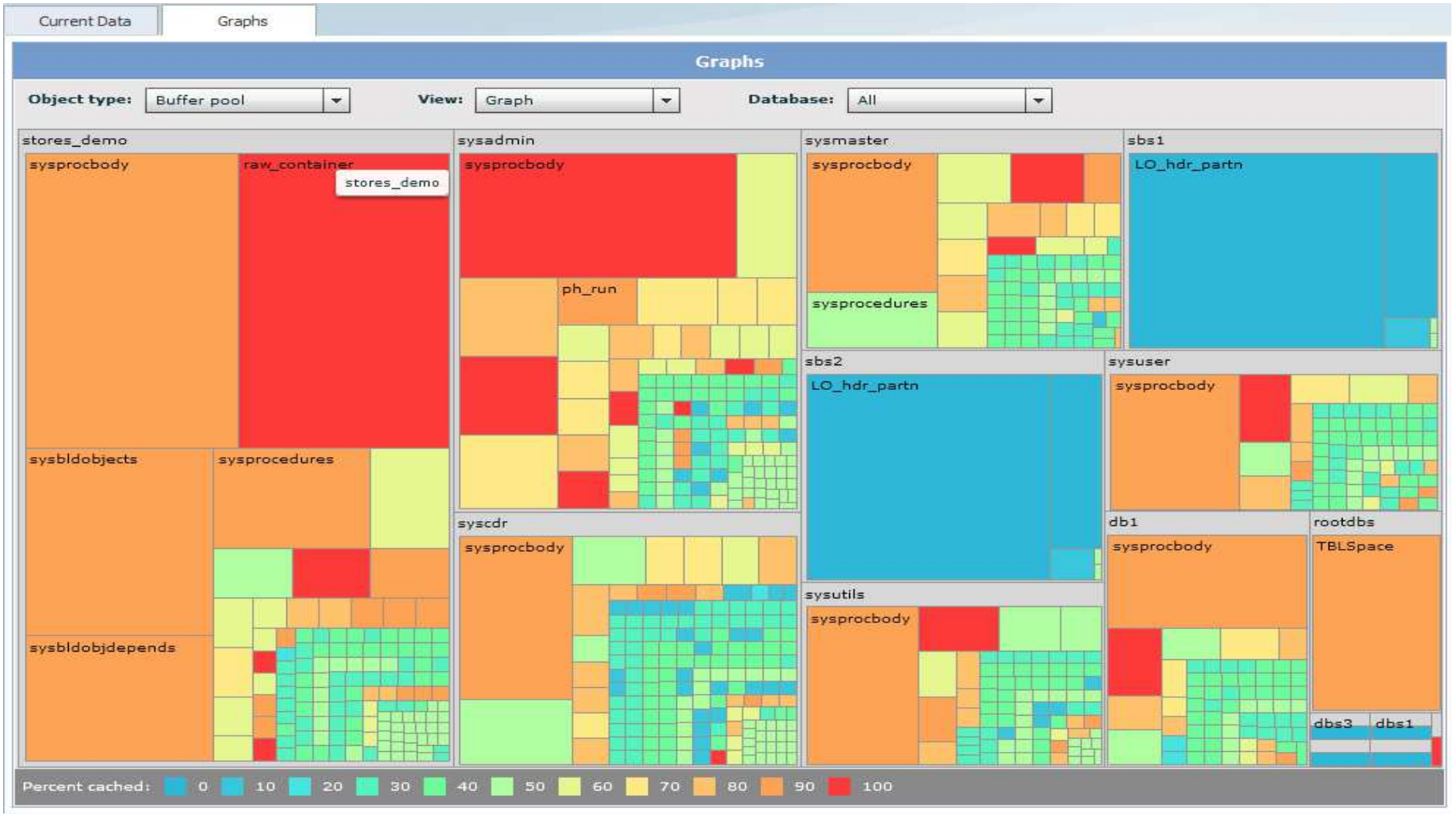
- **OAT Heat Map enables you to:**
  - Provide an easy-to-read graphical format display of server usage
  - Absorb vital usage information of the server
  - Help tune database server
- **What you can do is:**
  - View buffer-pool and extent usage in a graphical display
  - Identify the number of extents in databases or the percentage of cached pages in the buffer pools for databases
- **To generate one, you need to:**
  - Navigate through **Performance Analysis** > **System Reports** > **Graphs**
  - Use the drop-down controls at the top of the page to choose the type of data to graph (extents or buffer pool) and filter by a particular database.
- **The graph is also called a Heat map or a Tree map.**



OAT Heat Map: System Reports > Graphs > Extents



OAT Heat Map: System Reports > Graphs > Buffer Pool

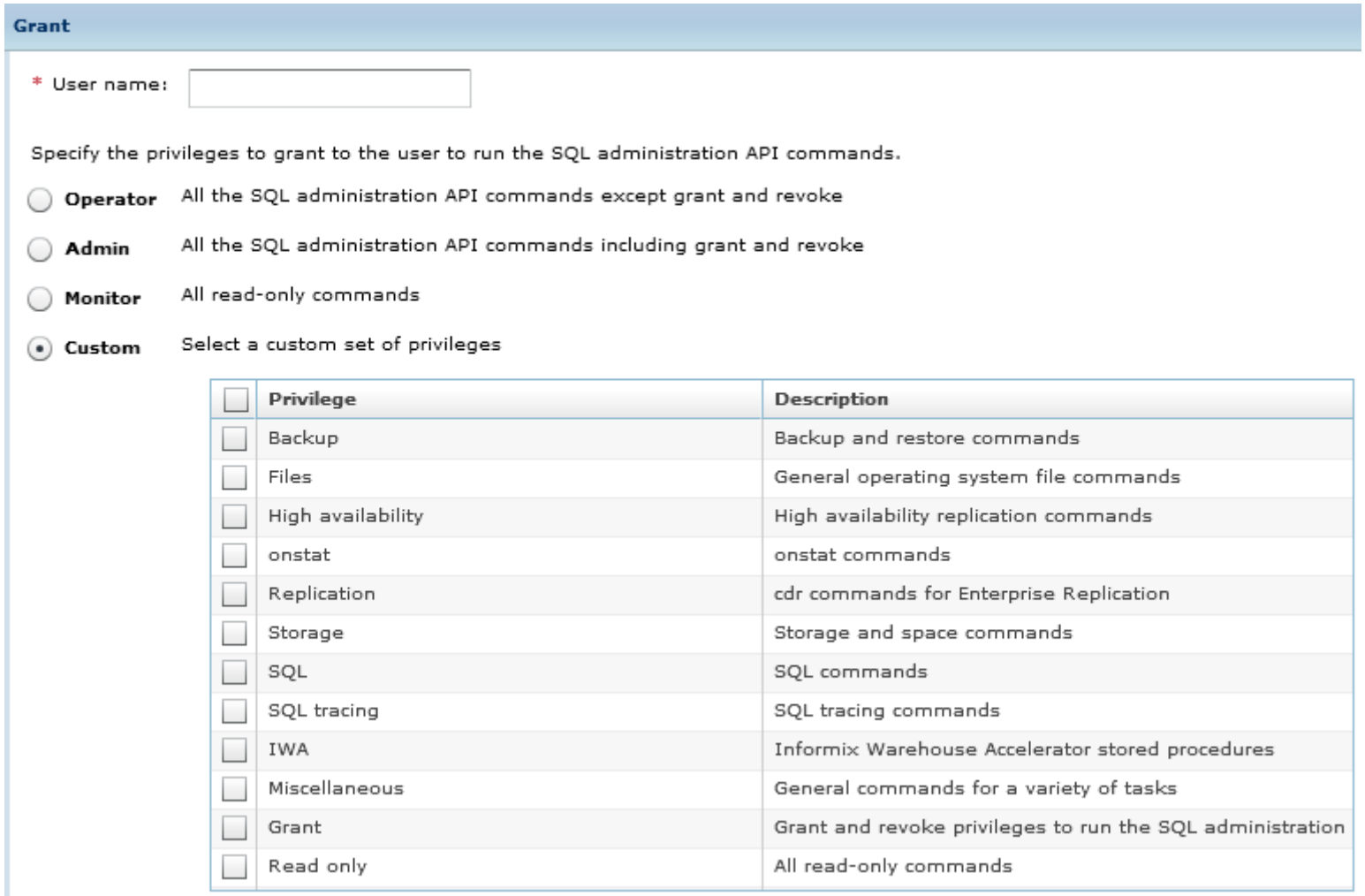


Log in to OAT with your own user ID

- **Log in to OAT with your user name**
  - You are no longer restricted to using OAT as user informix or a DBSA
  - Individual users can now be granted privileges to administer database servers by running SQL administration API commands. Users who are granted these privileges can log in to OAT with their user name.
  - Security in enhanced when OAT users can log in as themselves instead of all users sharing the informix login and password.
- **How to use it:**
  - User informix logs in to OAT and grants SQL Admin API privileges to various users
  - Those users can then login to OAT using their own user ID

Log in to OAT with your own user ID

- **Granting SQL Admin API privileges**



Storage Optimization – Automatic Compression

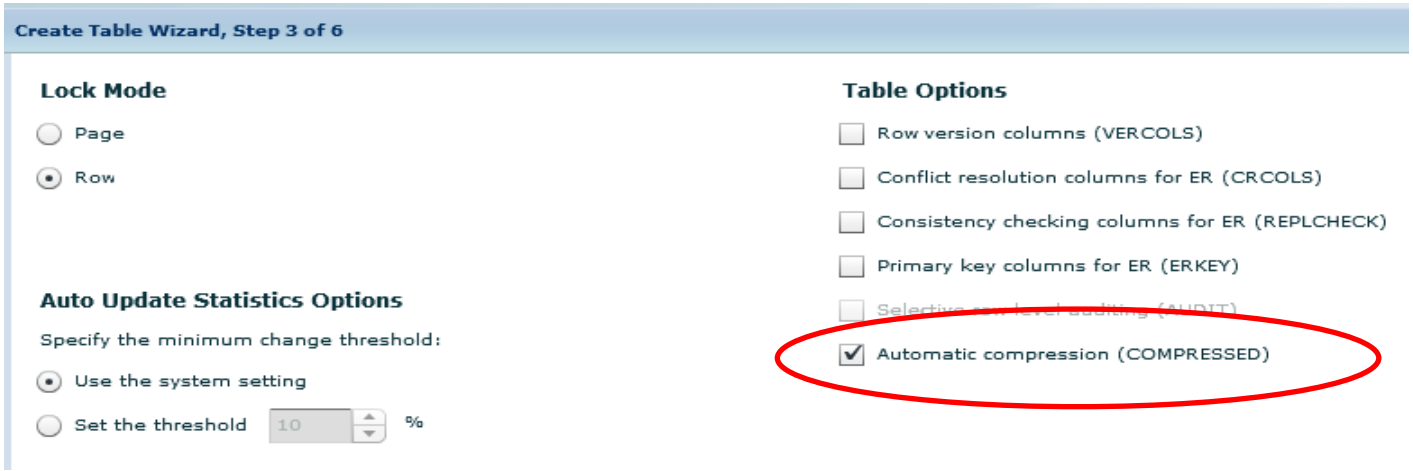
- **Earlier, you could enable:**
  - Compression only after data was loaded

- **Now, with Automatic Compression:**
  - You can enable the feature on tables even if they are empty
  - New data will be compressed as they are loaded
  - Creates compression dictionary for the data rows of a table automatically when it has enough number of rows.
  - Requires a minimum of 2000 rows to create a dictionary

- **Benefits**
  - Informix sets compression as a property of the table, so that:
    - Compression of data rows occur as they are loaded
    - Any new fragments added also get compressed automatically
    - No manual intervention nor external commands to initiate compression
  - To set a table or fragment for auto compression, you can use SQL admin API commands and SQL interface

Storage Optimization – Automatic Compression

- **Enabling Automatic Compression**
  - Using Admin API
    - Execute function task(“table compress”, “my_table”, “my_database”);
    - Execute function task(“fragment compress”, “my_fragid”);
    - Even if there are not enough rows to sample and create a compression dictionary, command will succeed indicating “Auto compression is set”.
  - Using SQL
    - Create table my_table on (my_col ….) compressed;
    - After the table is created, as data is loaded into the table, a compression dictionary will be created when 2000 rows are inserted
  - Using OpenAdmin Tool’s ‘Create Table Wizard’



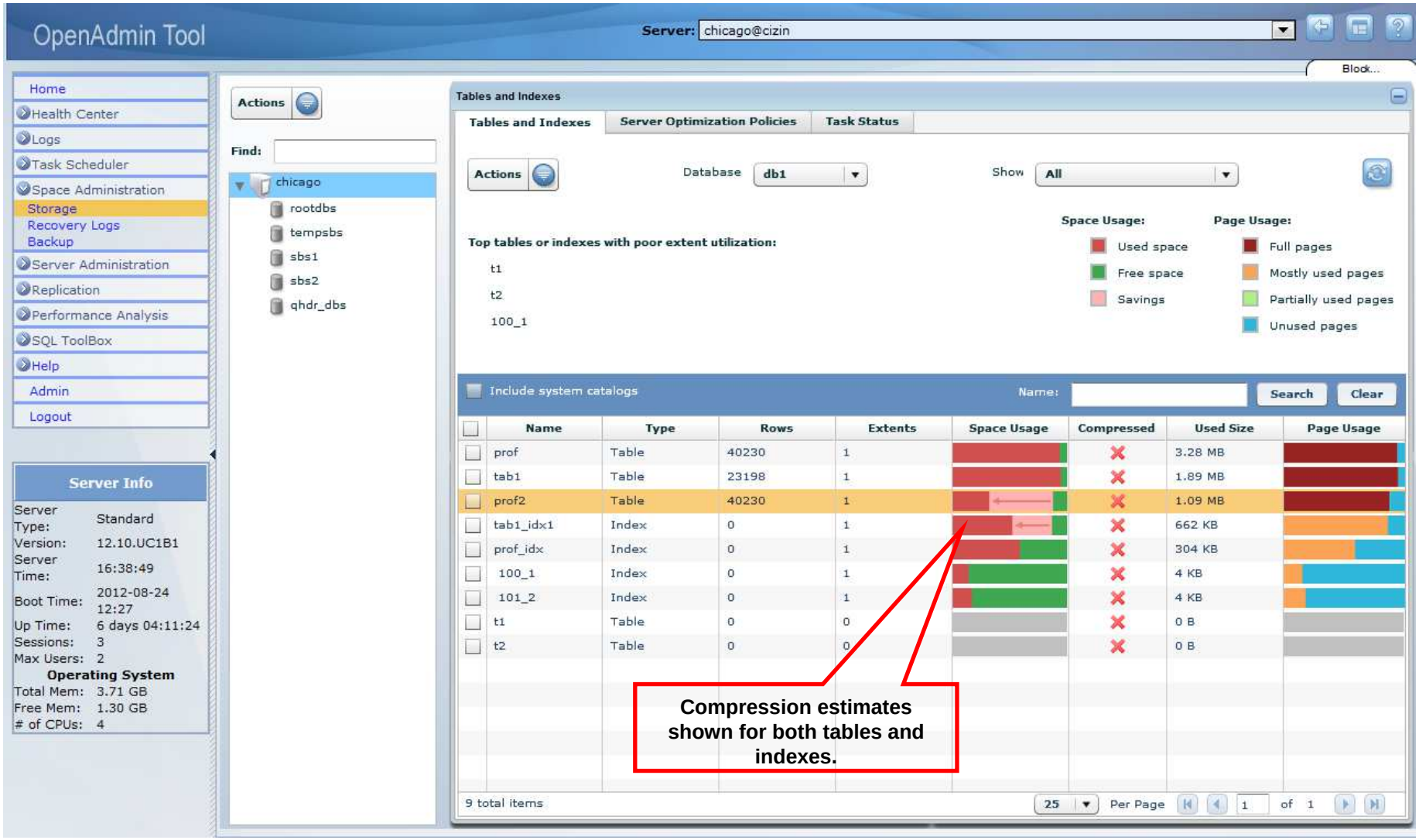
Storage Optimization: Simple Large Objects

- **Save disk space by compressing simple large objects (TEXT and BYTE data types) and indexes**

- **Simple Large Objects (TEXT and BYTE)**
  - Compress data in rows and simple large objects in dbspaces.
  - Compress simple large objects when you compress tables and fragments on the **Space Administration** > **Storage** > **Tables and Indexes** page.
  - Estimate the amount of space you can save on compressing SLOB's
  - You cannot compress simple large objects that are stored in BLOBSpaces

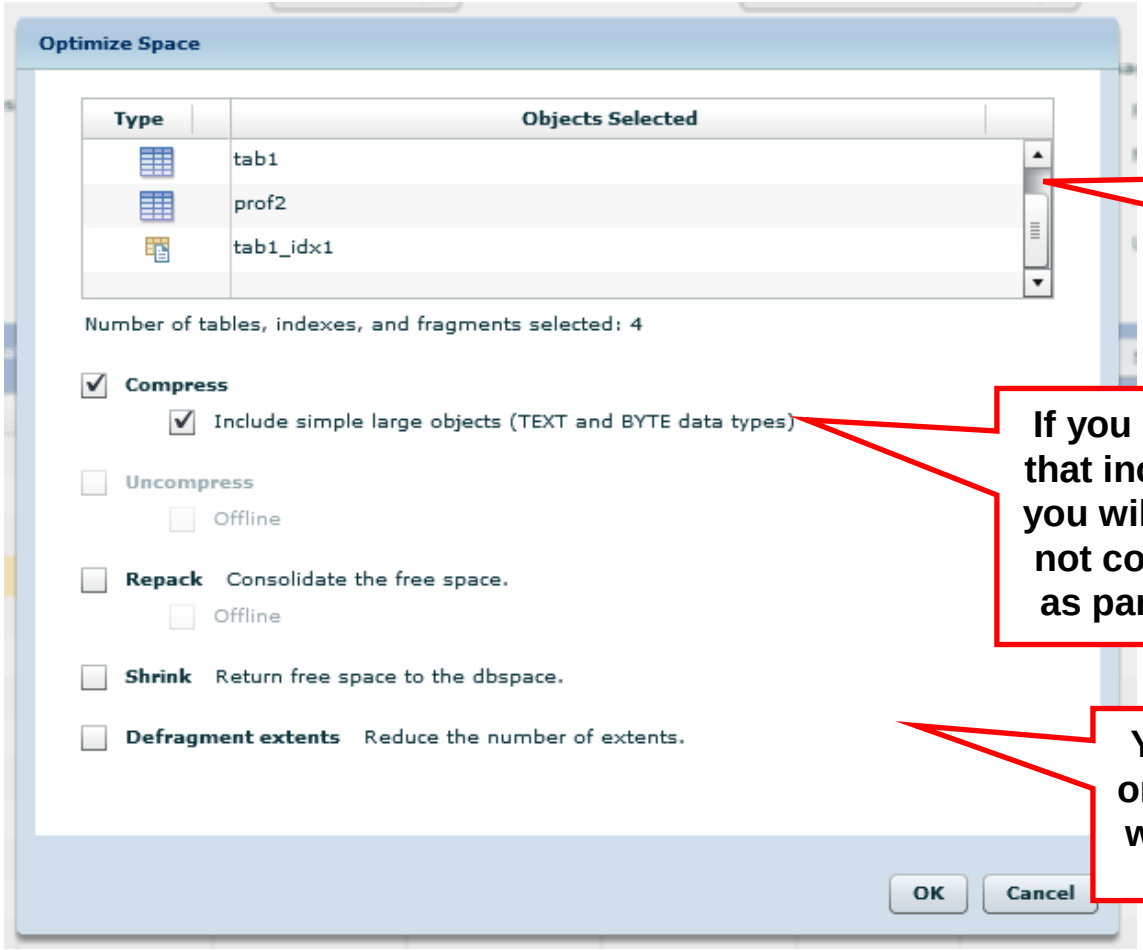
- **B-Tree Indexes**
  - Compress indexes from the same page where you compress tables and fragments.
  - Estimate the amount of space you save on compressing indexes.

Compression Enhancements



Compression Enhancements

- **The Optimize Space action:**



You can choose to Optimize Space on tables or indexes or both.

If you have chosen a table or fragment that includes TEXT or BYTE data types, you will have the option to compress or not compress the simple large objects as part of the compression operation.

You can run compression actions on tables or indexes in conjunction with other actions such as repack, shrink, and defragment.

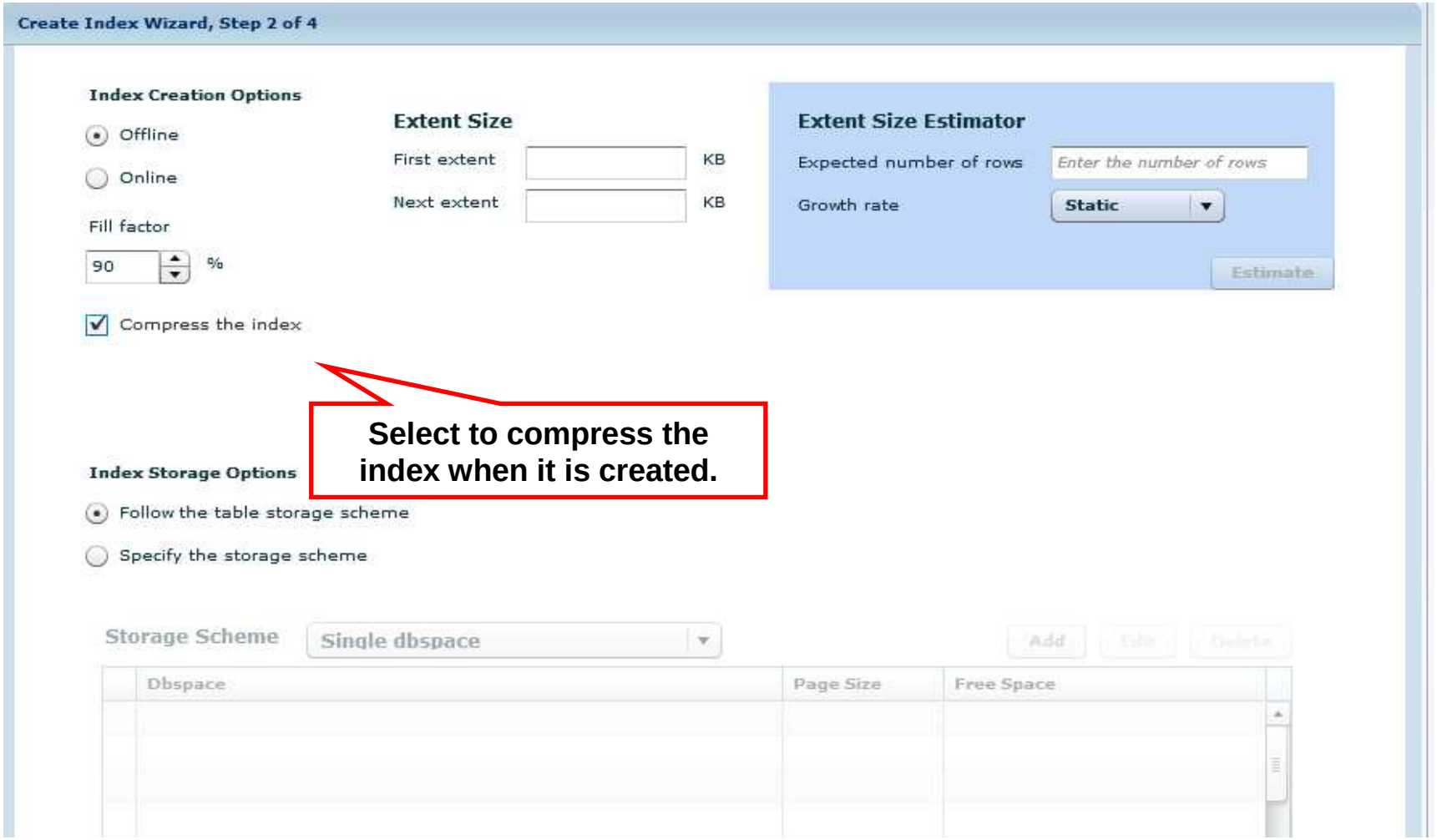
Compression Enhancements

- **For indexes, you can also choose to compress the index at index creation**

- **You will find this option in the Create Index wizard in the Schema Manager plug-in in OAT**
  - Go to the **SQL ToolBox > Schema Manager** page
  - Expand the database
  - Click on the table name for which you want to create the index
  - Expand the Actions menu and select **Indexes > Create Index**
  - The option to compress the index at the time of index creation is found on page 2 of the wizard.

Compression Enhancements

- **Create Index Wizard, page 2**

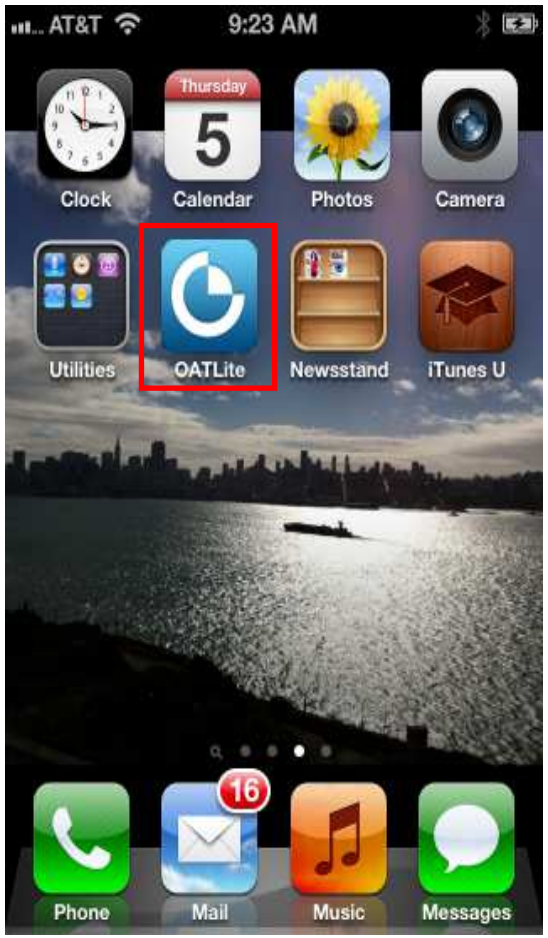


Mobile OpenAdmin Tool for Informix

- **Mobile OpenAdmin Tool is an App available for free download on iOS and Android**
- **Break free from getting tied down to a terminal**
- **Monitor the database server and demand Information on the move**
- **Watch out for Alerts, warnings and other vital signs on the run**
- **The App is designed categorically to suit the business needs**
  - Health Center
    - Information on Alerts and Logs
  - Performance Center
    - Information on Memory, Space, I/O and Tables
  - Users
  - Details
- **Mobile OAT is just an option and not an alternative**
- **Users still need to use a terminal or OAT to act on information provided by Mobile OAT**

Mobile OpenAdmin Tool for Informix

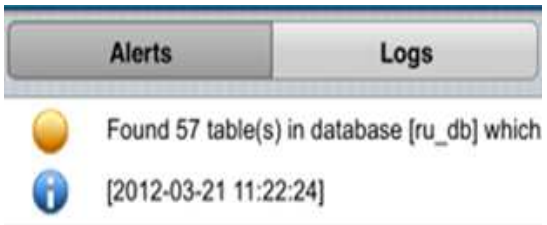
Mobile OAT App on iTunes



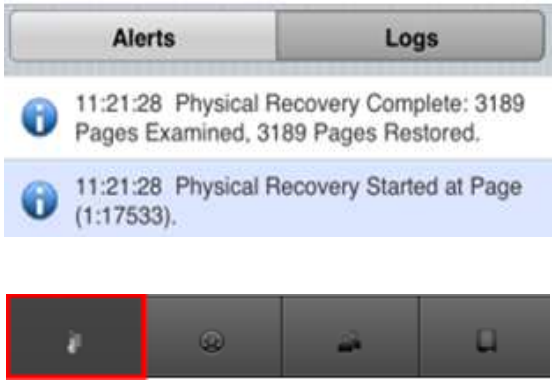
Server Login

Health Tab

Alerts



Logs

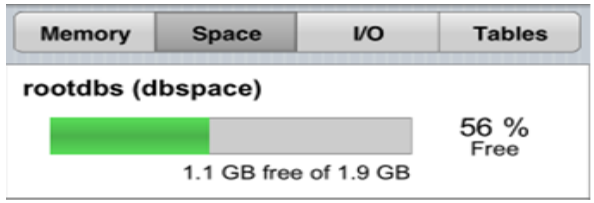


Mobile OpenAdmin Tool for Informix

Performance

Memory

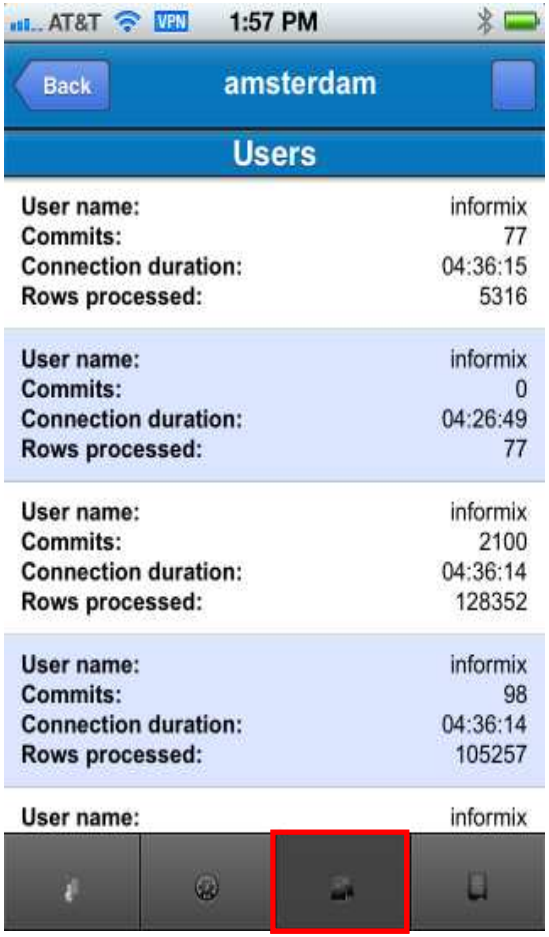
Space



I/O

Tables

Users



Details

Schedule Automatic Backup

- **Configure automatic ON-Bar backups**
  - Backup speed and other onbar parameters
  - Checkpoint and whole system backup (-w option)
  - Number of backup generations to retain
  - Backup schedule

- **For Primary Storage Manager (PSM) the backup and metadata location are displayed**

- **ON-Bar backs up data using a storage manager (SM). Options can be configured in OAT but an SM needs to be configured before data can be backed up.**

171

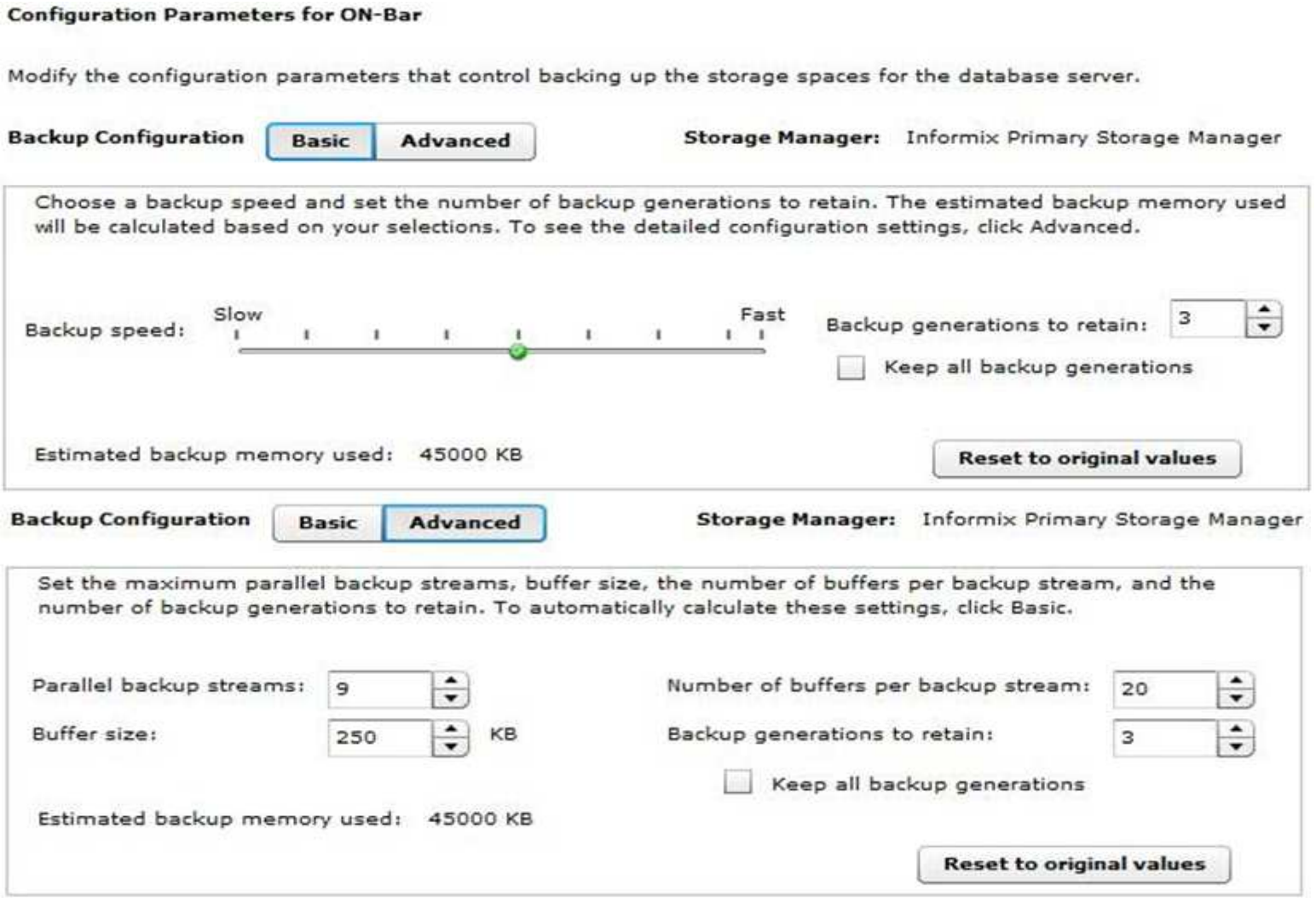
Storage Manager Backup Wizard – Using On-Bar

Step 1: Option 1: On-Bar utilizes Primary Storage Manager

Step 1: Option 2: On-Bar utilizes 3rd Party Storage Manager

Storage Manager Backup Wizard – Using On-Bar

Step 2: Basic & Advanced Configuration



Storage Manager Backup Wizard – Using On-Bar

Step 2: Basic & Advanced Configuration, contd ..

Backups will be scheduled and enabled for all days that are selected.

Level	Start Time	Monday	Tuesday	Wednesday	Thursday	Friday	Saturday	Sunday
Level 0	01 : 00 : 00	☐	☐	☐	☐	☐	☐	☐
Level 1	01 : 00 : 00	☐	☐	☐	☐	☐	☐	☐
Level 2	01 : 00 : 00	☐	☐	☐	☐	☐	☐	☐

Step 3: Review the Configuration

Review the configuration. To change the configuration, click Back.

Step 4: Confirm & Save the Configuration

The backup configuration was saved successfully.
You can change the configuration on the Backup > Configuration page.

Rolling Window Tables

- **Embedded applications have a need to manage a limited amount of space automatically without direct intervention**

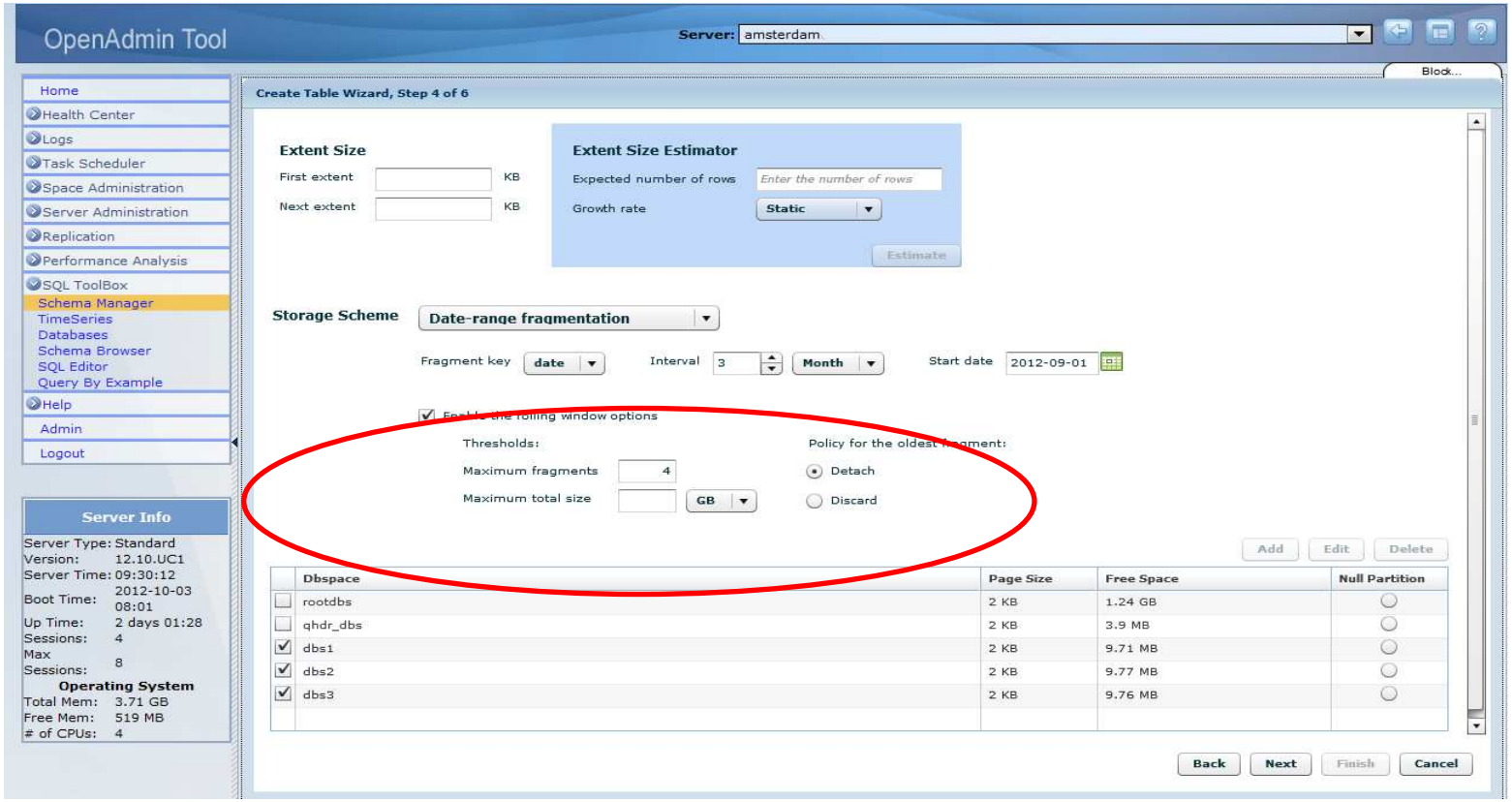
- **Extension of interval fragmentation strategies (called range fragmentation or date-range fragmentation in OAT), which create new fragments when they are required.**

- **Rolling Window Thresholds:**
  - Limit table to a maximum number of fragments
    - When fragments exceed the set value (that is when a new one is created), the one holding the lowest (oldest) set of values will be detached or discarded.
  - Or limit table to a maximum total size
    - When limit exceeded, fragments holding the lowest (oldest) value will be detached/discarded until space used is below the limit.

- **Control the policy for old fragments**
  - DISCARD - eliminate the fragment for good
  - DETACH - preserve the data by detaching the fragment in a new table

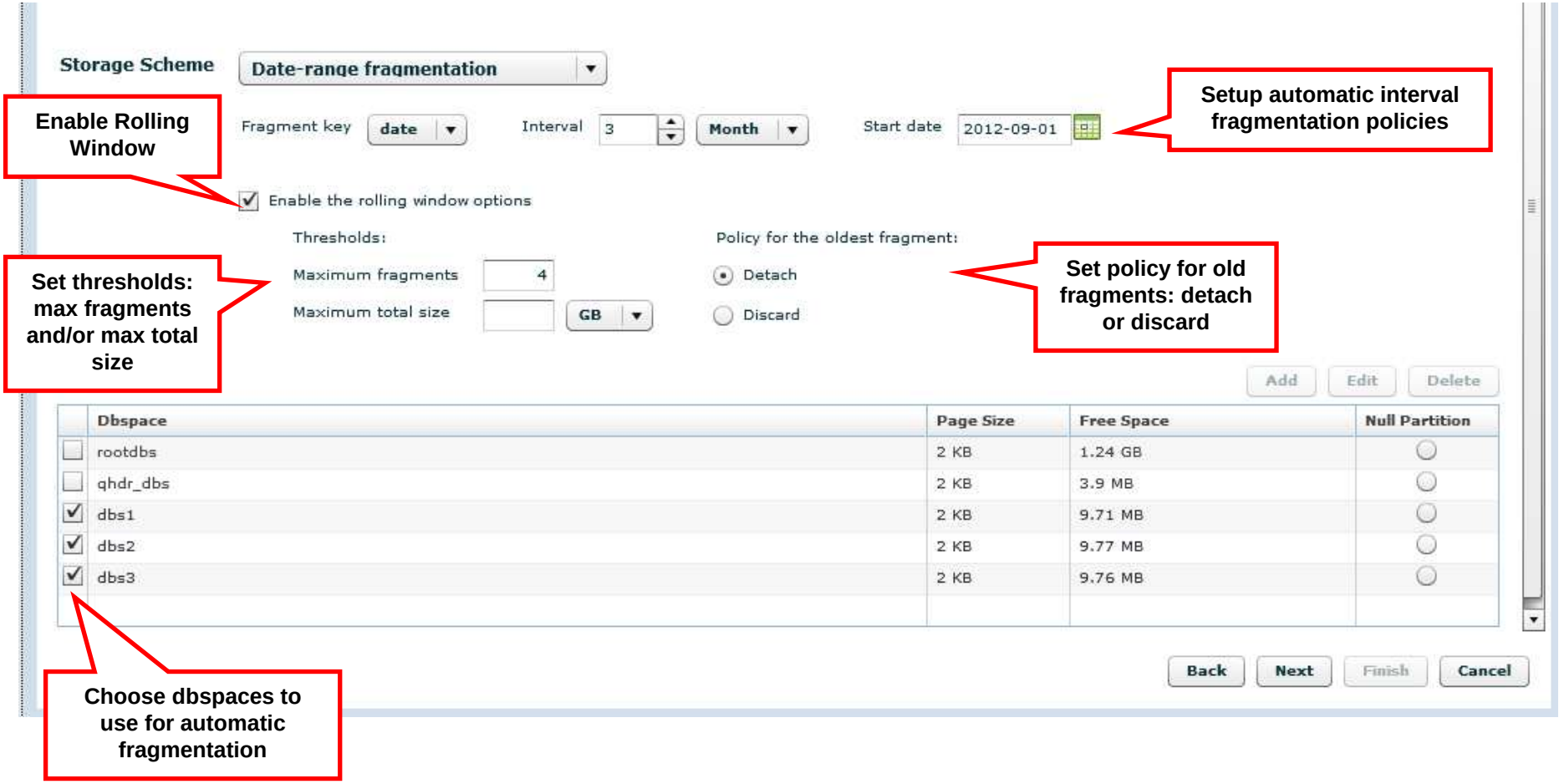
Rolling Window Tables in OAT

- **Setup Rolling Window Tables when you create a table using the** Schema Manager **plug-in's** Create Table Wizard
  - Rolling Window options found on page 4 of Create Table Wizard as part of Date-range fragmentation and Range fragmentation strategies

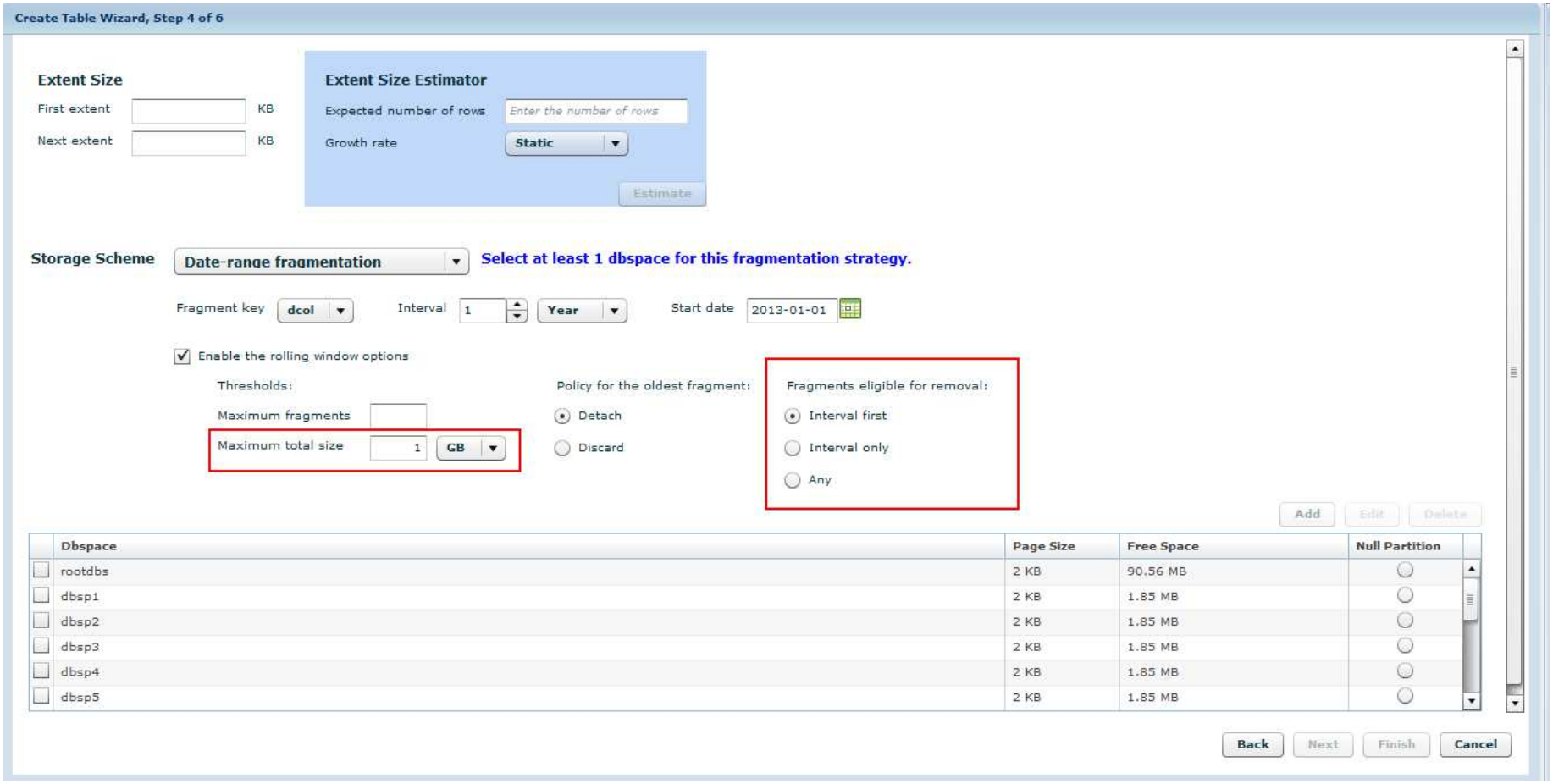


Rolling Window Tables in OAT

- **Setting up Rolling Windows Tables in OAT**



SQL – Rolling Window

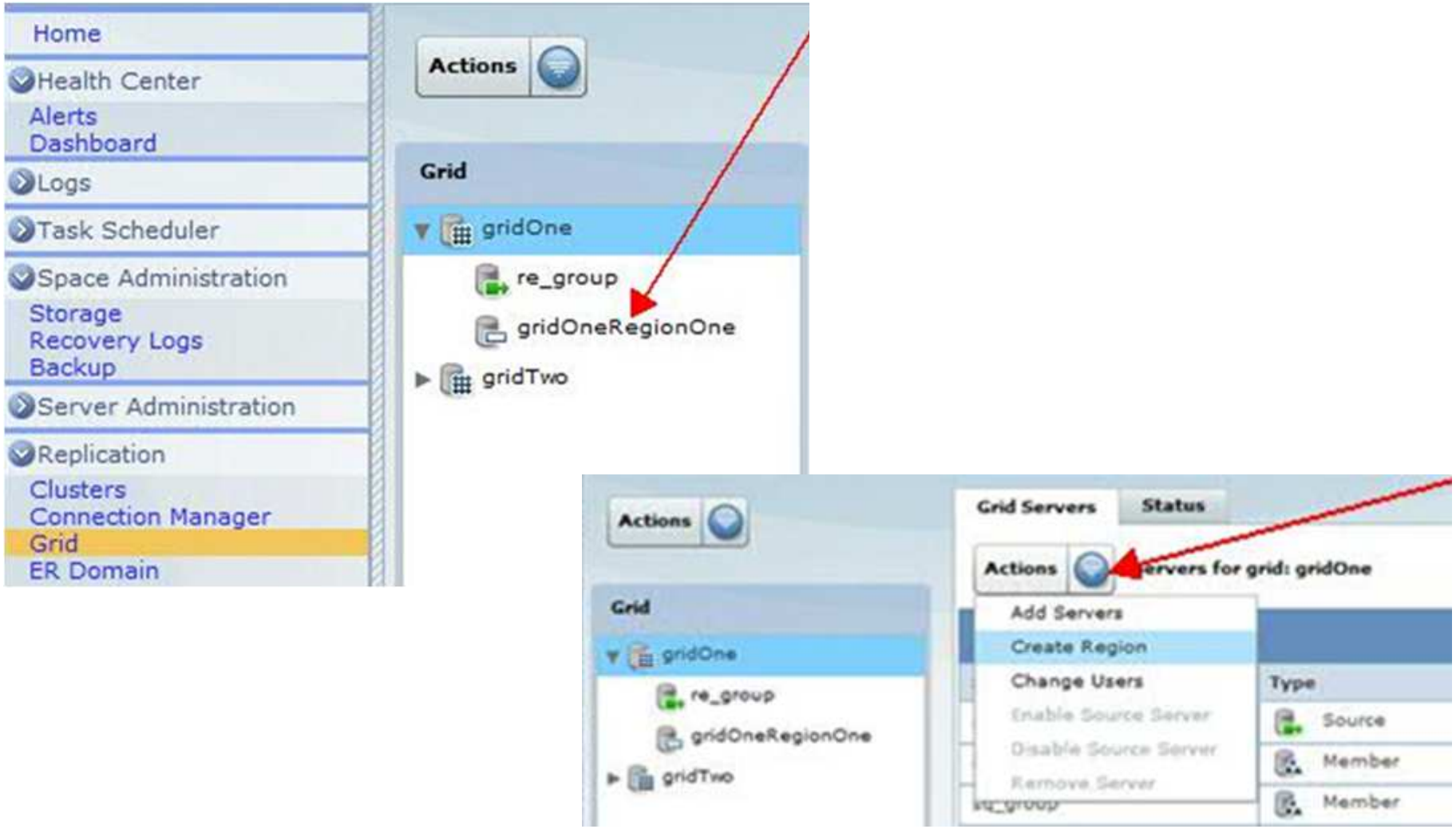


Flex Grid - Administer Grid Queries

- **Administer Grid Regions**
  - Making use of the Replication plug-in
  - Partition the Flexible Grid based on business requirements
  - The Grid Regions are shown on the Left Navigation tree on the Grid page along with the Source servers (screen-shot in the next slide)

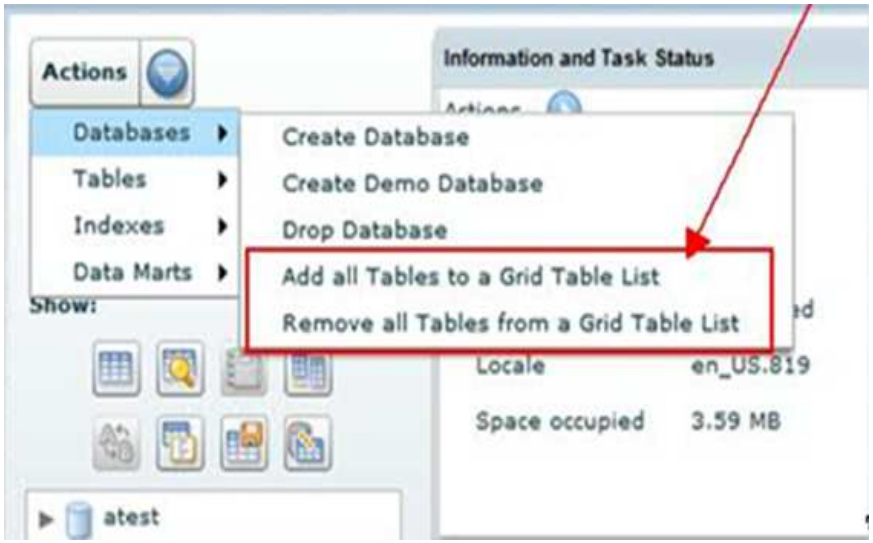
- **Add/remove**
  - All tables to/from Grid list
  - Individual tables to/from Grid list

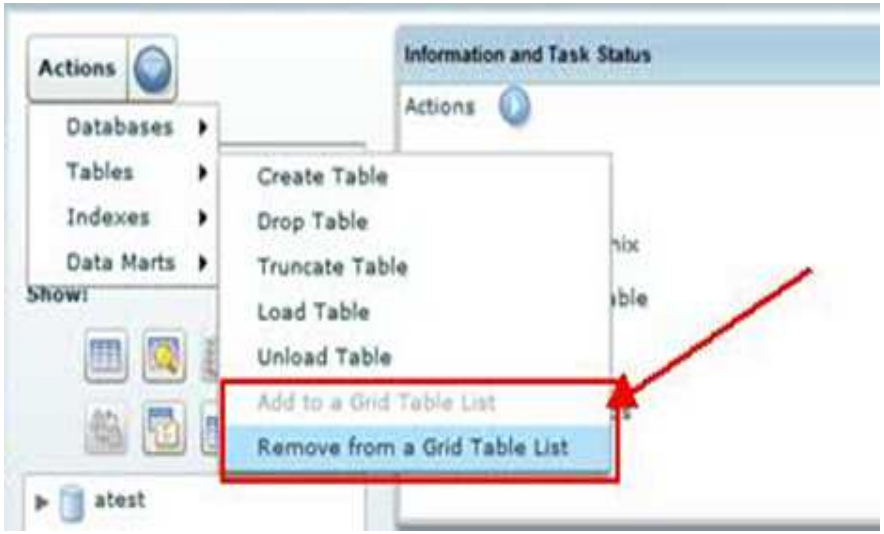
Flexible Grid – Administer Grid Regions



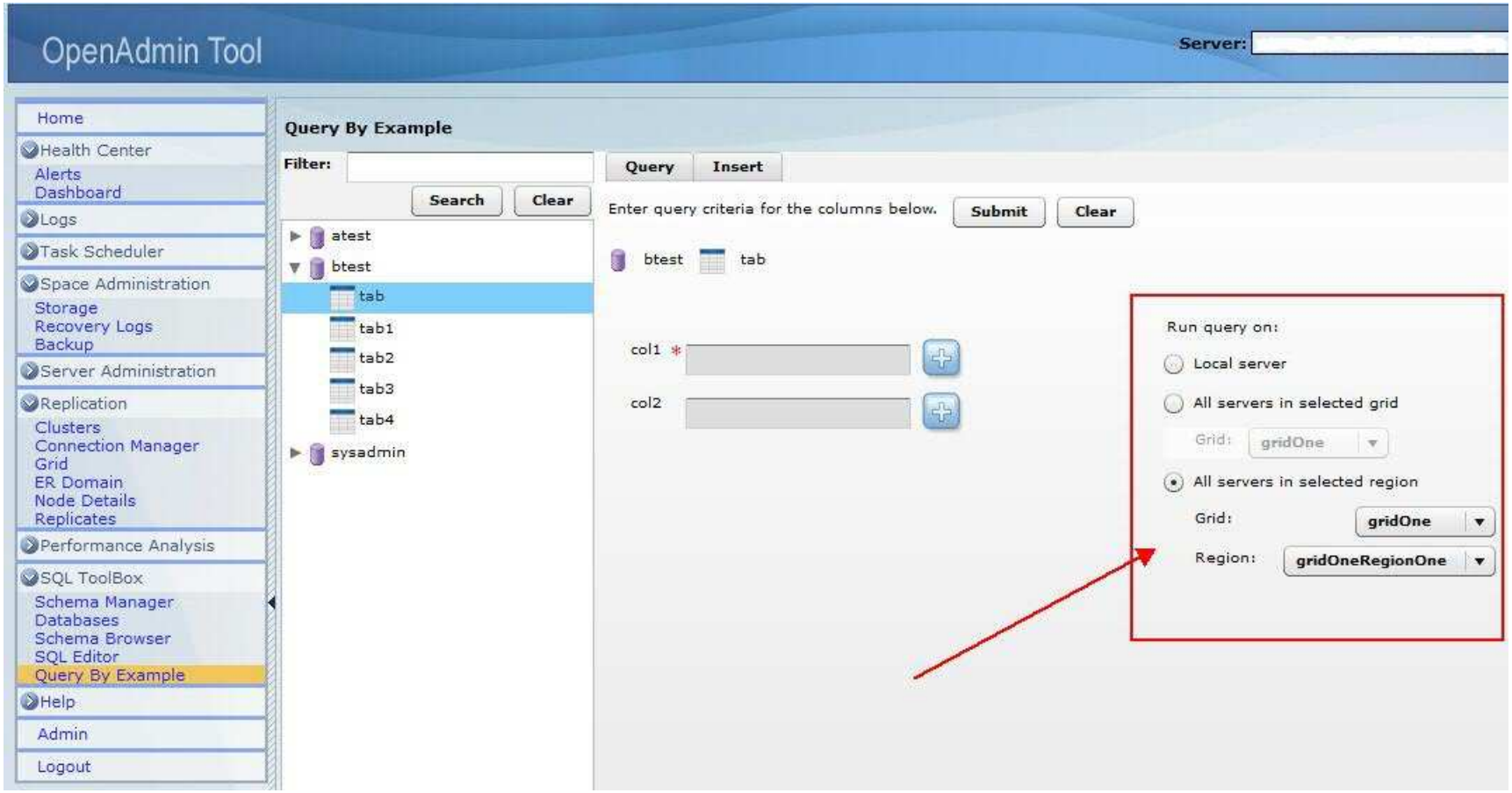
Flex Grid - Add/Remove tables to/from Grid list

- **To perform Add/ Remove tables to/from Grid list:**
  - Use Schema Manager plug-in to work with Grid tables
  - Tables need to be in the Grid List for queries to span across nodes in a Grid
- **Perform Add/Remove:**
  - All tables to/from the Grid list for a database
  - Individual tables to/from the Grid list





Query By Example – Run queries across a Grid or Region

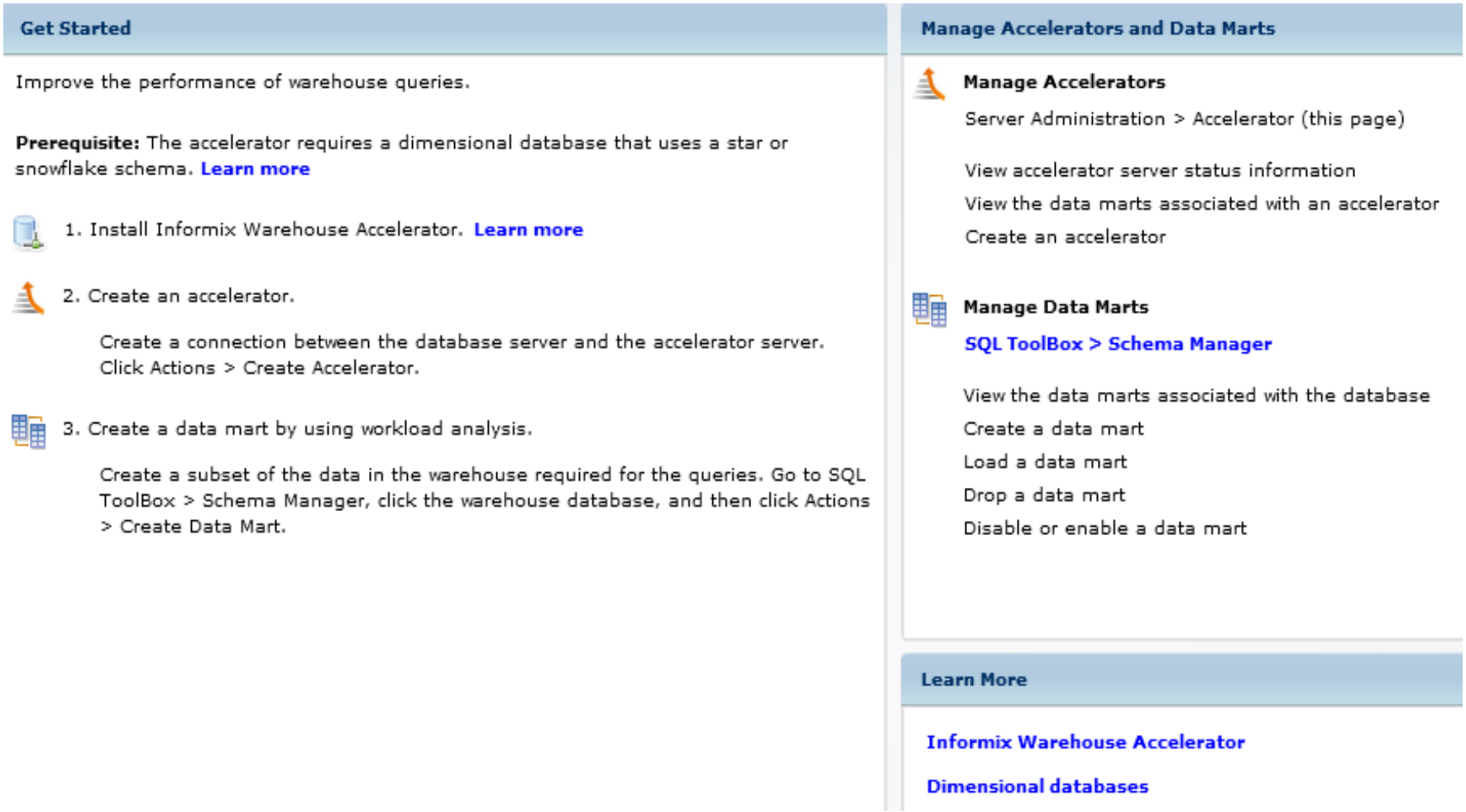


182

Informix Warehouse Accelerator in OAT

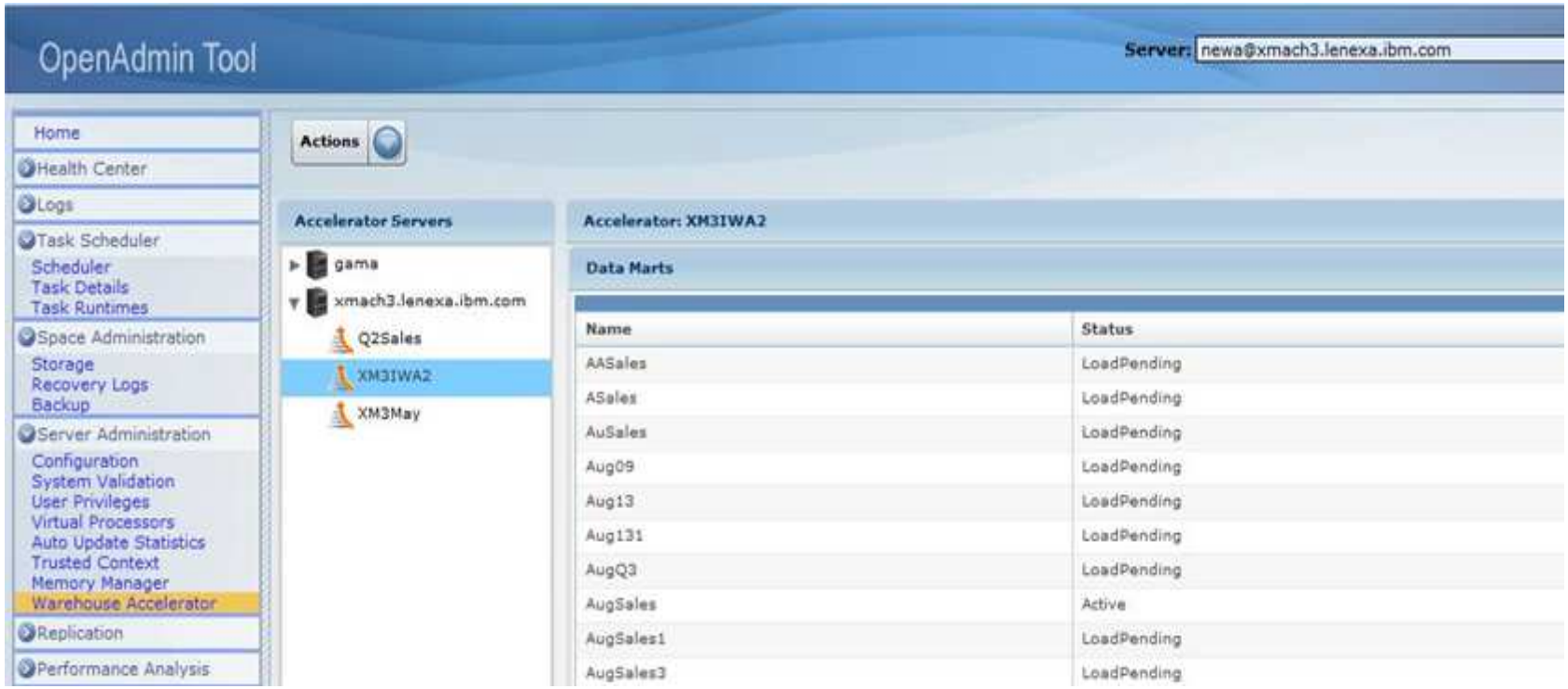
- **Administer the Informix Warehouse Accelerator (IWA) using OAT**
- **Administer Informix Analytics functionality**
- **Administer Informix Warehouse Accelerators and Data Marts.**
  - Create and Drop Accelerators
  - Create Data Marts using SQL workload analysis
  - Enable, Disable, Load and Drop Data Marts
- **Accomplish these warehouse tasks using the same tool (OAT) used for other Informix administration tasks.**

Welcome Page



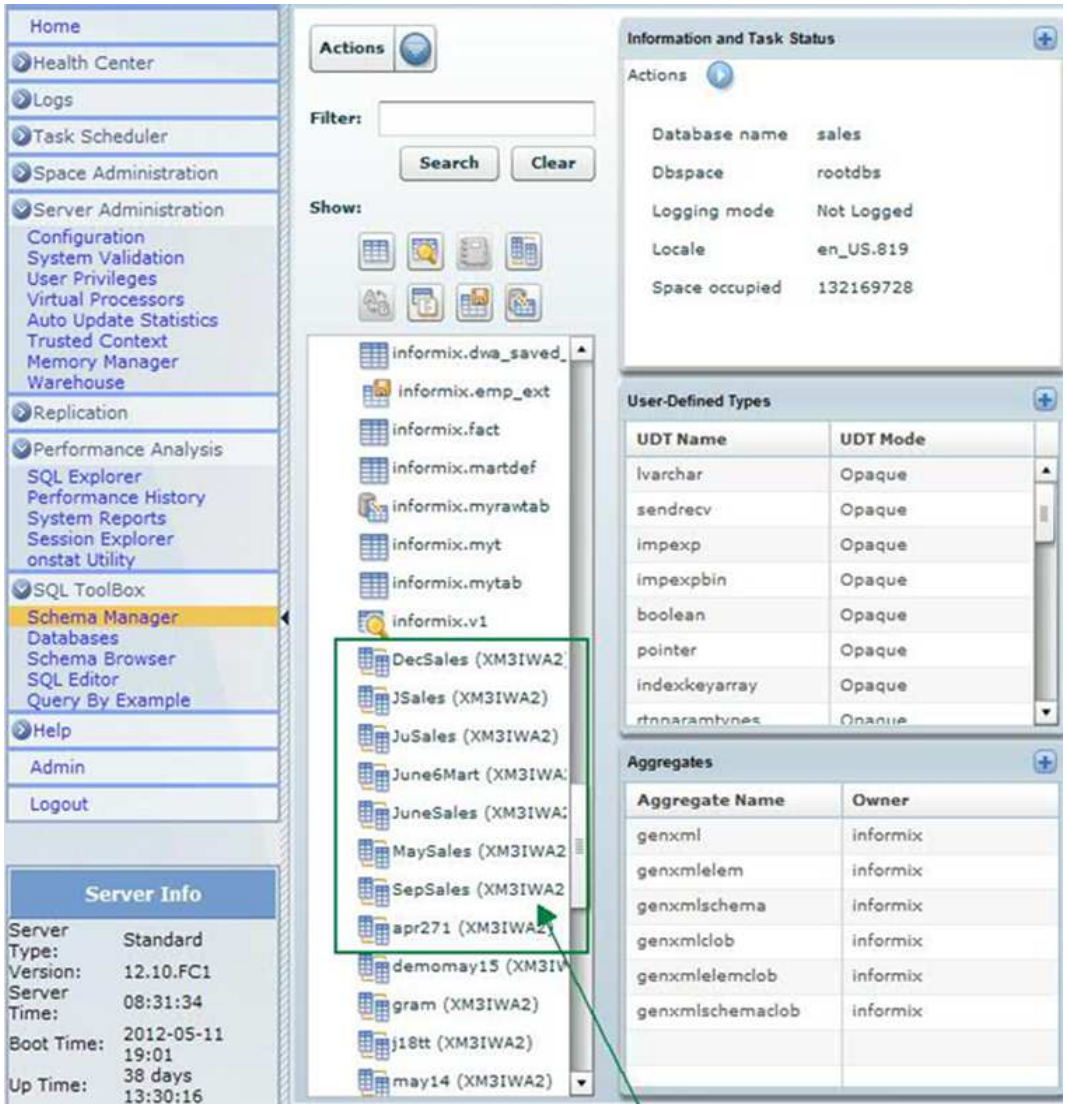
Data Marts Listing

- **Manage Accelerators (Warehouse Administrator)**
- **Manage Data Marts (Data Mart Administrator)**



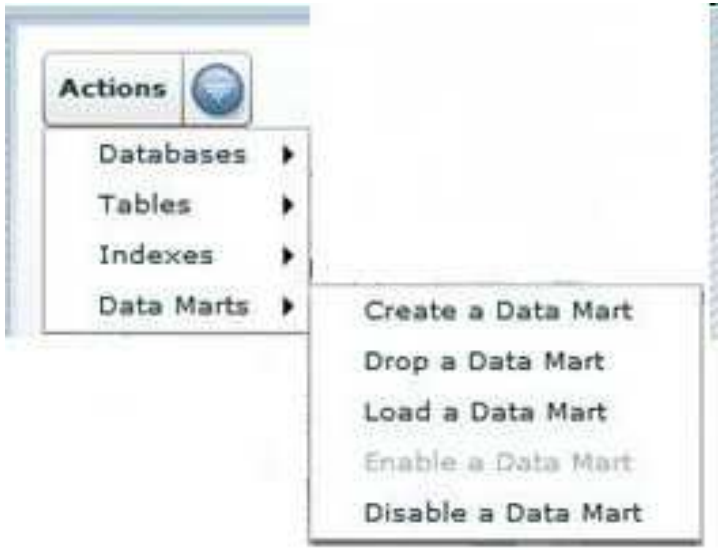
Data Marts in Schema Manager

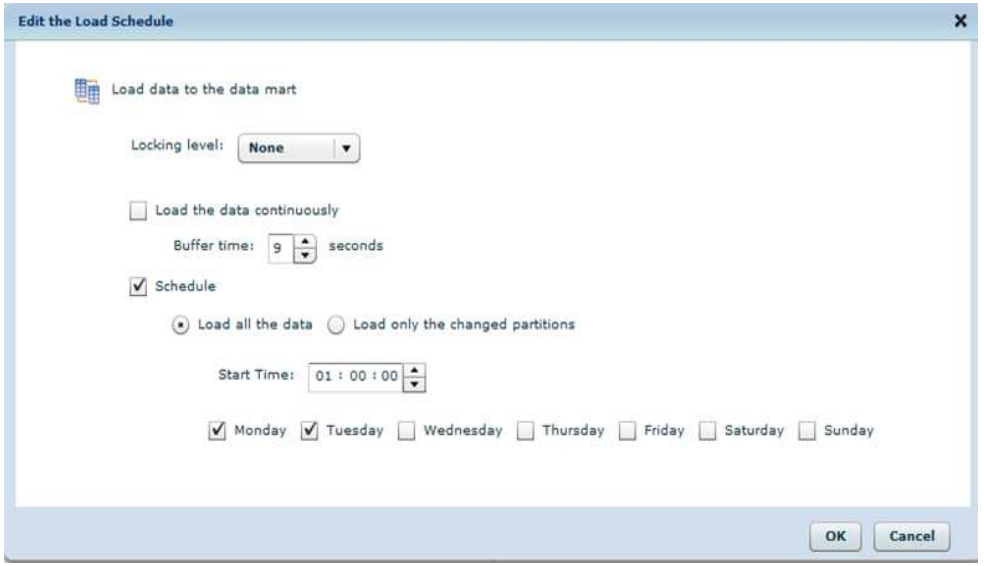
- Data Marts in the Navigation Tree like tables and other database objects
- Filtering on Data Mart names.
- Data Mart status and load schedule information.
- Data Mart table information



Data Mart Actions

- **Create and drop a Mart.**
  - Mart creation using workload analysis of the server.
    - User runs queries that are captured as a workload using SQL Tracing.
    - Workload is probed and Data Mart created in Accelerator Server.
- **Enable and disable a Mart.**
- **Schedule and load a Mart.**
  - Load scheduling setup using DB Scheduler of the server.





IBM Informix Genero v2.41

- **eGA'ed in December, 2012**
- **Genero Application Server and Genero Web Client**
  - HTML5 snippets and CSF (aka javascript) update
  - Enhanced support for Genero features
- **Feature enhancements worth taking a note**
  - The Layout
  - Store Settings and Table Columns UI
  - Traditional mode (to ease migration from i4GL)
  - TTY attributes and Genero 4ST
  - Modal windows and Popup menus
  - Picture flow
    - Better touch based (tablets and phones) ergonomics
    - Android4 compatibility
    - iOS5 (iPad and iPhone) compatibility
    - Hybrid Application templates for iOS and Android

IBM Informix Genero v2.41

- **Enhancements to Genero Studio**
  - BAM has new templates
  - Provide additional modeling options to define the functionality provided by each form, such as add/modify/delete.
  - Data access code and constraints will be centralized at the database meta-schema level.
  - Operations on forms will be published as Web services to provide a high level access on data by program

- **Defect fixes for all components**
  - GWS, FGL, GDC, GRV, GAS, GWC, GST, GRE

IBM Informix v12.10
Customer Quotes (EVP Program)

Customer Quotes (EVP Program)

IBM Informix 12's real strength is to maximize the computing power of your hardware, ***making the dba confident and safe even when load reaches critical levels,*** *while still being extremely simple to administrate.*

- Eric Vercelletto, Data Management Architect, BEGOODEN-IT CONSULTING

Using ***OAT has allowed Century Software to implement a database monitoring solution that satisfies our clients IT teams requirements and gives us the capability to provide support remotely*** *via the browser.*

- Mark Rees, Chief Technology Officer, Century Software

Customer Quotes (EVP Program), cont'd

As the biggest Informix user in Croatia, we had our share of performance challenges over the previous years. This is why performance is a crucial issue for us, so we derived a number of fairly complex stress-tests to verify each Informix version before we engage it in a production environment. ***With this tests we have verified the story that new versions out-of-the-box are faster than previous: using the same hardware and configuration 12.10 was up to 3% faster than 11.70.***

- Ognjen Orel, M.Sc, Project Manager, Special Programmes Department, SRCE - University of Zagreb University Computing Centre