ORACLE 9I OLAP—HOW DOES IT REALLY WORK?

Dan Vlamis, Vlamis Software Solutions, Inc.
dvlamis@vlamis.com

PREFACE

At the time of this writing (January, 2002), Oracle 9i release 2 is still in development. This release adds significant additional capabilities over Oracle 9i release 1. Limited access to information has prevented a thorough review of Oracle 9i OLAP release 2. More information should be available once beta copies of Oracle 9i release 2 become available. The IOUG Live! presentation will include more information on Oracle 9i OLAP release 2.

INTRODUCTION

With Oracle 9i, Oracle finally did it. It took six years since Oracle purchased Express from Information Resources, Inc. (IRI), but Express is finally integrated into its flagship RDBMS engine. Release 2 of Oracle 9i OLAP promises tighter integration, with the 9i OLAP engine being imbedded in the kernel of the RDBMS engine. Data can be stored in relational tables, or in a multidimensional "analytic workspace." In either case, the data can be presented multi-dimensionally.

HISTORY

To understand what has changed, it is best to start back at the beginning of Express. The Express language and environment has gone through many changes over the years. Express has been purchased by various companies searching for an analytic platform for decision support applications.

EXPRESS IS BORN

Express was started by several people from MIT who needed a platform to develop decision support applications. In the 1970's, their company, Management Decision Systems (MDS), developed this into a product on the Primos and IBM Virtual Machine (VM) platforms. This product was eventually called *Mainframe Express* (MFX) or "classic Express." In 1985 the product was rewritten in C and ported to the PC platform running MS-DOS. This product would be known as *pcEXPRESS*. The language went through several changes at this time, but the underlying concepts used remained the same—storing data in multidimensional arrays.

IRI ACQUIRES MDS

Information Resources, Inc. (IRI) had been using Express as a platform to analyze its consumer packaged goods supermarket scanner data. In 1985, IRI purchased MDS and eventually developed *DataServer* (later renamed Oracle Sales Analyzer) as a software product designed to deliver its scanner data and the analysis tools required by its clients. Along the way, IRI ported the new pcEXPRESS database engine to various mainframe environments and added many extensions still found in Express today.

ORACLE PURCHASES EXPRESS

In 1995, Oracle Corporation needed a tool to analyze its financial data and was impressed enough with Express and the product eventually renamed Oracle Financial Analyzer that they bought the Express product line from IRI. Oracle opted to keep the Massachusetts-based development team in a separate division rather than integrate the developers into its California-based organization. Oracle attempted to integrate Express into its flagship RDBMS product offering, especially with its Relational Access Manager offering, but never truly integrated the tools or databases. In 1998, Oracle integrated the development teams into the rest of its organization. Major enhancements necessary to support data warehouses in the RDBMS made possible delivering OLAP functionality directly out of the RDBMS engine, and the 9i OLAP project was started as part of Oracle 9i development.

9i OLAP Introduced

9i OLAP changes the way that Oracle-based OLAP data is stored and OLAP applications are built. Data can now be stored natively in the RDBMS. Access to the data is through a new Java OLAP API (application programming interface). BI Beans provides a new interface for accessing the data. Managing OLAP data no longer requires knowledge of an environment foreign to traditional RDBMS developers and DBAs.

Storing Data Relationally Instead of Multi-dimensionally

At its heart, 9i OLAP is about storing data in a series of standard relational tables of rows and columns and accessing the data as if it is stored multi-dimensionally. This, of course, enables standard Oracle tools such as loaders, backup utilities, and data management tools to be used against this data. Materialized views enable higher-level totals to be stored in a data warehouse for efficient access, eliminating many of the traditional performance problems in ROLAP systems. By placing the responsibility for maintaining these materialized views in the server database (instead of application code), applications are greatly simplified and are much less error prone.

RDBMS Data Access Catches up to MDBMS Data Access

Traditionally, one of the key benefits of a MDBMS has been fast data access. Array access is inherently fast because a DBMS can calculate the exact position to locate a cell of data through simple multiplication and addition—no indexes required. See *How Does Express Really Work Anyway* at www.vlamis.com for more information. As larger multidimensional databases are stored with greater sparsity, the need to efficiently compress out NA values with structures such as conjoints and composites has mitigated this advantage. In addition, advanced index methods such as bitmapped indexes have improved RDBMS access for data warehousing purposes. RDBMS systems are finally up to the task of managing OLAP data.

Analytic Workspaces Provide Multidimensional Storage

So, what if you still want to store data multi-dimensionally? Oracle retained the ability to store data natively in arrays with analytic workspaces. Oracle did not advertise this heavily with Oracle 9i OLAP release 1. Express databases are still there—they are just called "analytic workspaces." They are even still stored in separate files with a .db extension. This lack of full integration explains why Oracle didn't advertise this fact when Oracle 9i OLAP first came out. In release 2, Oracle will be changing the data storage mechanism to remove this disadvantage and will be storing the analytic workspaces in special structures in the RDBMS. This, along with access to data in a analytic workspace via SQL, will allow Oracle to take full advantage of multidimensional data

BI Beans Enable Rapid Application Development

While the Java OLAP API enables Java programmers to access 9i OLAP data, most developers will want to use Oracle's enterprise Java Beans for Business Intelligence (BI Beans) to develop applications. These enterprise Java Beans provide the visual interface to interact with 9i OLAP data. Although BI Beans works in many standard Java Beans integrated development environments, Oracle delivers wizards with JDeveloper that simplify the task of developing OLAP applications.

Working with 9i OLAP—Server Side

9i OLAP is part of Oracle 9i. It is so integrated with the rest of the RDBMS that it is not possible to install just 9i OLAP without the RDBMS. As of the writing of this paper, software is unavailable so descriptions are necessarily vague. Some concepts are very familiar, with seemingly only name changes, while other concepts are totally new.

Managing 9i OLAP

The 9i OLAP management tools are integrated into the rest of Oracle Enterprise Manager. What used to be the Express Instance Manager is now the 9i OLAP Instance Manager. Express services has been renamed 9i OLAP. While some of the configuration keys have changed, many of the concepts and details remain the same, such as individual sessions, and settings such as ServerDBPath. Other areas have totally changed, such as the replacement of Batch Manager with the job scheduler in Oracle Enterprise Manager and using RDBMS security to control which user ids have rights to manage 9i OLAP.

JAVA OLAP API EXAMPLES

The Java OLAP API is the application programming interface that provides multidimensional access to data warehouse data. The API includes primitives that handle typical OLAP needs, such as calculated measures and selection scripts. The following code fragment demonstrates selecting the products where the dollars measure is greater than 1,000,000 for geography Orlando for time period May2001:

```
Source geogSel = geography.selectValue("ORLANDO");
Source timeSel = time.selectValue("MAY2001");
Source result = prodSel;
NumberSource DollSel = (NumberSource) dollars.join(geogSel).join(timeSel).join(prodSel);
Source prodSel = product.select(DollSel.gt(1000000));
```

When data is stored in relational tables, 9i OLAP translates this OLAP API language into SQL, using new SQL primitives such as concatenated rollup, scrollable cursors, and query rewrite for efficient access.

When data is stored in analytic workspaces, 9i OLAP translates this OLAP API language into OLAP DML (a new name for Express SPL).

Since the OLAP API translates into either SQL or OLAP DML, the same application can access RDBMS or analytic workspace data with a few simple changes.

WORKING WITH 9I OLAP—CLIENT SIDE

While the Java OLAP API is available, few developers will actually use it. Oracle primarily built the Java OLAP API for its own internal use to provide a foundation for future development. Rather than using the Java OLAP API, applications will be developed using Java using Oracle's BI Beans, delivered with JDeveloper 9i. As of the writing of this paper, BI Beans is still in beta, so descriptions are necessarily vague.

BI BEANS ENABLE ACCESS TO OLAP DATA FOR THE MASSES

BI Beans are enterprise JavaBeans that provide access to 9i OLAP calls through a standard interface. Some Beans are "back-end" beans that supply services such as database connections, while others are user interface beans such as Table, Crosstab, and Graph. These form the heart of an OLAP user interface.

CONCLUSION

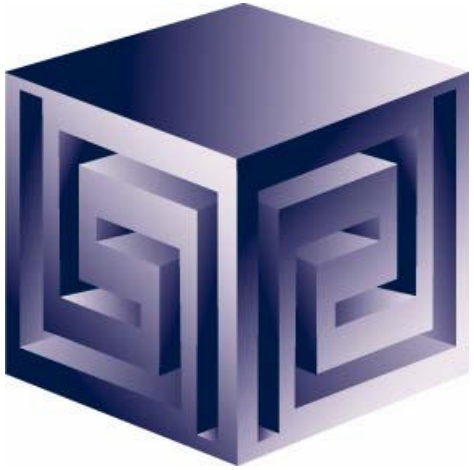
So what does the future hold? Clearly, 9i OLAP represents a major leap forward for integration between Oracle's RDBMS and OLAP offerings. This integration makes it possible for all Oracle databases to benefit from OLAP technology. Many of the concepts applicable to Express design carry over to Oracle 9i OLAP, but the changes in storage and new user interface possibilities will cause most users of existing Express-based applications to reevaluate what can and can't be done. Oracle 9i OLAP release 2 extends the technology to effectively re-introduce multidimensional storage in analytic workspaces. This will enable better storage mechanisms for "personal data" such as forecasts, what-if analyses, and scenario management. Adoption of 9i OLAP has been slow because without BI Beans, there was no "interface" to take view 9i OLAP data. With BI Beans release, the only item missing is the analytic applications based on BI Beans. You can wait for Oracle to develop these applications, develop them yourself, or purchase one from a third party. Why wait? The power of OLAP in the relational database is here today.

Oracle 9i OLAP—How Does it Really Work?

presented at

IOUG Live! 2002

paper number 409



Presented by:

Dan Vlamis (dvlamis@vlamis.com)

Vlamis Software Solutions, Inc.
(816) 781-2880

http://www.vlamis.com

Vlamis Software Solutions, Inc.

- **Founded in 1992 in Kansas City, Missouri**
- **A Member of Oracle Partner Program since 1995 along with various Oracle Beta Programs**
- **Designs and implements databases/data marts/data warehouses using RDBMS and Multidimensional tools**
- **Specializes in Data Transformation, Data Warehousing, Business Intelligence, Applications Development**
- **Founder Dan Vlamis is former developer of Sales Analyzer Application**
- **Oracle Certified Solutions Provider**

ORACLE | CERTIFIED SOLUTION PARTNER

Disclaimer

Oracle 9i is being changed in release 2 and all statements, illustrations and features shown during this presentation are subject to change.

Agenda

- **Historical background**
- **Why Oracle 9i OLAP?**
- **What is Oracle 9i OLAP and how does BI Beans fit in?**
- **Oracle9i OLAP storage options**
- **Structure of Analytic Workspace**
- **Differences from Express**

Requirements of OLAP Systems

- **Multidimensional user view**
- **Drill down, rotate**
- **User-created measures**
- **Iterative discovery process**
- **Multiple levels (embedded totals)**

Do these attributes imply a proprietary MDBMS?

No.

Key Developments

- **Integrating Express Server team into Oracle Server team**
- **RDBMS gets OLAP functionality in SQL**
- **Data Warehouse features in Oracle 8i:**
  - **Bitmap and bitmap join indexes**
  - **Materialized views**
  - **Query rewrite**
  - **“N-pass” functions**
- **Increasing use of very sparse data**
- **Oracle focus on integration**

Express Features and Limitations

- Multidimensional data store provides quick access
- All data in single proprietary file
- Express SPL powerful for calcing and extending apps
- GUI environment uses Visual Basic language
- RAM moves data from RDBMS into Express
- Designed for multiple read/only users
- Difficult for IT to adopt, Max database size
- Difficult to integrate and schedule back ups
- Long learning curve for new developers
- Not consistent with Oracle direction
- Star and snowflake only, performance concerns
- No multi-writer support

Express Has Not Gone Away... It has just been absorbed

- **Analytic Workspaces are Express DB files**
- **Oracle marketing will downplay Express**
- **Design considerations for Analytic Workspaces same as for Express DBs**
- **Express engine still there**
- **Express SPL still there (but only operates on storage in analytic workspaces)**
- **Analytic Workspaces better at complex calculations (for now)**
- **Use OLAP Worksheet for OLAP DML commands**

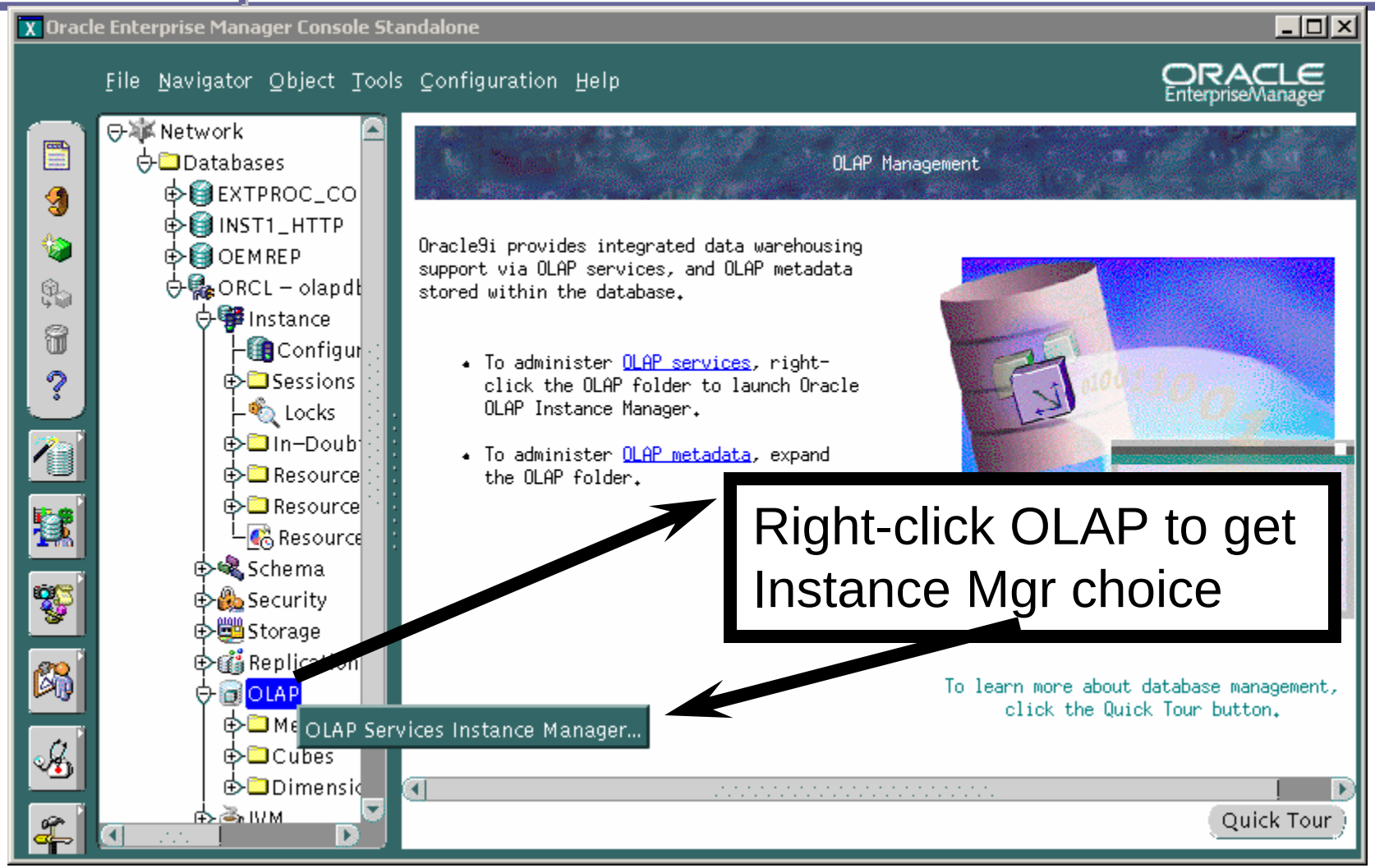
Components of 9i OLAP

- **Instance Manager now in Oracle Enterprise Manager**
- **OLAP folder of OEM defines multidimensional structure (replaces RAA)**
- **Query Processor processes Java OLAP API**
- **OLAP Worksheet provides command line to OLAP DML**
- **BI Beans provide linkage to Java OLAP API**
- **JDeveloper is environment for building apps (replaces OEO)**

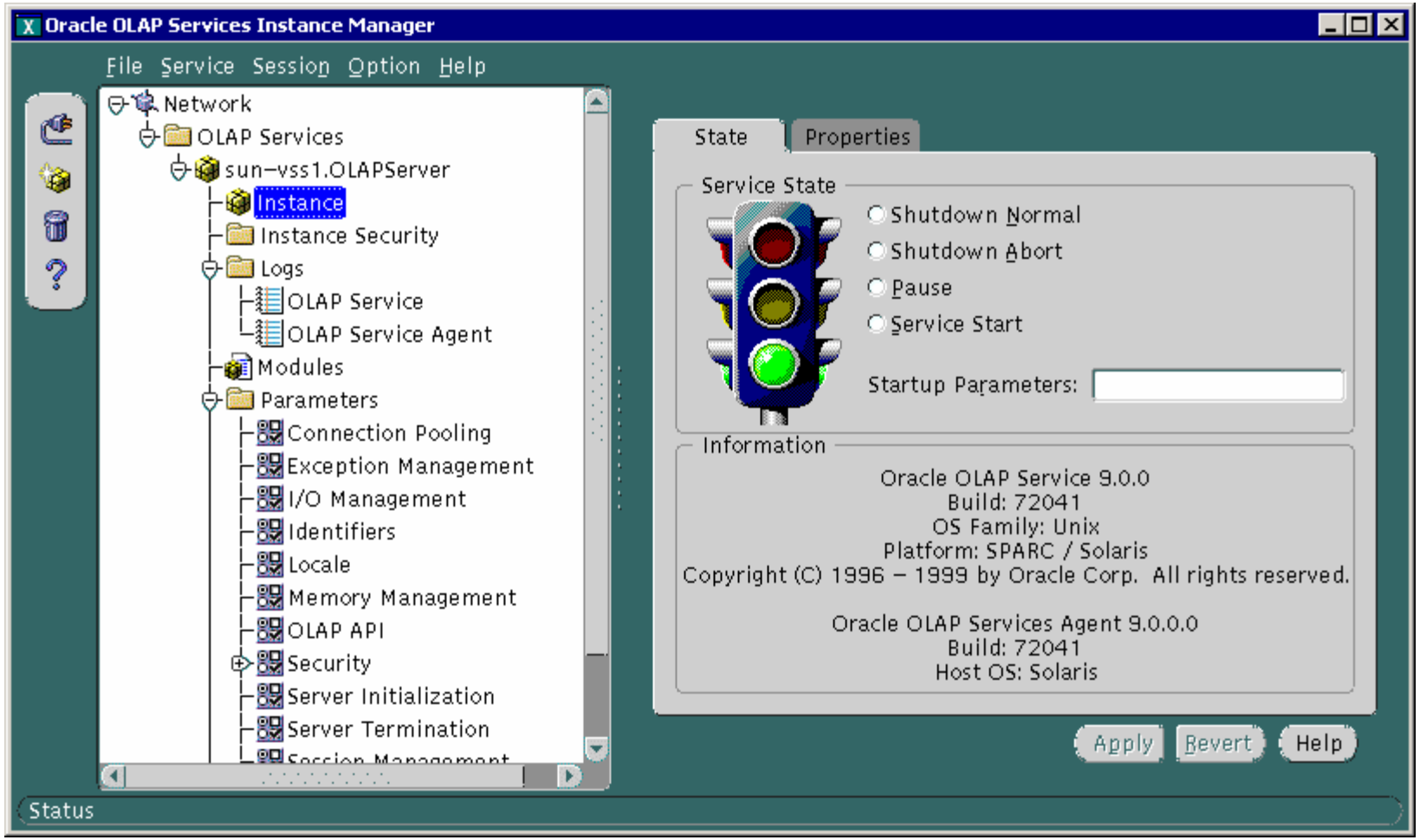
Administration with OEM

- Acquiring System Administration Privileges
- Starting / Stopping / Pausing OLAP Services
- Changing the Configuration Settings
- Managing Sessions
- Viewing Status Messages
- Granting Access Rights to Users
- Creating Databases (Metadata, dimensions, Measures and Cubes)
- Running Batch Jobs
- Scheduling Jobs
- Managing the Service Environment
- Managing OLAP Services Agent

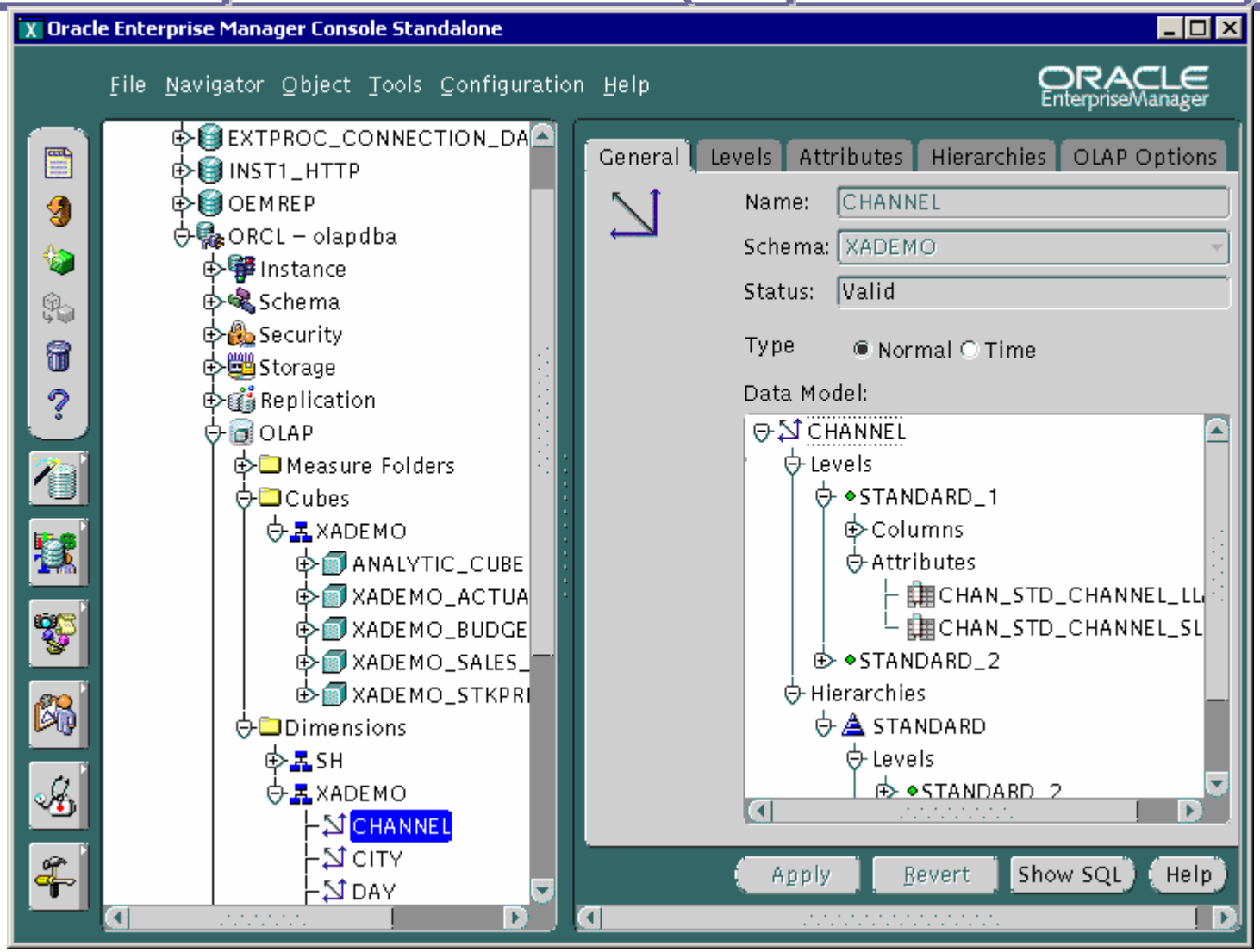
Oracle Enterprise Manager Sample Screen



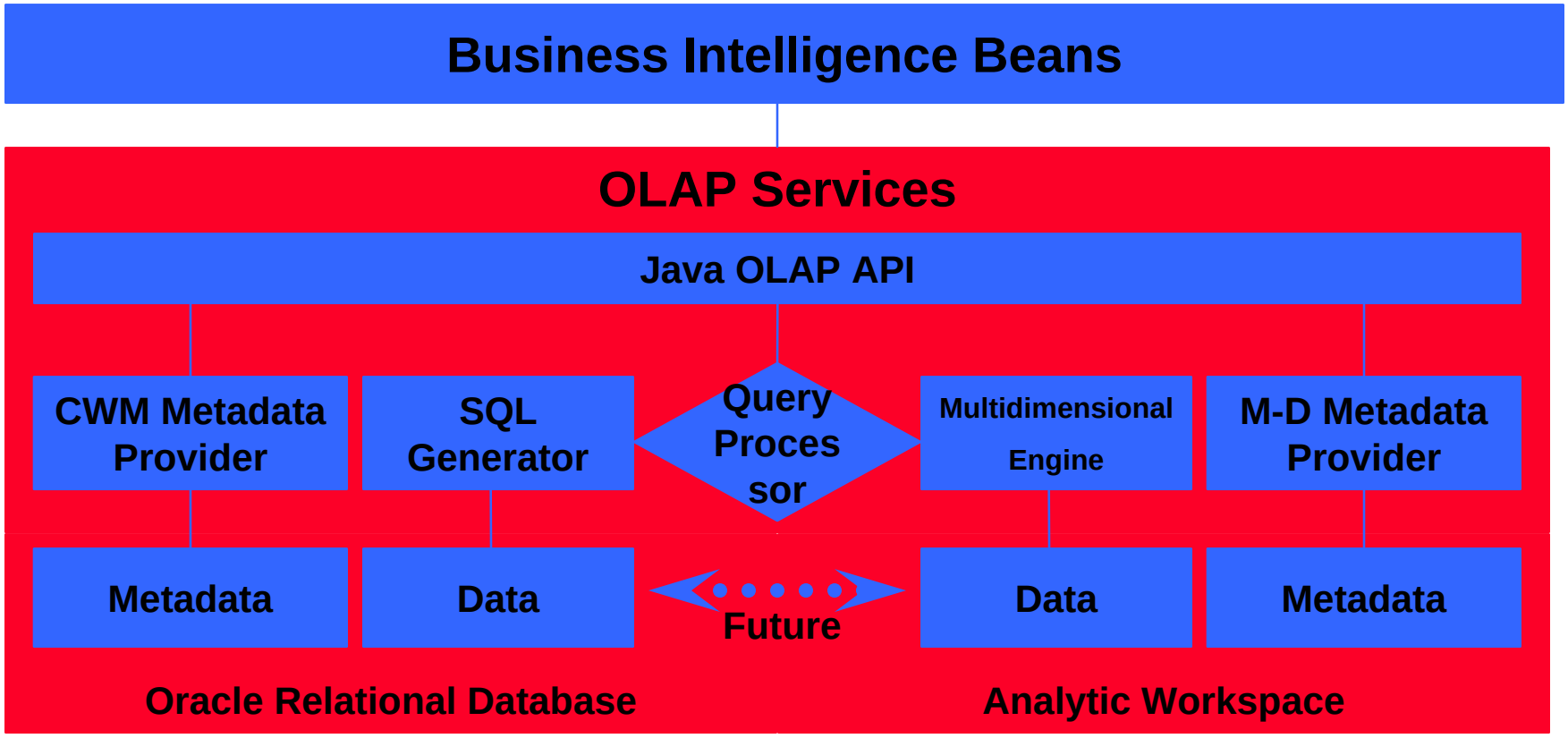
OLAP Services Instance Manager



Oracle Enterprise Manager Sample Screen (replaces RAA)



Oracle 9i OLAP Query Path



Data Warehouse - Query and Reporting

Forecasts • Models • Allocations
Consolidations • Scenarios • Custom Functions

Query Processor Details

- **If data in Analytic Workspace:**
  gets data using OLAP DML
- **If data in RDBMS:**
  generates SQL, using new N-pass functions, etc. to calculate data necessary
- **Can generate incredibly complex SQL**
- **Data filtered in RDBMS**

Java OLAP API

- **Object-oriented**
- **Mathematically consistent**
- **Java (industry standard)**
- **Compatible with JOLAP standard**
- **Declaratively (not procedure) based**
- **Data in relational or analytic workspace**
- **Multidimensional cursors**
- **Really designed for low-level access**
- **Most developers will use BI Beans instead**

Metadata Required

RDBMS data

- **Stored in "Olap catalog"**
- **Edited with OEM**
- **Based on CWM**
- **Once set up, can use OLAP API against data**

Analytic Workspace data

- **Stored in Express Common Metadata (ECM) format**
- **Similar to OEO/Adminstrator metadata**
- **Once set up, can use OLAP API against database**
- **ECM stored in series of catalogs in AW so 9i OLAP and BI Beans can access**

OLAP API Calculation Capabilities

- **Multidimensional object model**
- **Totals broken out by multiple attributes**
- **Row and column calculations**
- **Union dimensions**
- **Measures as dimensions**
- **Calculated dimension members (e.g. income 0-20K, 20-50K, 50-75K, >75K)**
- **Asymmetric queries**
- **Multiple measures per cell (e.g. color-coding)**

Simple Java OLAP API Example

English

Select the products where the dollars measure is greater than 1,000,000 for geography Orlando for time period May2001.

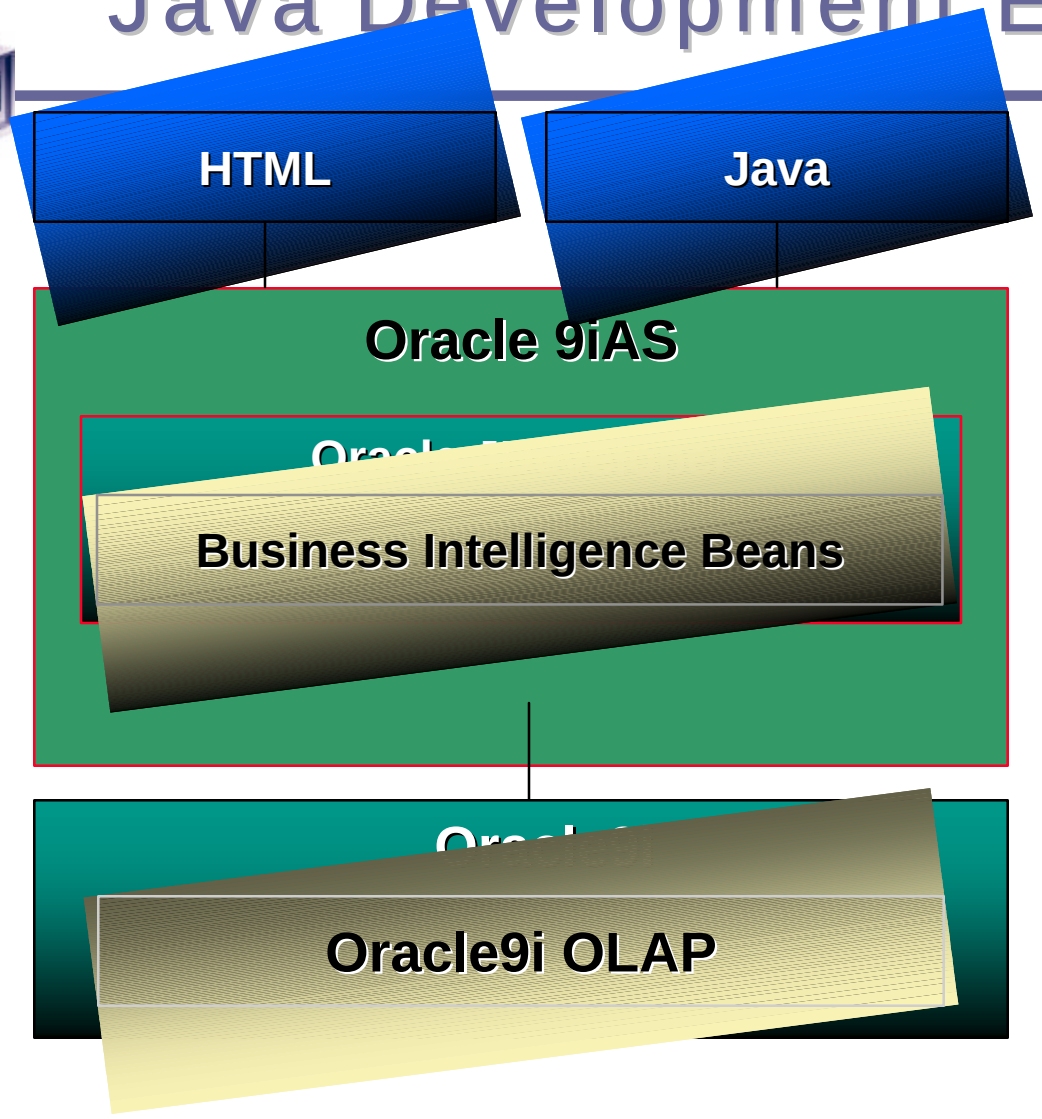
Express

```
limit geography to 'ORLANDO'
limit time to 'MAY2001'
limit product to dollars gt 1000000
```

Java OLAP API

```
Source geogSel = geography.selectValue("ORLANDO");
Source timeSel = time.selectValue("MAY2001");
Source dolByProd = dolSrc.join(geogSel).join(timeSel);
Source prodSel = product.select(dolByProd.gt(1000000));
Source dolGT1Mill =
     dolSrc.join(geogSel).join(timeSel).join(prodSel);
```

Java Development Environment

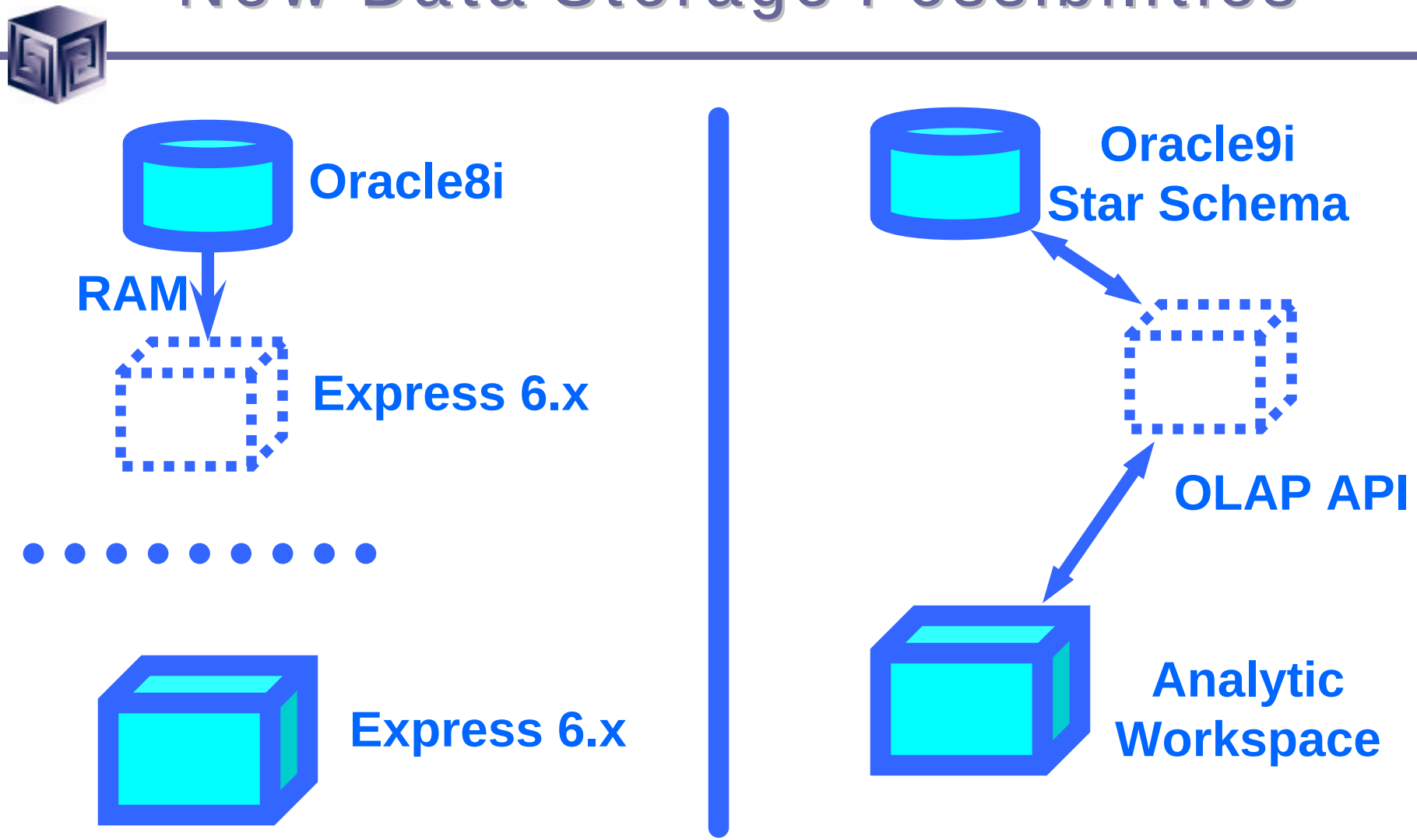


Development

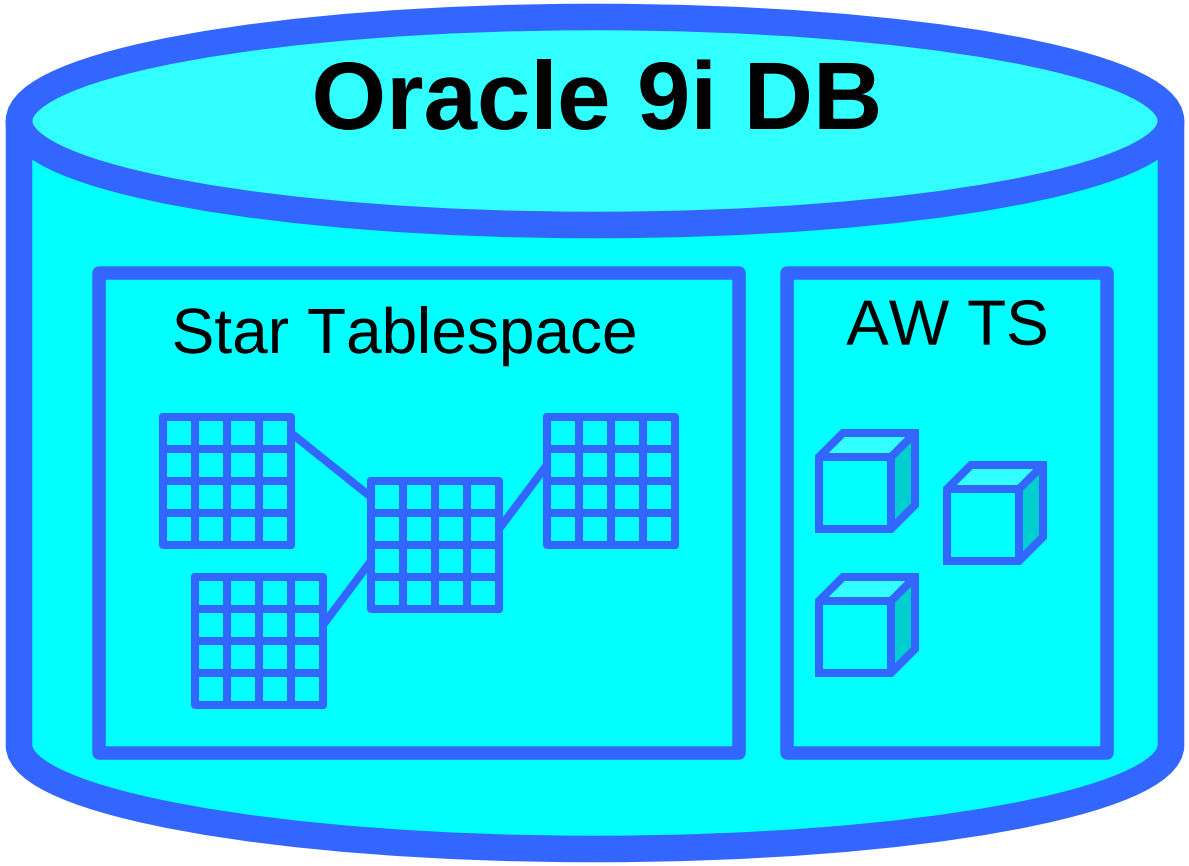
Deployment

Database Services

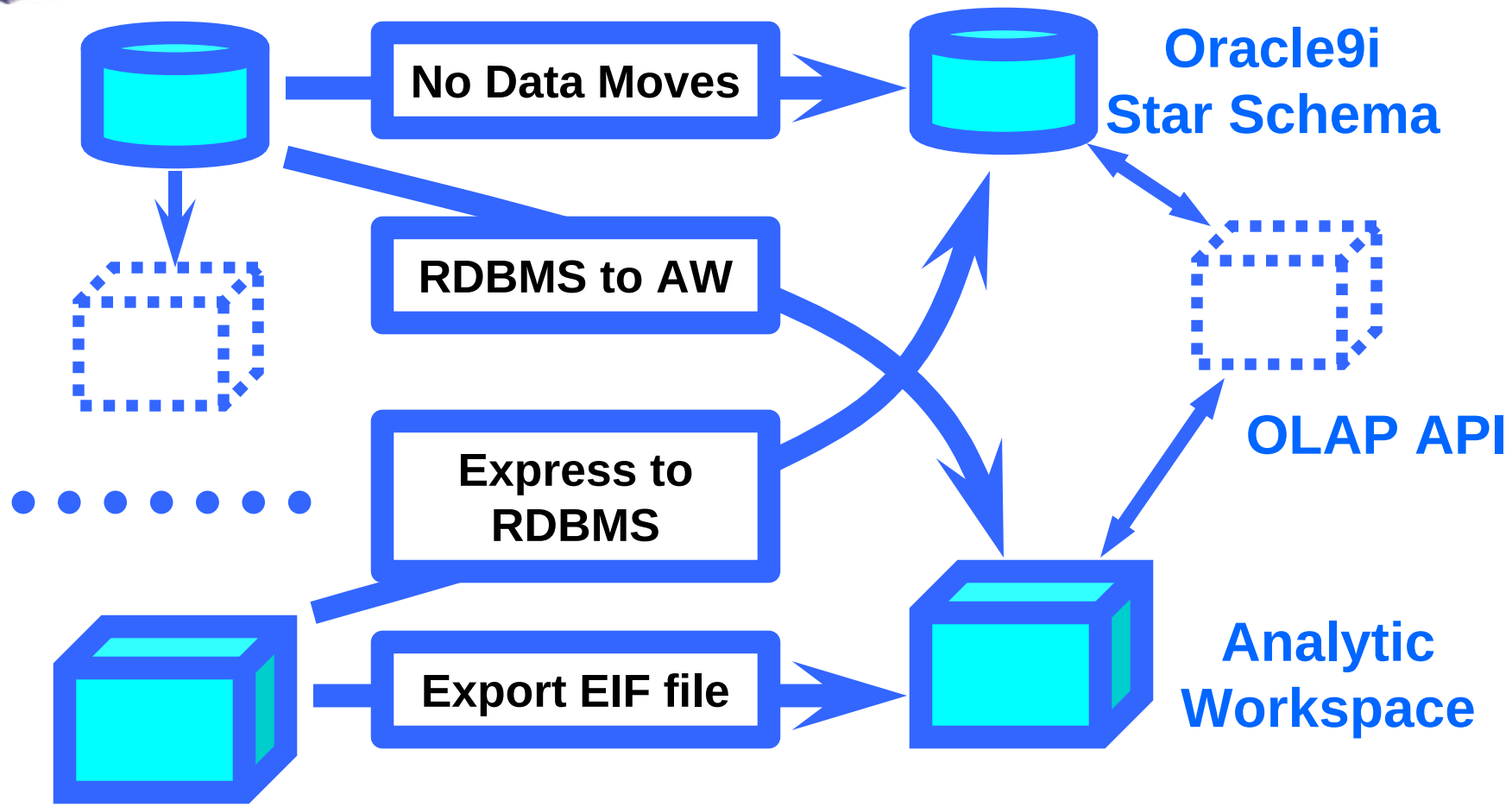
New Data Storage Possibilities



Analytic Workspaces Are Stored in Tablespaces in 9i OLAP 2

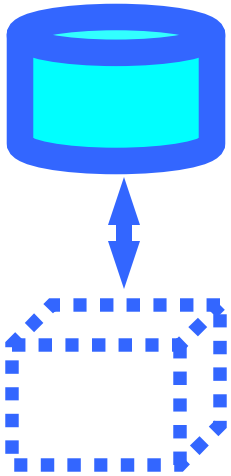


Migration Choices



Advantages of RDBMS Storage

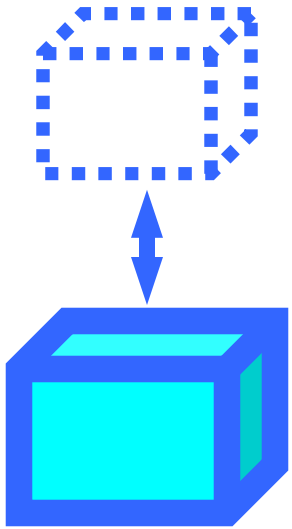
Oracle9i
Star Schema



- **Store data in familiar RDBMS**
- **Easy access to data using SQL**
- **Can use materialized views**
- **Best for read-only applications**
- **Model with OWB**
- **Data may already be in schema**

Advantages of AW Storage

- **Faster multidimensional access**
- **Personal user workspaces**
- **Best for read/write applications**
- **Best for heavier analysis**
- **OLAP DML language**



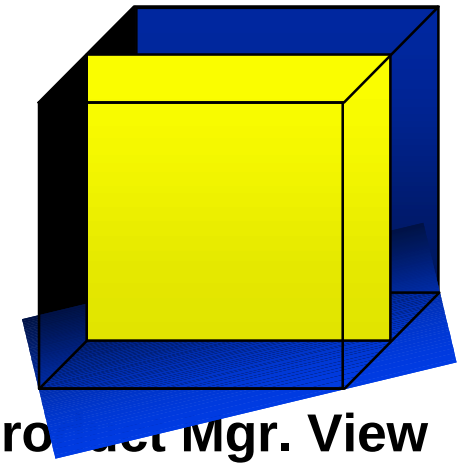
Analytic Workspace

Oracle 9i AW Tips

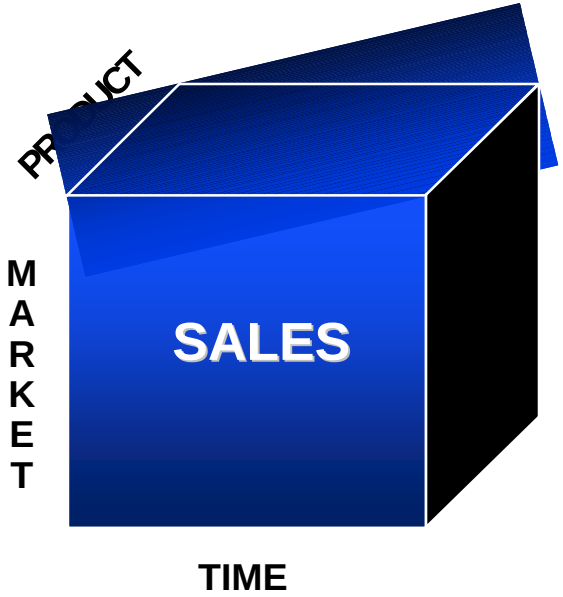
- **Use separate tablespaces for AWs**
- **Keep out of SYS tablespace**
- **Stripe across drives with multiple pathnames**
- **Make sure they autoextend**
- **Users should use TABLESPACE keyword when creating AWs**

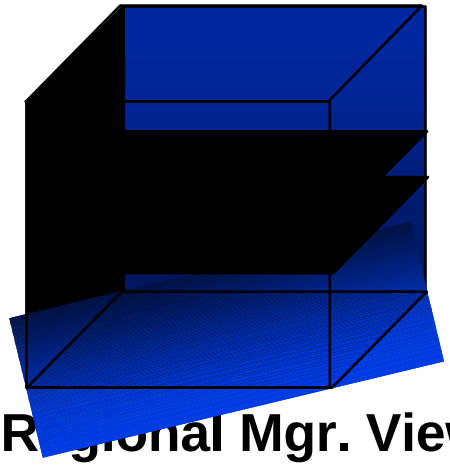
9i OLAP AW Stores Data in Cubes

Fast Flexible Access to Summarized Data

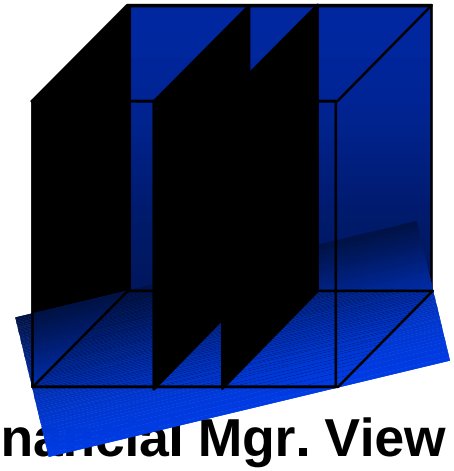


Pro[illegible]ct Mgr. View

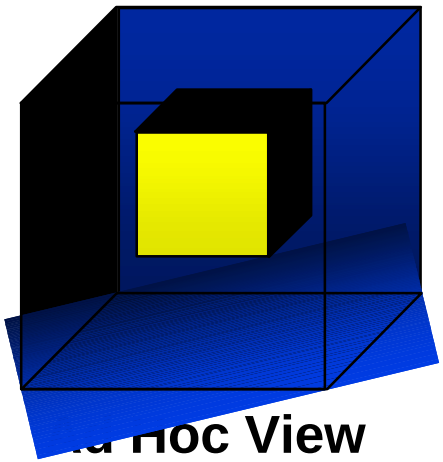




R[illegible]gional Mgr. View



Fina[illegible]cial Mgr. View



[illegible] Hoc View

What Are AW Cubes?

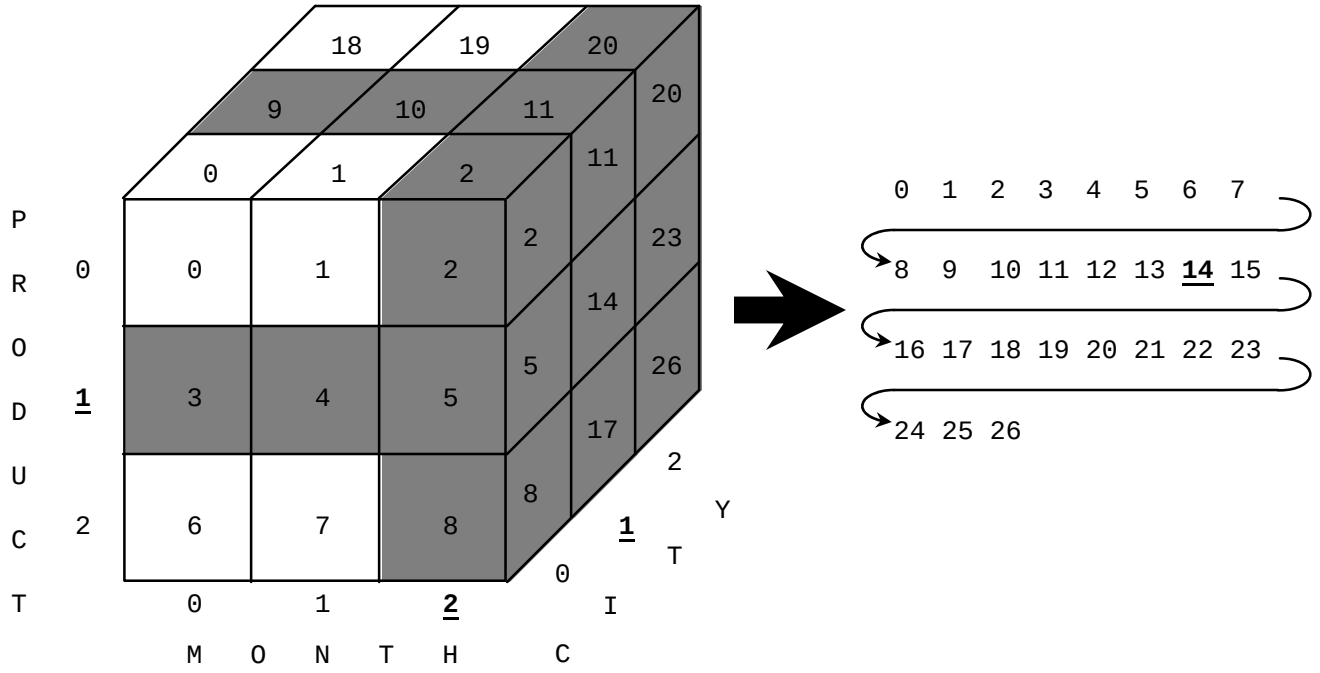
- Data stored as arrays
- Dimension values are internally integers
- Offset calculated using simple multiplication
- Offset tells exactly where to look for data
- Pages and segmentation complicate design
- Conjoints and composites handle sparsity

Calculating the cell offset is simple multiplication and addition

Formula for calculating cell offset:

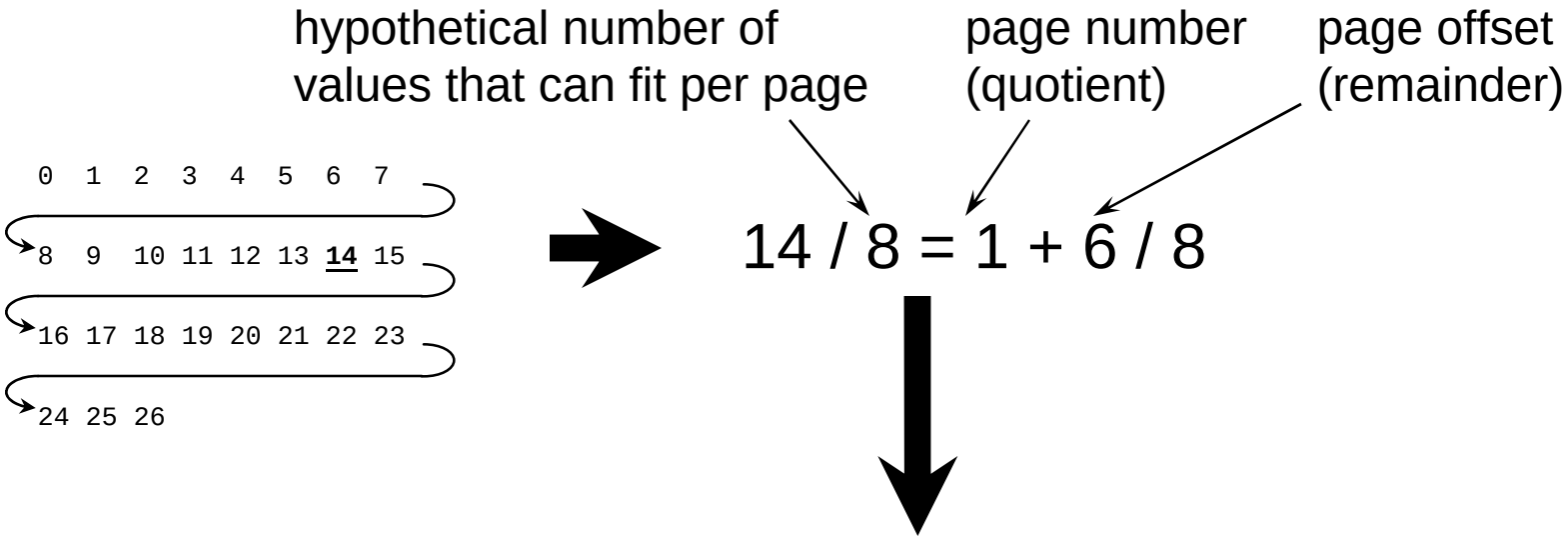
month + product * (# of months) + city*(# of months * # of products)

2 + **1** * (3) + **1** *(3 * 3) = 14



Converting to pages of data

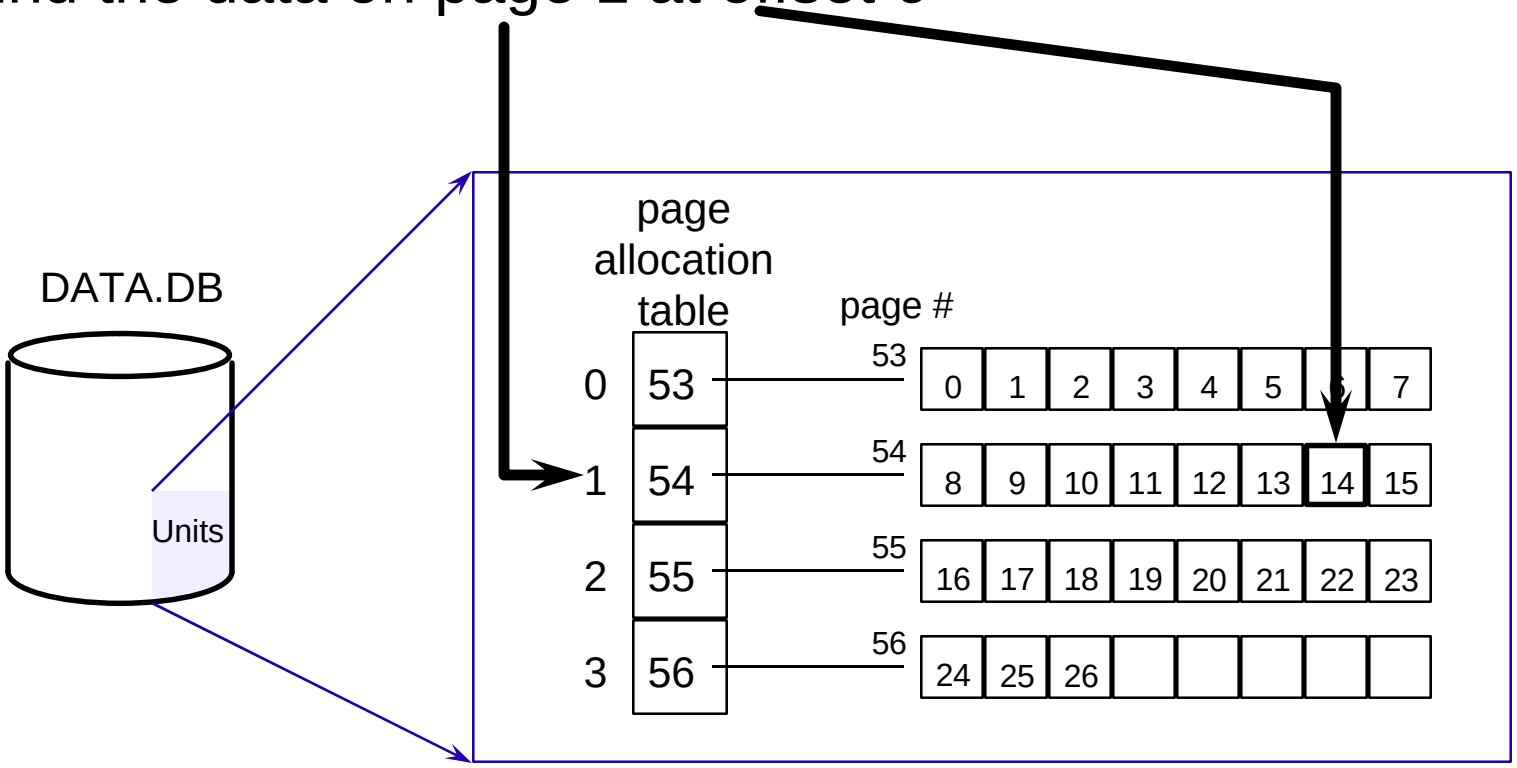
Express then converts the index to
a page number and a page offset number



Find the data on page 1 at offset 6

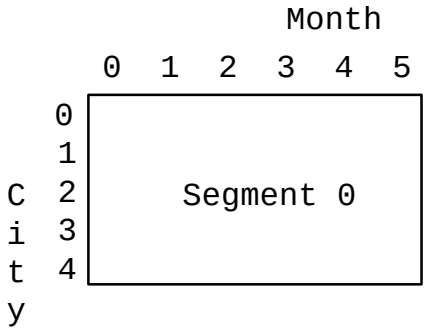
Page allocation table points to pages

Find the data on page 1 at offset 6



Segments allow for dimension maintenance

Units variable



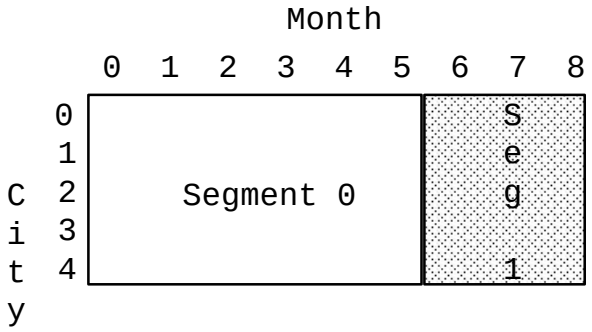
Segment table for <MONTH CITY>

Seg#	Month min	Month max	City min	City max
0	0	5	0	4

- **Segment 0 created when UNITS populated with MONTH 5, CITY 4**

Segments allow for dimension maintenance

Units variable



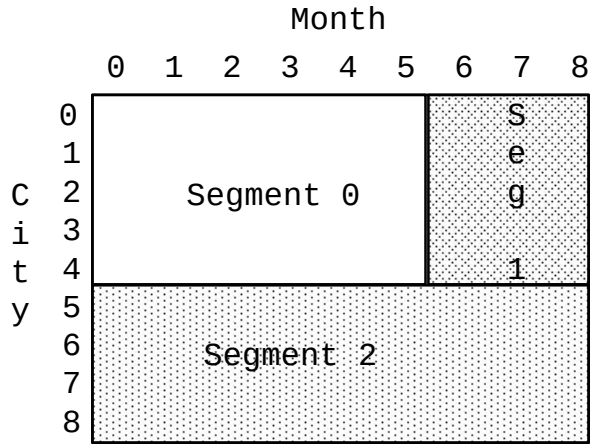
Segment table for <MONTH CITY>

Seg#	Month min	Month max	City min	City max
0	0	5	0	4
1	6	8	0	4

- **Segment 0 created when UNITS populated with MONTH 5, CITY 4**
- **Segment 1 created when UNITS populated with MONTH 8, CITY 4**

Segments allow for dimension maintenance

Units variable



Segment table for <MONTH CITY>

Seg#	Month min	Month max	City min	City max
0	0	5	0	4
1	6	8	0	4
2	0	8	5	8

- **Segment 0 created when UNITS populated with MONTH 5, CITY 4**
- **Segment 1 created when UNITS populated with MONTH 8, CITY 4**
- **Segment 2 created when UNITS populated with MONTH 8, CITY 8**

Conjoints and Composites store sparse data efficiently

Sparse Units variable

City \ Month	0	1	2	3	4	5	6	7	8
0	1				1			2	
1		4							
2			5		8				
3			2			2			
4	4								
5	3			3					
6							5		
7		9							
8									5

Conjointed Units variable

<City	Month>	Units
0	0	1
0	4	1
0	7	2
1	1	4
2	2	5
2	4	8
3	2	2
3	5	2
4	0	4
5	0	3
5	3	3
6	6	5
7	1	9
8	8	5

- **Conjoint values exist only for combinations that have data**
- **Data is stored by the conjoint dimension, so sparse data is compressed**

Executing OLAP DML From SQL

- **DBMS_AW.EXECUTE procedure**
- **DBMS_AW.INTERP function (returns a CLOB with the log from the commands)**
- **DBMS_AW.GETLOG() function gets the log from the most-recently executing OLAP DML command or program or OLAP_TABLE function**

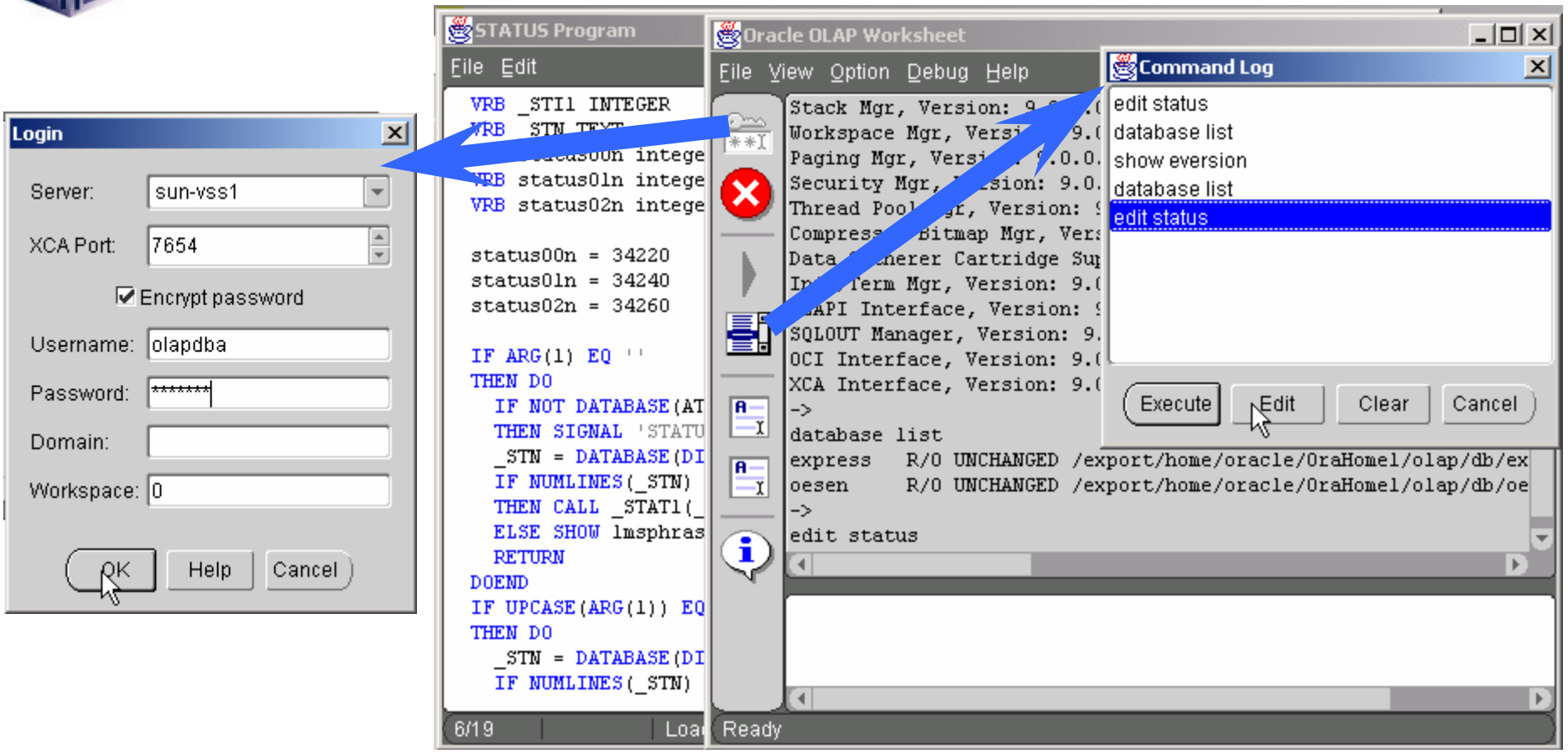
OLAP DML Changes

- **All SPL commands there except following:**
  - **Operating and file system commands limited**
  - **XCA commands and options (and shared sessions)**
  - **No interactive debugging**
  - **External Call (EXTCALL) command**
  - **SQL CONNECT command and SQL.DBMS option**
  - **SNAPI (use Java OLAP API instead)**
- **Added support for:**
  - **Parallel aggregate**
  - **Allocate**
  - **Data conversion functions, new data types**

OLAP DML Changes (continued)

- **Database is now Analytic Workspace**
  - **Stored in LOB in RDBMS (9.2)**
  - **AW create command creates AW's**
  - **Can attach multiple AW's**
  - **Non-unique object names OK now**
- **Update AND THEN COMMIT**
- **SQL PREPARE for high-speed AW->RDBMS**
- **SQL access to AW via OLAP Table Function (9.2)**

OLAP Worksheet



Terminology Map

Express	Oracle 9i OLAP
Express database	**Analytic Workspace**
Oracle Express Server	**OLAP Service**
Express Instance Manager	**OLAP Services Instance Manager**
Express Agent	**OLAP Agent**
Express SPL (or 4GL)	**OLAP DML**
RAA	**OLAP folder of OEM**
RAM	**Automatic***

* Not necessary to use RAM any more since relational data directly accessible

Terminology Map (continued)

Express	Oracle 9i OLAP
oes.key	olap.key
oesdba	olapdba
SNAPI	Java OLAP API
Oracle Express Objects	JDeveloper
Express Basic	Java
Express Administrator	None
OESCMD and Administrator command line	OLAP Worksheet

Other Presentations

400: Migrating Express Data to Oracle 9i
Dan Vlamis, Vlamis Software Solution, Inc.

Monday, April 15, 2002 Time:10:00 AM – 11:30 AM

404: Converting Express Applications to Oracle9i and BI Beans
Chris Claterbos, Vlamis Software Solutions, Inc.

Tuesday, April 16, 2002 Time:10:00 AM - 11:30 AM

412: Oracle9i OLAP - The Platform for Web-Enabled Applications
Bud Endress, Oracle Corporation

Wednesday, April 17, 2002 Time:1:30 PM – 2:30 PM

422: Using Oracle Warehouse Builder 3i and Oracle9i to Create OLAP-Ready Warehouses
Chris Claterbos, Vlamis Software Solutions, Inc.

Thursday, April 18, 2002 Time:9:45 AM – 10:45 AM

427: Oracle9i Integrated Business Intelligence
John Entenmann, Oracle Corporation

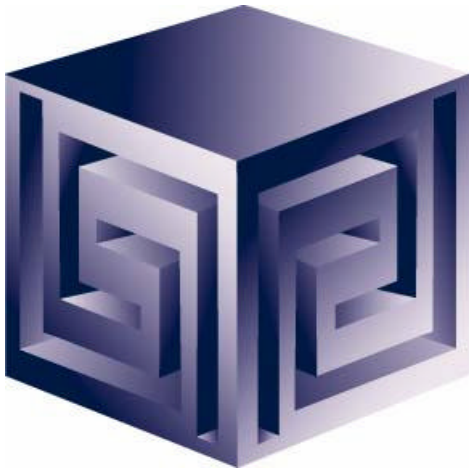
Thursday, April 18, 2002 Time:11:00 AM – 12:00 PM

Oracle 9i OLAP—How Does it Really Work?

presented at

IOUG Live! 2002

paper number 409



Presented by:

Dan Vlamis (dvlamis@vlamis.com)

Vlamis Software Solutions, Inc.
(816) 781-2880

http://www.vlamis.com