
ORACLE®

Oracle OLAP Option

Fred Louis

Solution Architect, Ohio Valley

ORACLE

ORACLE FUSION MIDDLEWARE

ORACLE
BUSINESS INTELLIGENCE

COLLABORATIVE ENTERPRISE PORTAL

Portals, Collaboration, Mobile, Desktop, Search

DEVELOPMENT TOOLS

Modeling, Development Tools, Frameworks

COMPOSITION & PROCESS ORCHESTRATION

BPM, ESB, B2B

INFORMATION AGGREGATION & ANALYSIS

ETL, Hubs, Content Mgmt., BI, BAM

MANAGEMENT

Systems Management

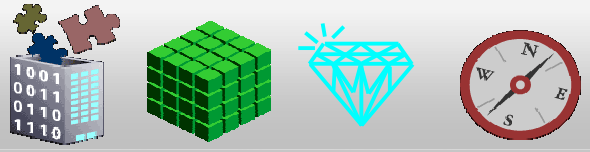
ENTERPRISE APPLICATION SERVER

J2EE, WS-*, Events, Metadata, Registry

SECURITY

Identity, Services Management

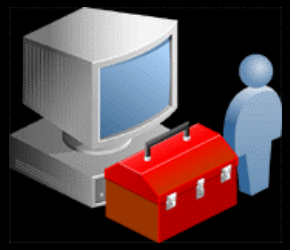
BI/DW PLATFORM



Data Warehousing, OLAP, Data Mining, Spatial

ORACLE

Reporting Spectrum



Production Reporting

End-User Reporting

Ad-hoc Query & Analysis

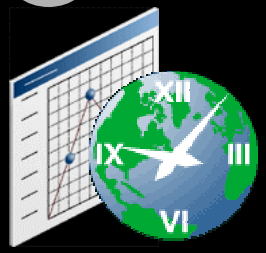
Reporting Spectrum

BAM

XML Publisher

Oracle Reports

Oracle BI



Real-time Data

Historical Data

OLAP at Oracle

- Over thirty years of innovation yields a complete and compelling OLAP platform
  - Express, the first multidimensional database
  - Oracle 9iR2, the first (and only) relational-multidimensional database
  - Oracle 10g
    - The first (and only) Grid capable OLAP platform
    - All new administration
    - All new data access tools
    - All new applications

The role of OLAP in the Database …

Why Multidimensional?

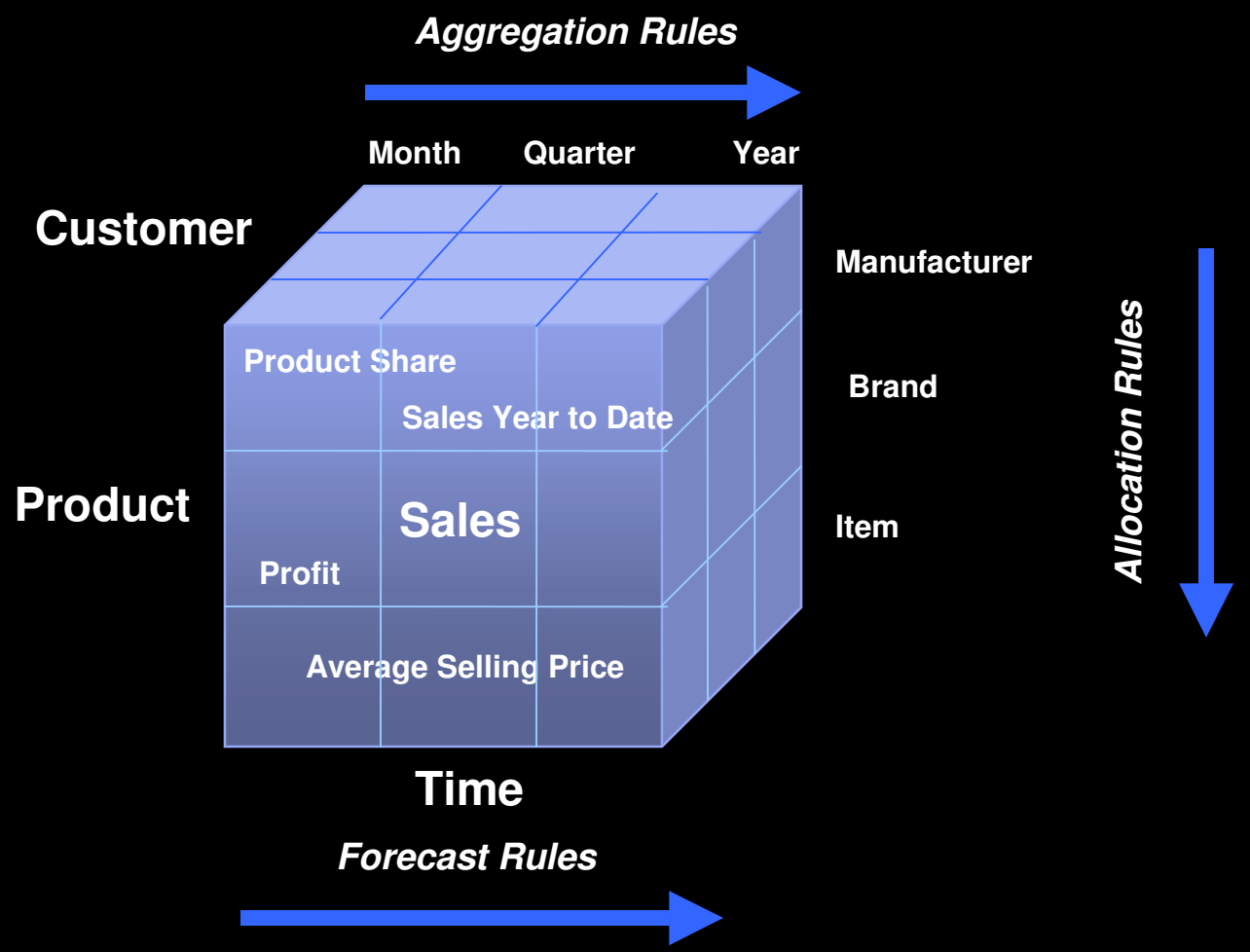
- Calculations
  - Models, forecasts, statistics, custom functions, etc.
- Multidimensional models
  - End user model – provides structure and context
  - Physical model - facilitates ease of expression
- Transaction Model
  - Session isolated, read repeatable
  - Supports what-if/planning applications
- Ad-Hoc Query Optimization
  - Efficient build and solve processes
  - Uniform performance across entire logical model
  - Excellent runtime calculation performance

Dimensional Model

- Promotes ad-hoc navigation and calculation definition
  - Easily understood by end users
    - "Sales by product and customer over time"
  - Embedded business rules
    - Users don't need to understand how all data is calculated
  - Provides context for query and calculation definition
    - Users don't need to understand the physical model

Dimensional Model

Provides both Structure and Business Rules



Transaction Model

- Session isolated, read repeatable transaction model
  - Session level DDL
  - Session level DML
- Supports applications such as budgeting and demand planning

Ad-Hoc Query Optimization

- Predictable query environment
  - Predefined reports
  - Predefined calculations
  - Less exploration of data
- Ad-hoc query environment
  - Users define reports
  - Users access any data
  - Users define calculations
  - More users amplify this effect

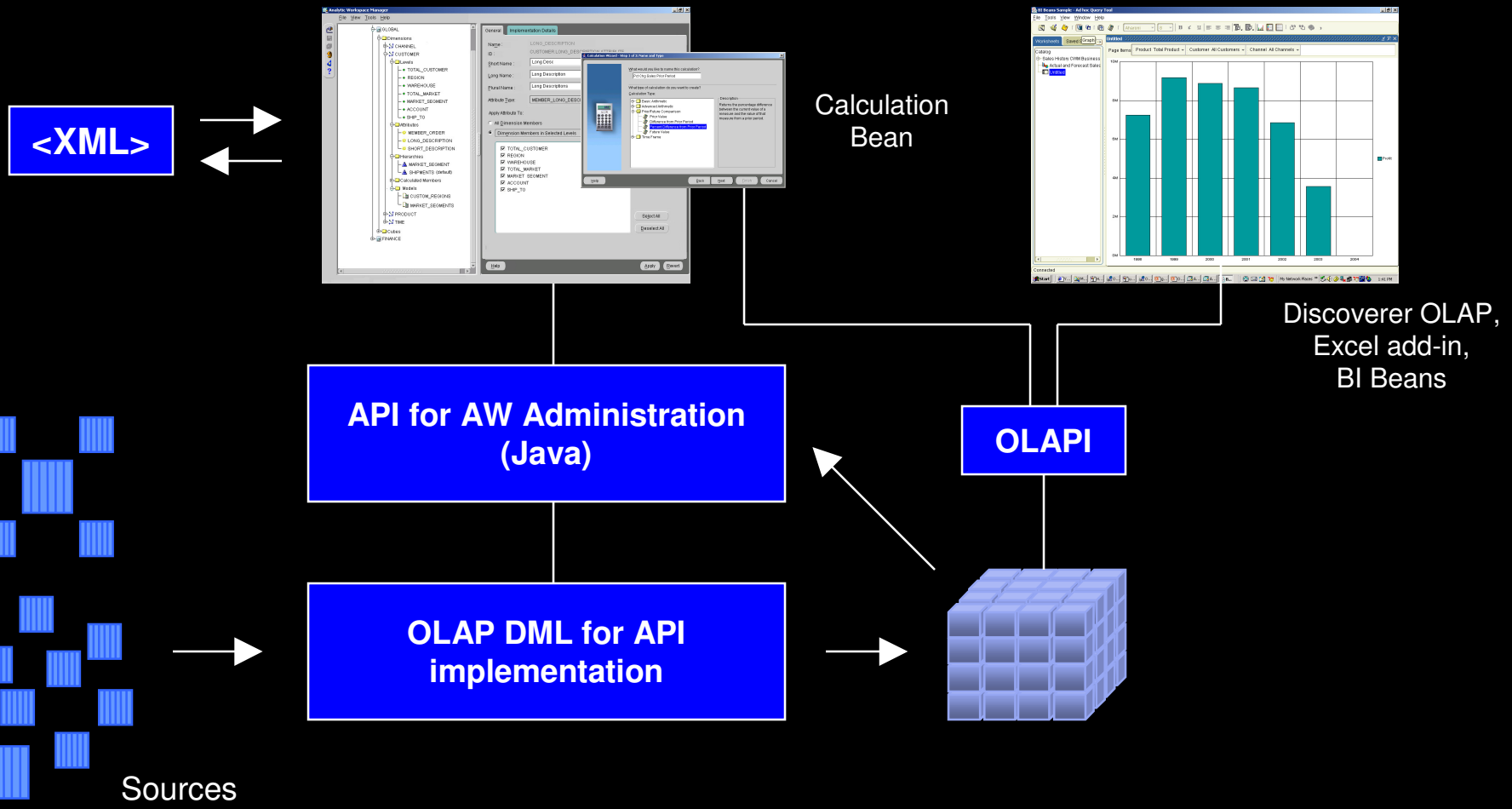
Ad-Hoc Query Optimization

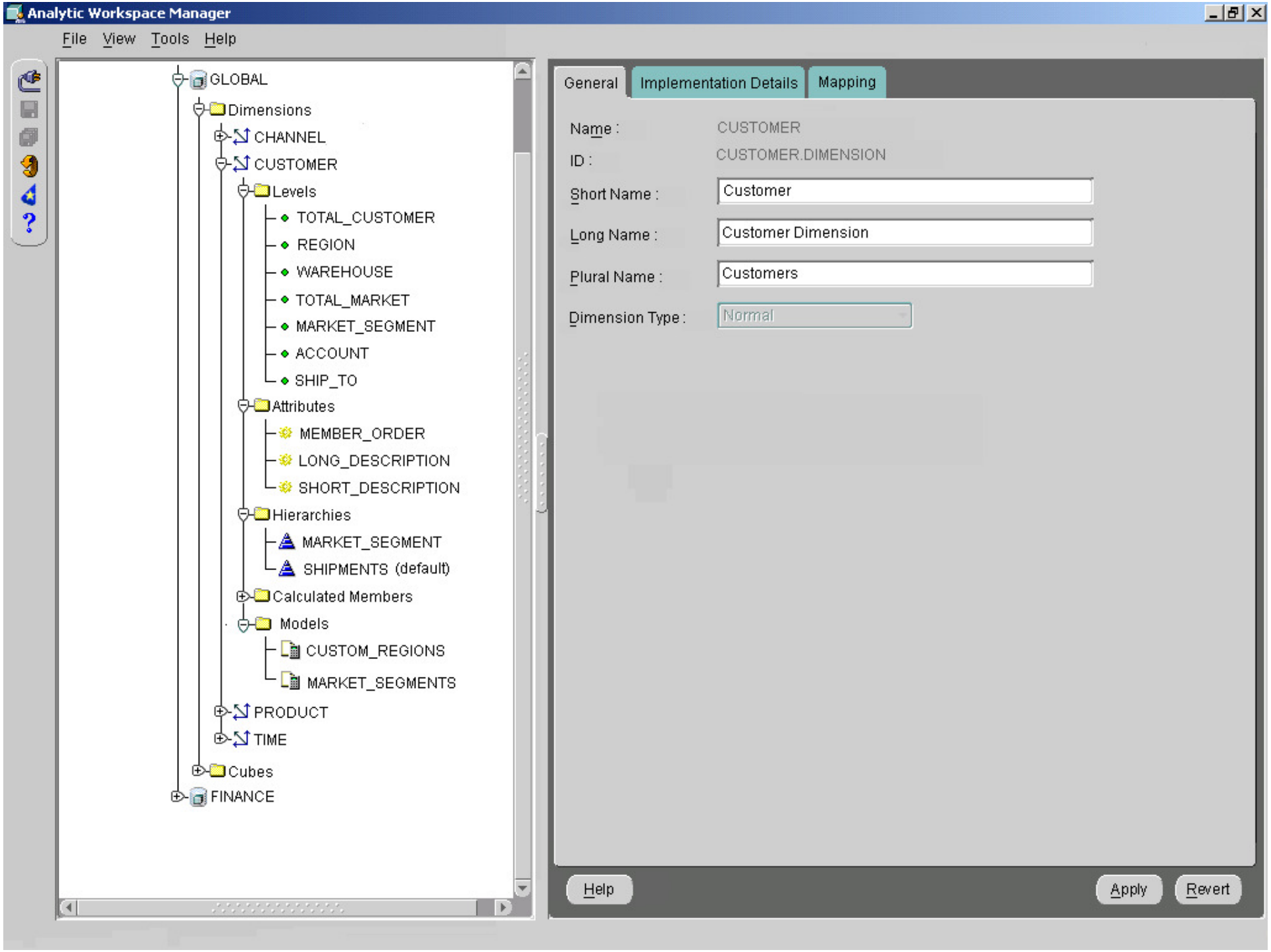
- Multidimensional data types
  - Array based measure storage
  - Measures are prejoined to dimensions
  - Measures share dimensions
  - Optimizations for sparse data
  - Summary management in multidimensional engine
  - Computational scalability
  - Partitioning and parallel processing

How do we get started?

Analytic Workspace Manger and Administration …

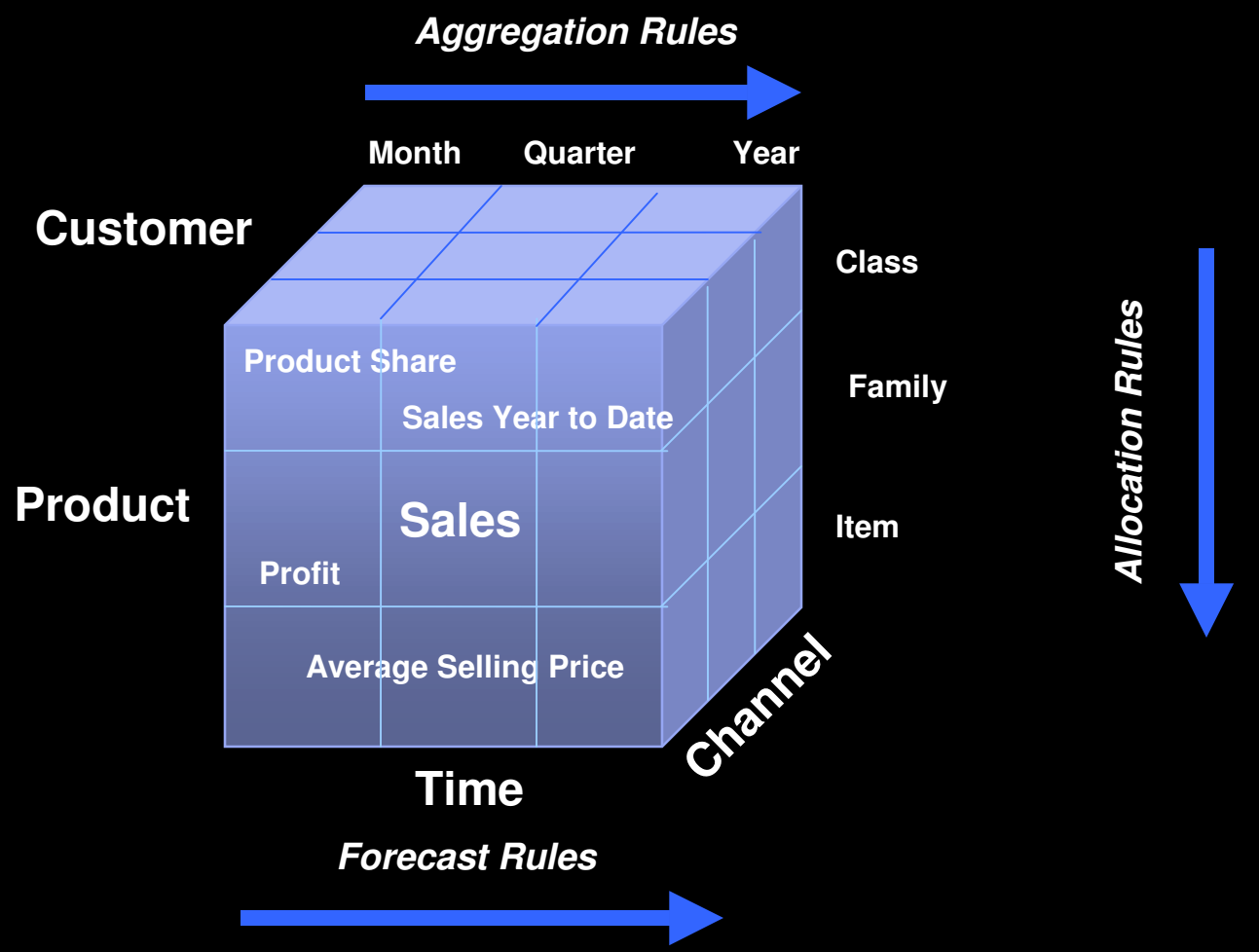
Architecture



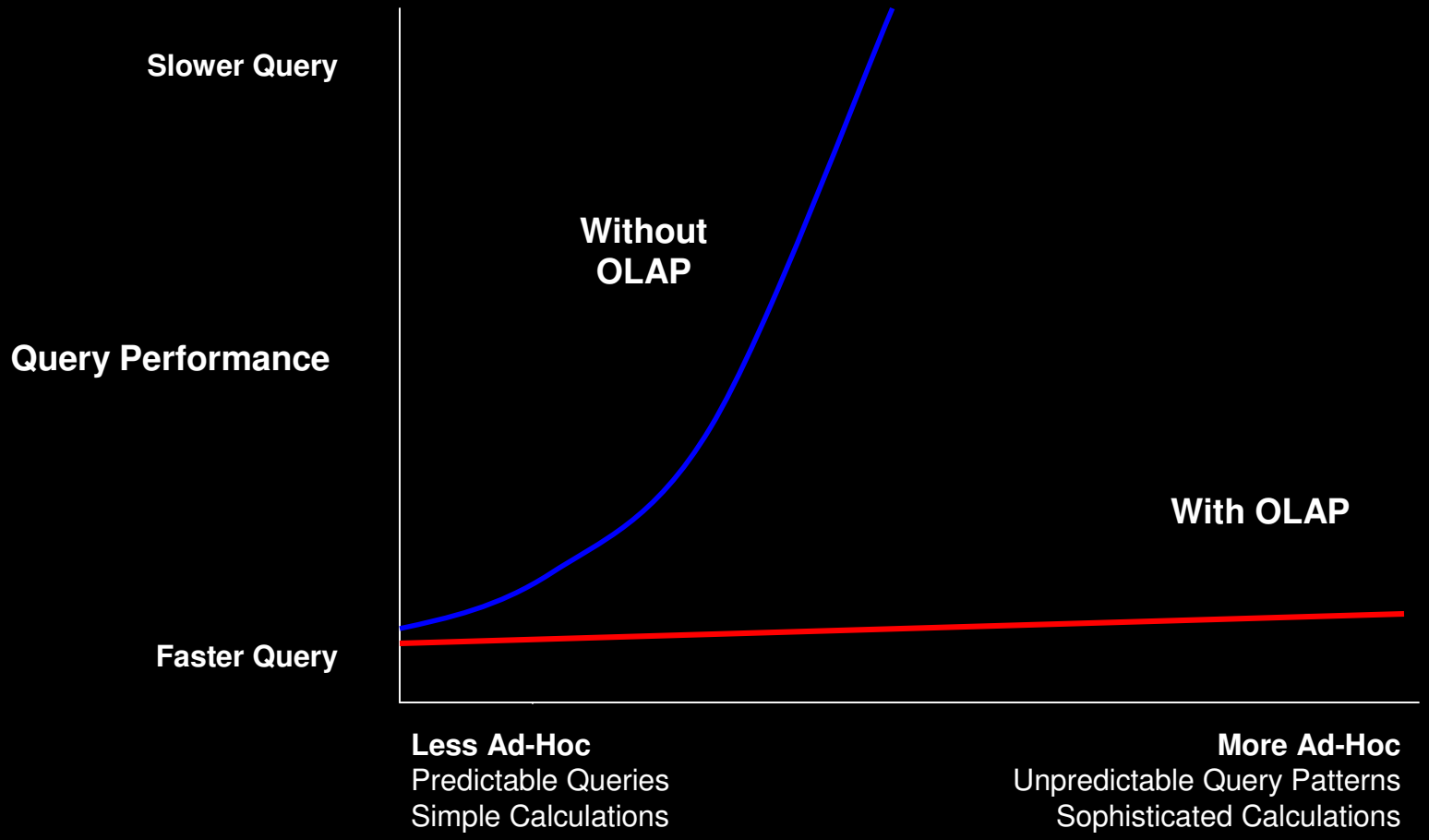

Analytic Workspace Manager
File View Tools Help
GLOBAL
Dimensions
CHANNEL
CUSTOMER
Levels
TOTAL_CUSTOMER
REGION
WAREHOUSE
TOTAL_MARKET
MARKET_SEGMENT
ACCOUNT
SHIP_TO
Attributes
MEMBER_ORDER
LONG_DESCRIPTION
SHORT_DESCRIPTION
Hierarchies
MARKET_SEGMENT
SHIPMENTS (default)
Calculated Members
Models
CUSTOM_REGIONS
MARKET_SEGMENTS
PRODUCT
TIME
Cubes
FINANCE
General
Implementation Details
Mapping
Name : CUSTOMER
ID : CUSTOMER.DIMENSION
Short Name : Customer
Long Name : Customer Dimension
Plural Name : Customers
Dimension Type : Normal
Help
Apply
Revert

Dimensional Model

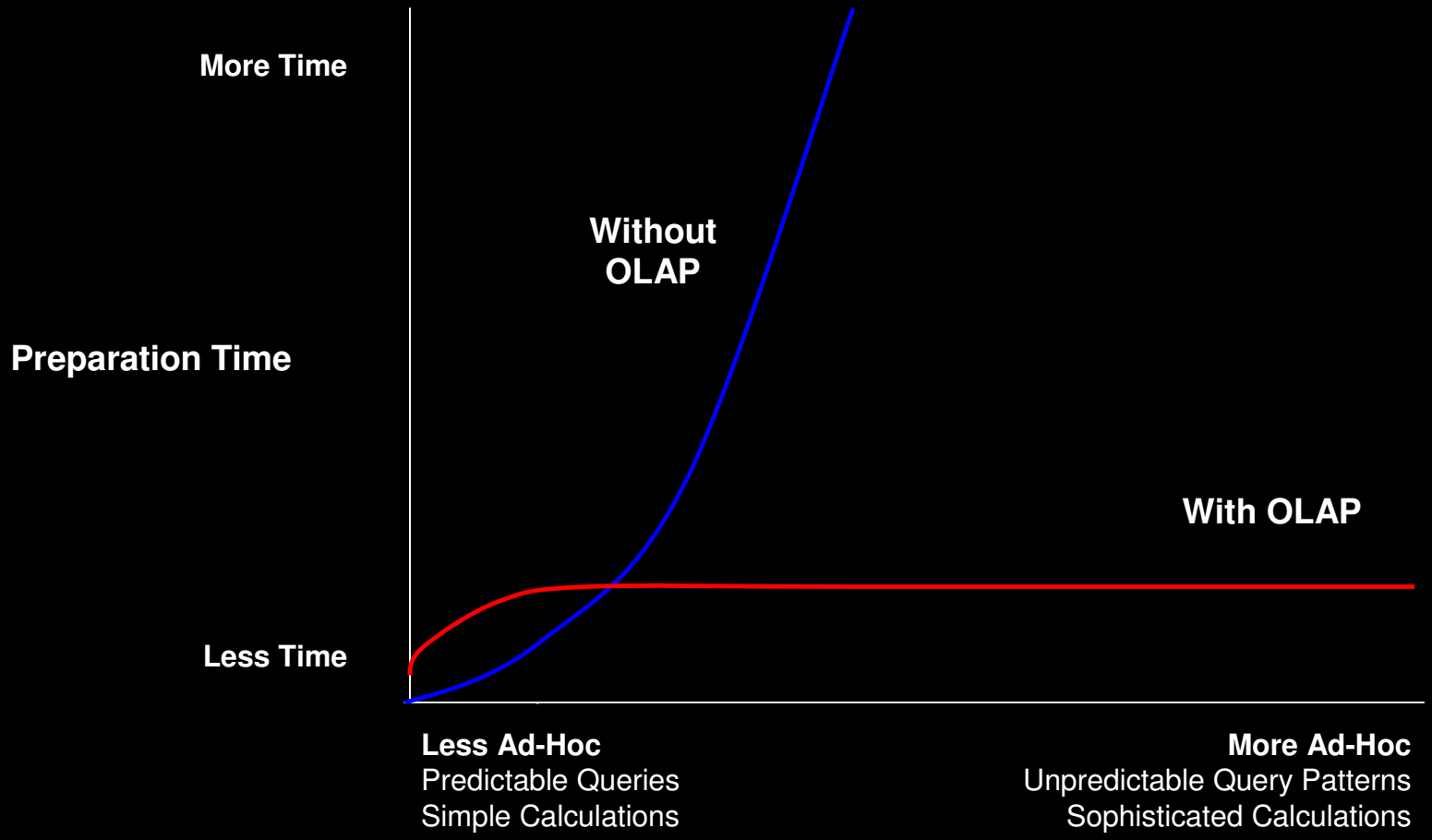
Provides both Structure and Business Rules



Query Performance



Time To Prepare Data for Query



Fragmentation

- Types of fragmentation
  - Subject specific data marts
    - Sales, finance, human resources, etc.
  - Geographically fragmented data
    - Americas, Europe, Asia Pacific, etc.
  - Database technology fragmentation
    - Relational and multidimensional
  - Tools induced fragmentation
    - Separate tools for relational and multidimensional databases

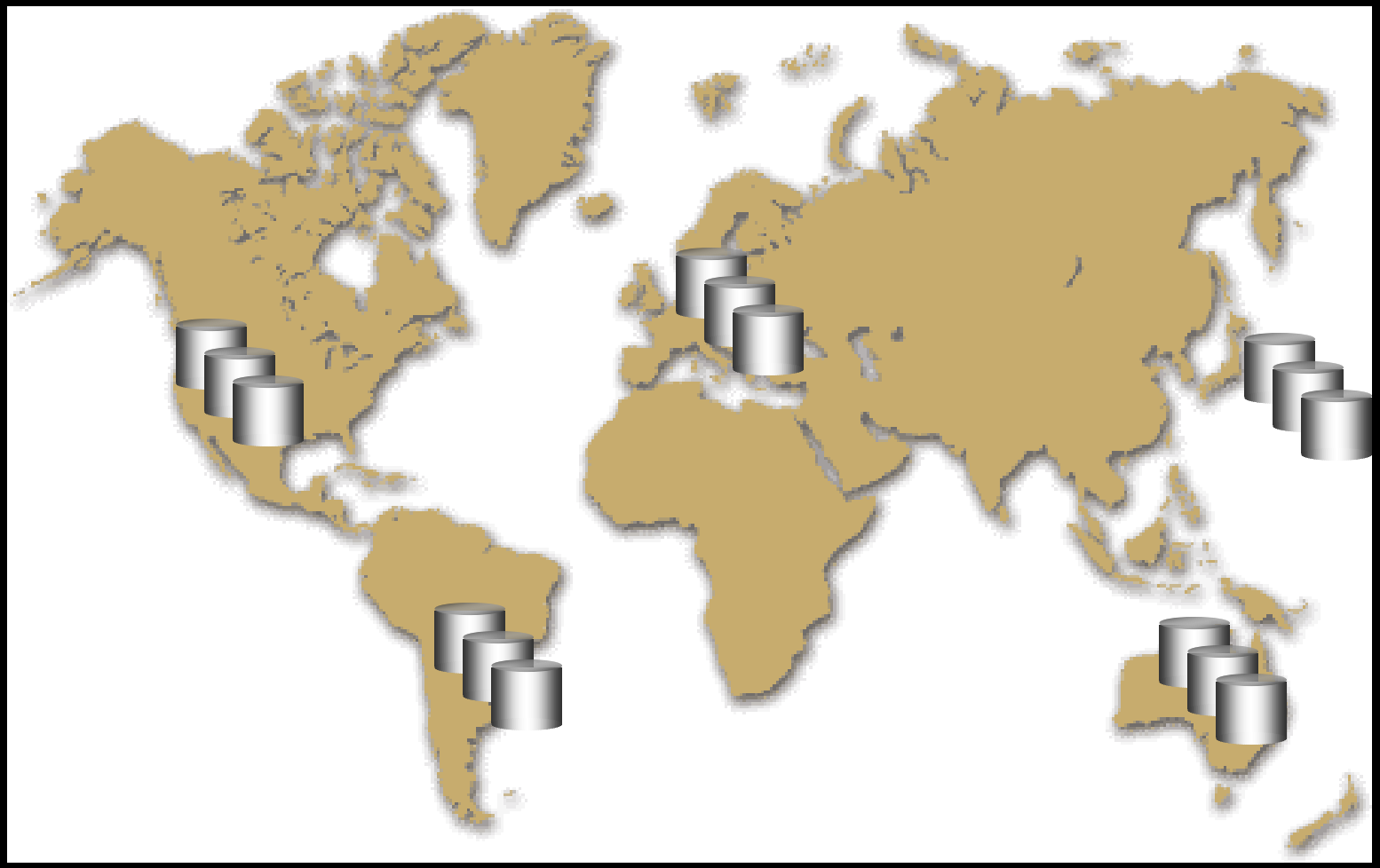
Fragmentation

- What if you had …
  - Five subject areas
  - Five geographic regions
  - Relational and multidimensional analysis of each
  - Separate tools for both relational and multidimensional databases

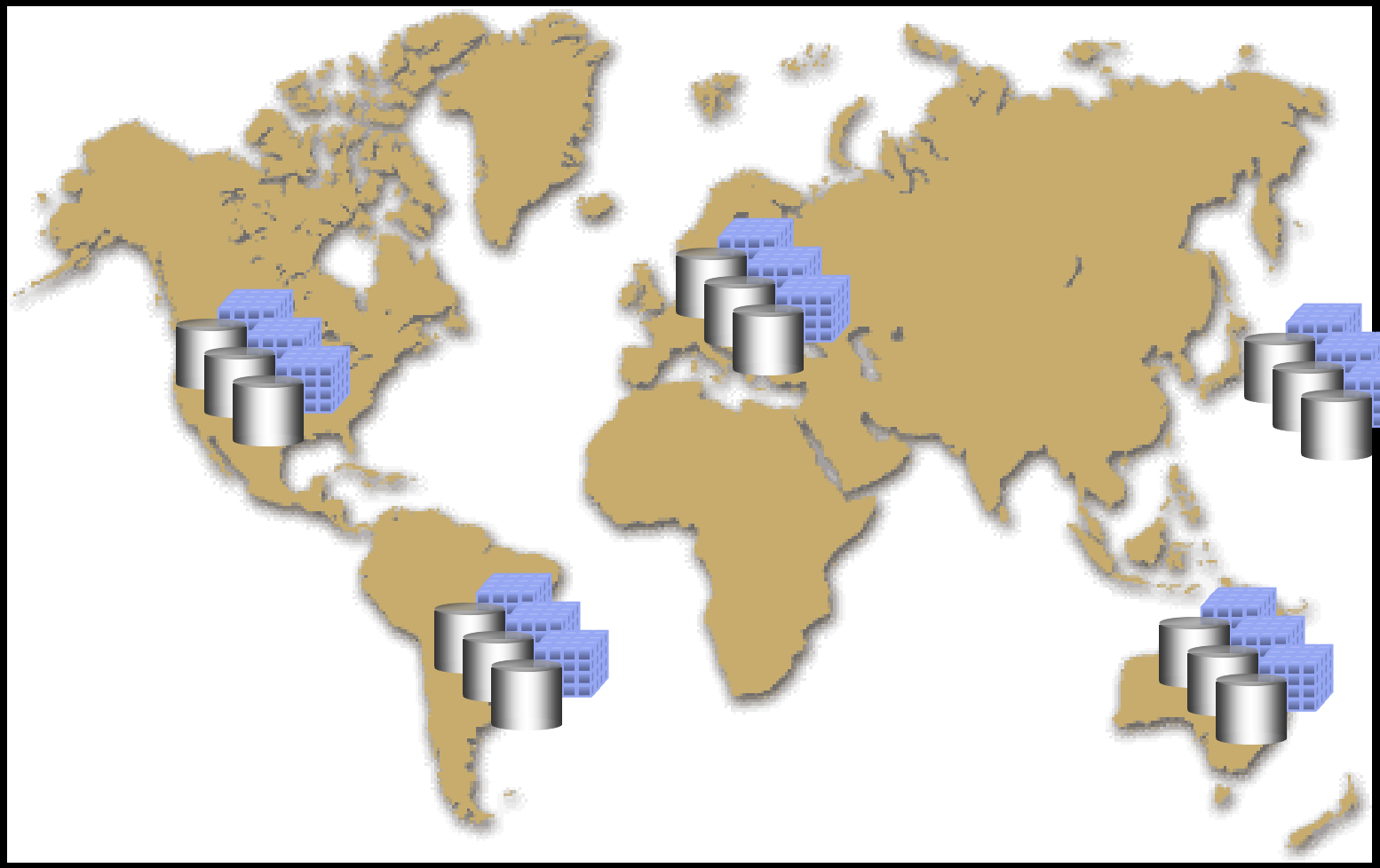
Fragmentation

- You would have …
  - 50 separate databases to manage
  - Latency and synchronization problems
  - Business rules in each database and in each tool
- You would not have a single, unified view of the enterprise with
  - A single version of the data
  - Consistency of business rules

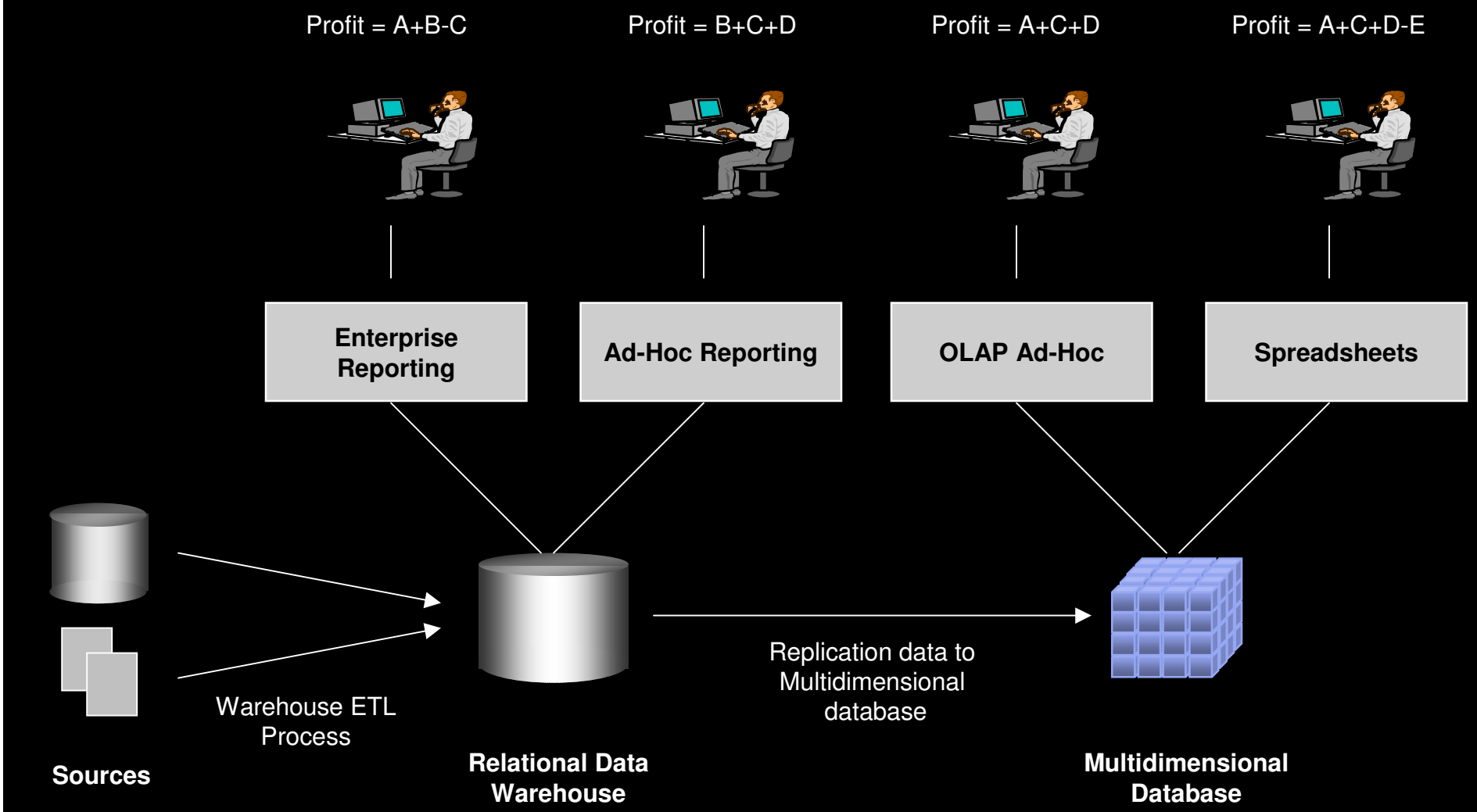
Geographic Fragmentation



Technology Fragmentation

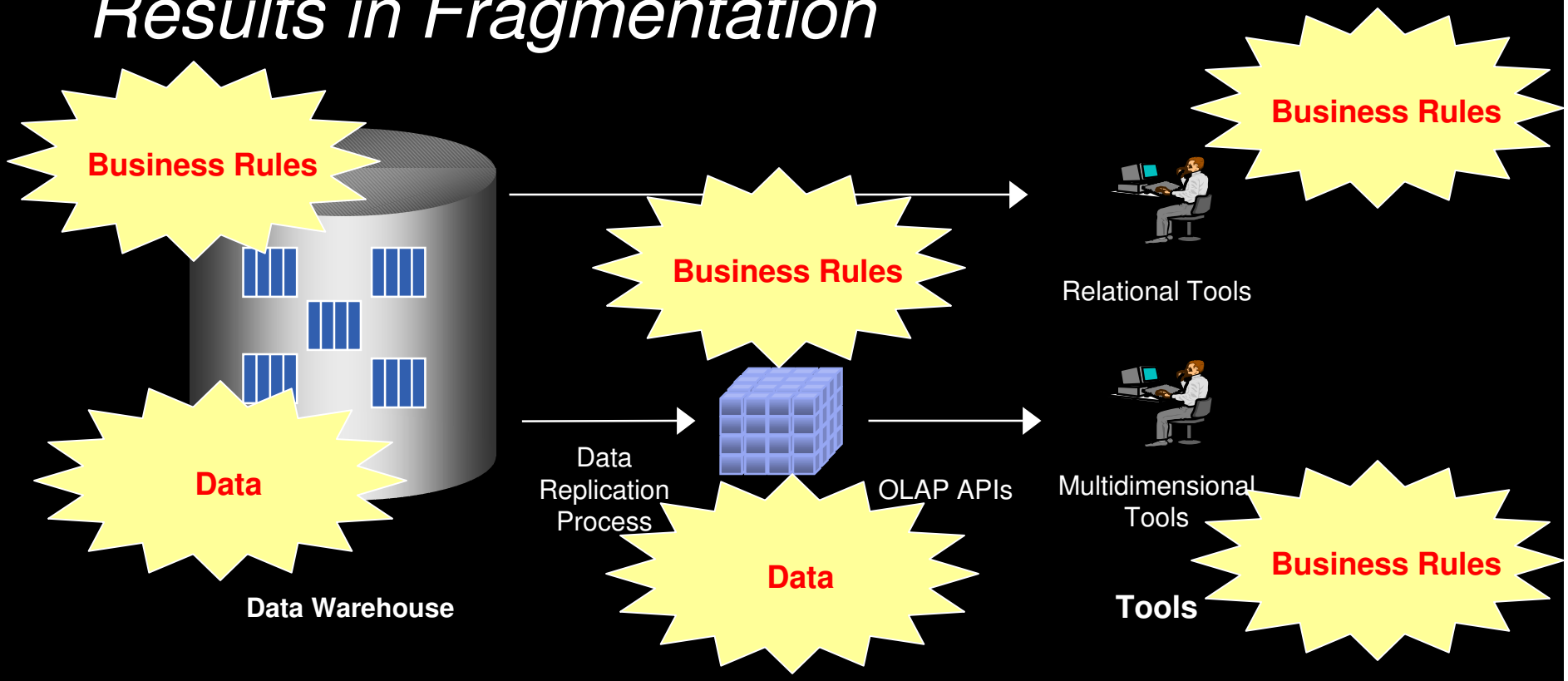


Tools Fragmentation



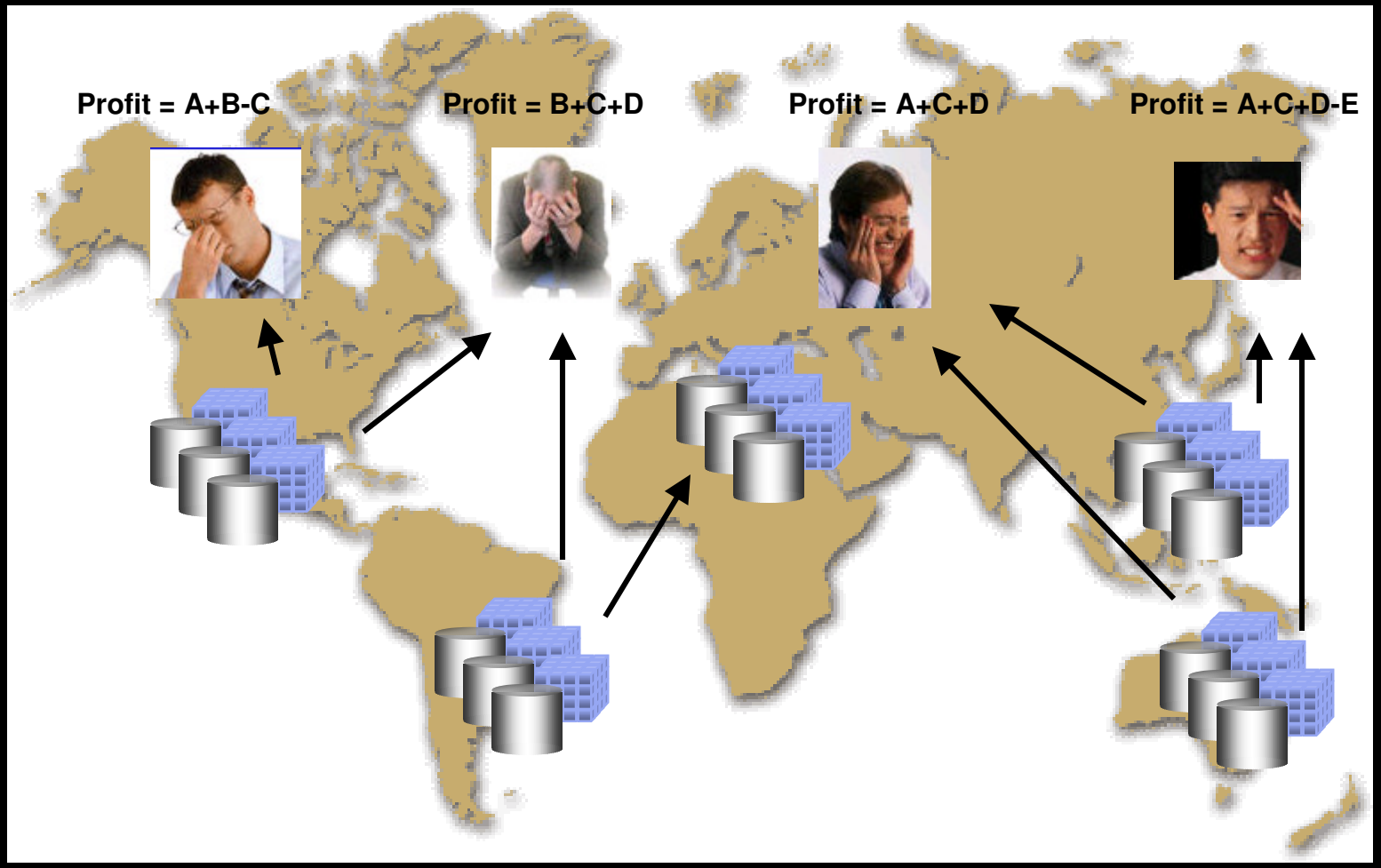
Traditional BI Platforms

Results in Fragmentation



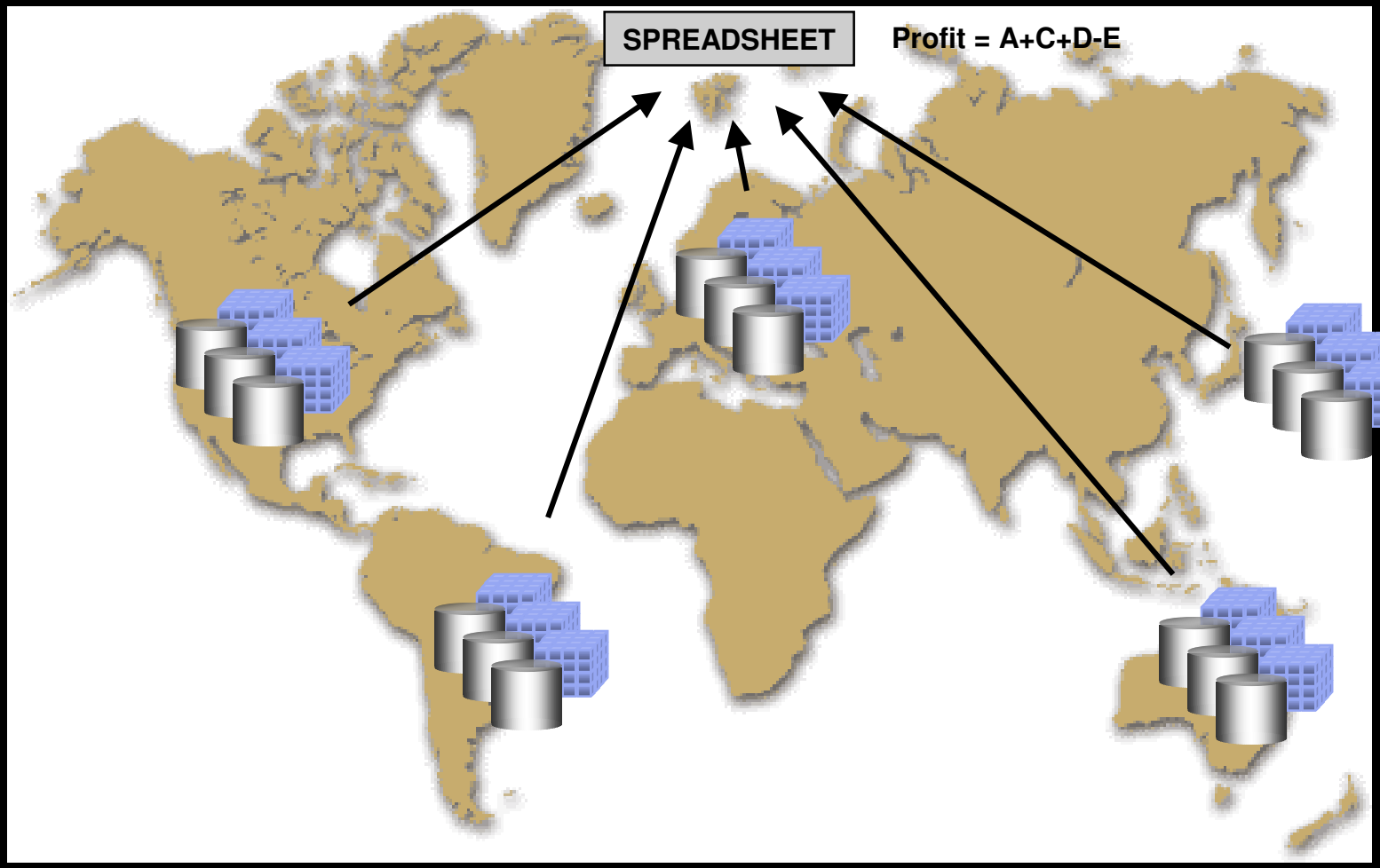
Stand-alone OLAP

Fragmented data and business rules



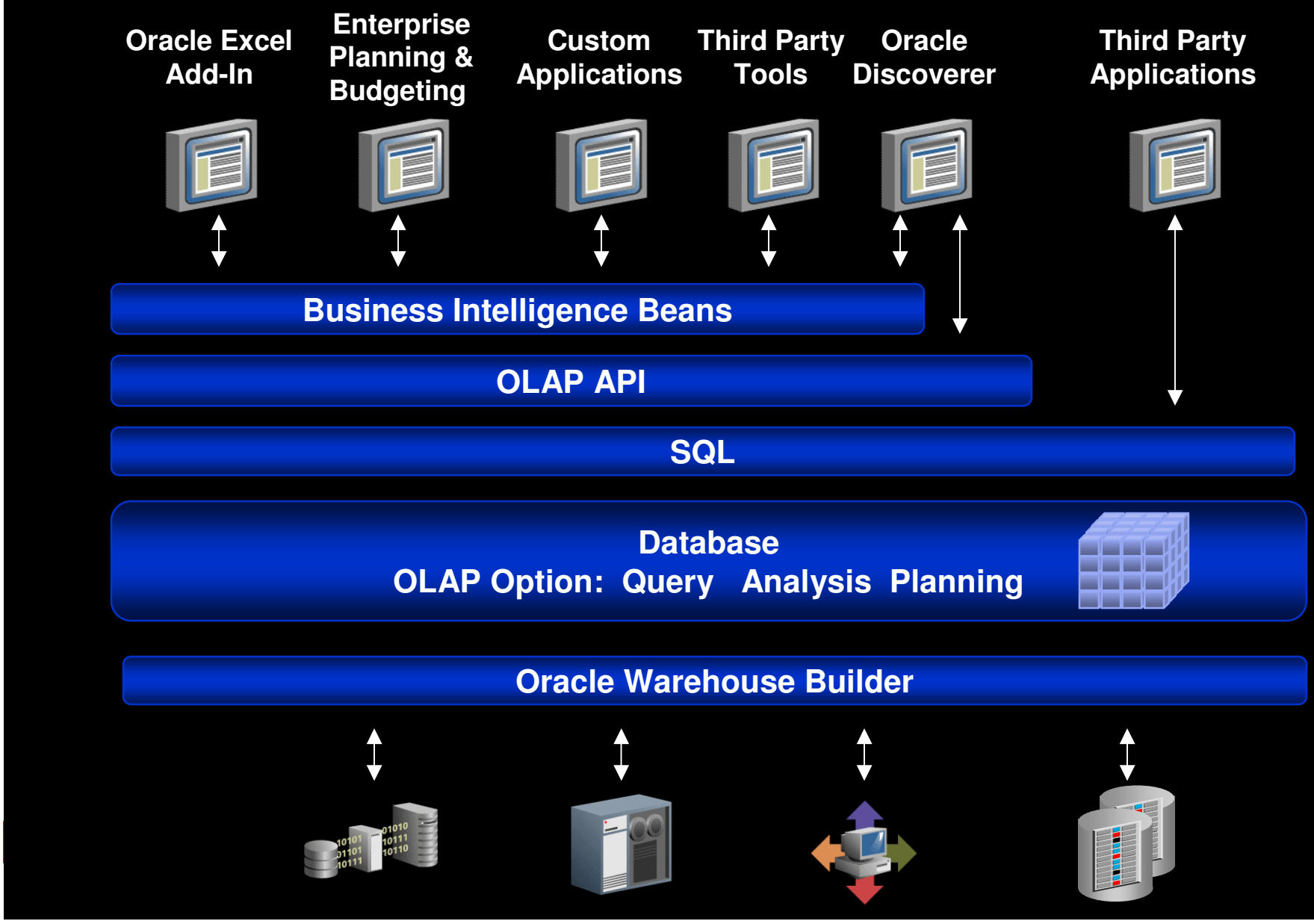
Traditional BI Platforms

Results in spreadsheet-level integration



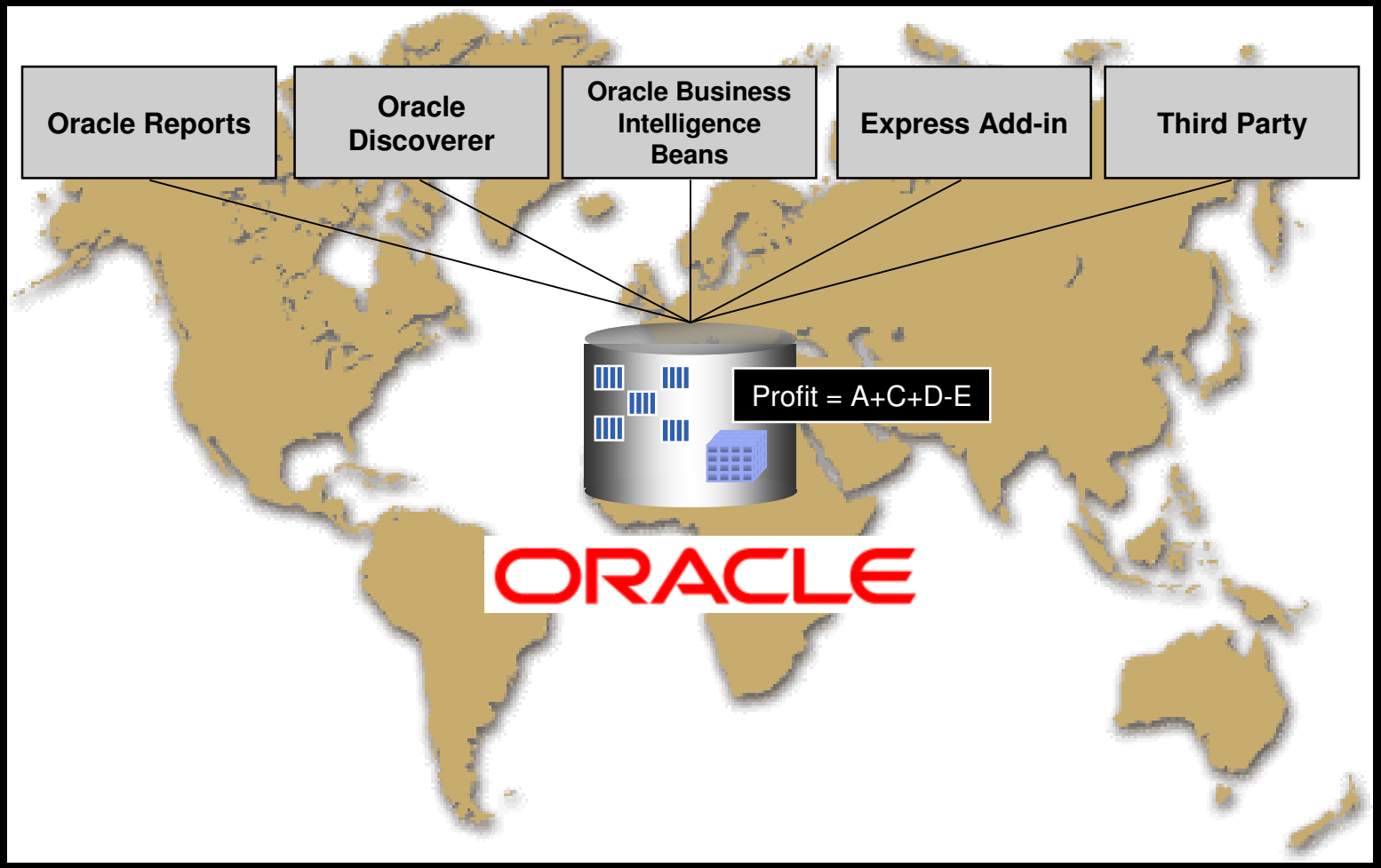
Integrated RDBMS-MDDS …

OLAP Platform



Integrated RDBMS-MDDS

Open access and consistency



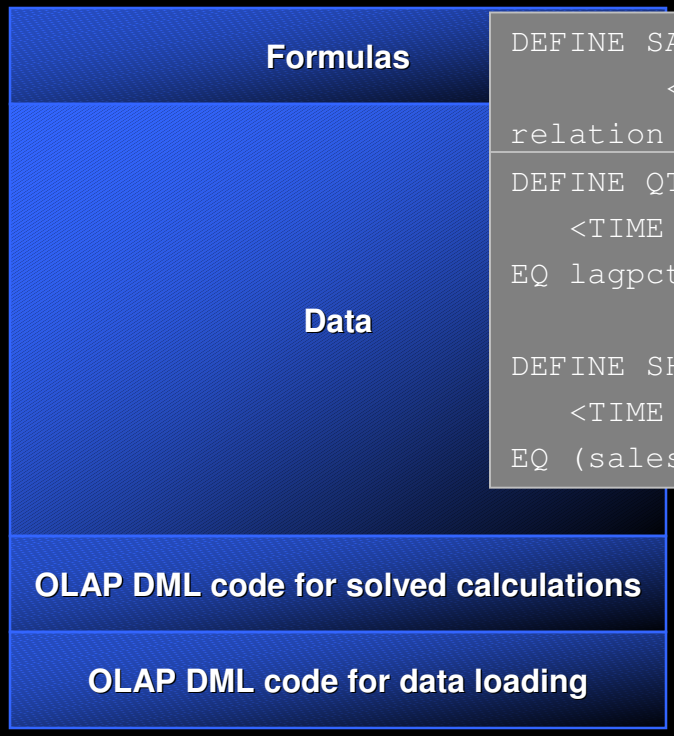
What is an Analytic Workspace?

Analytic Workspace

- Container for collections of multidimensional data types
  - Usually organized by subject matter
    - Sales, marketing, finance, HR, etc.
  - Contains one or many ‘cubes’
  - Can contain data, formulas, stored procedures, etc.
  - Stored in Oracle data files
  - Determines the scope of an OLAP transaction

Analytic Workspaces

Typical contents of an analytic workspace




```
DEFINE SALES_AGGMAP AGGMAP <TIME CUST_PRO_DEMO_CHAN_PROM
        <CUSTOMER PRODUCT DEMOGRAPHIC CHANNEL PROMOTION>>
relation parentrel_time precompute
```

```
DEFINE QTY_PCTCHG_PP FORMULA DECIMAL
   <TIME PROMOTION CHANNEL DEMOGRAPHIC PRODUCT CUSTOMER>
EQ lagpct(quantity,1,time,nostatus)*100

DEFINE SHARE_SALES_CHAN FORMULA DECIMAL
   <TIME PROMOTION CHANNEL DEMOGRAPHIC PRODUCT CUSTOMER>
EQ (sales/sales(channel 'Total')) * 100
```

gations, allocations,

```
SQL IMPORT SELECT SALES FROM
SALES_FACT INO SALES_DATA
```

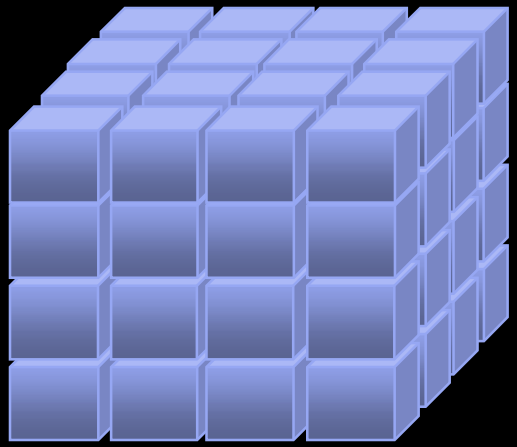
OLAP DML programming code for loading data from relational tables and flat files

Analytic Workspaces

Concept of the “solved cube”

```
Select time_id, product_id,
customer_id, sales, product_share
from sales_view
where product_level = 'BRAND'
and month_level = 'MONTH'
and customer_id = 'TOTAL CUSTOMER';
```

Client applications asks for data without expressing the calculation rules



Runtime Calculation Definitions

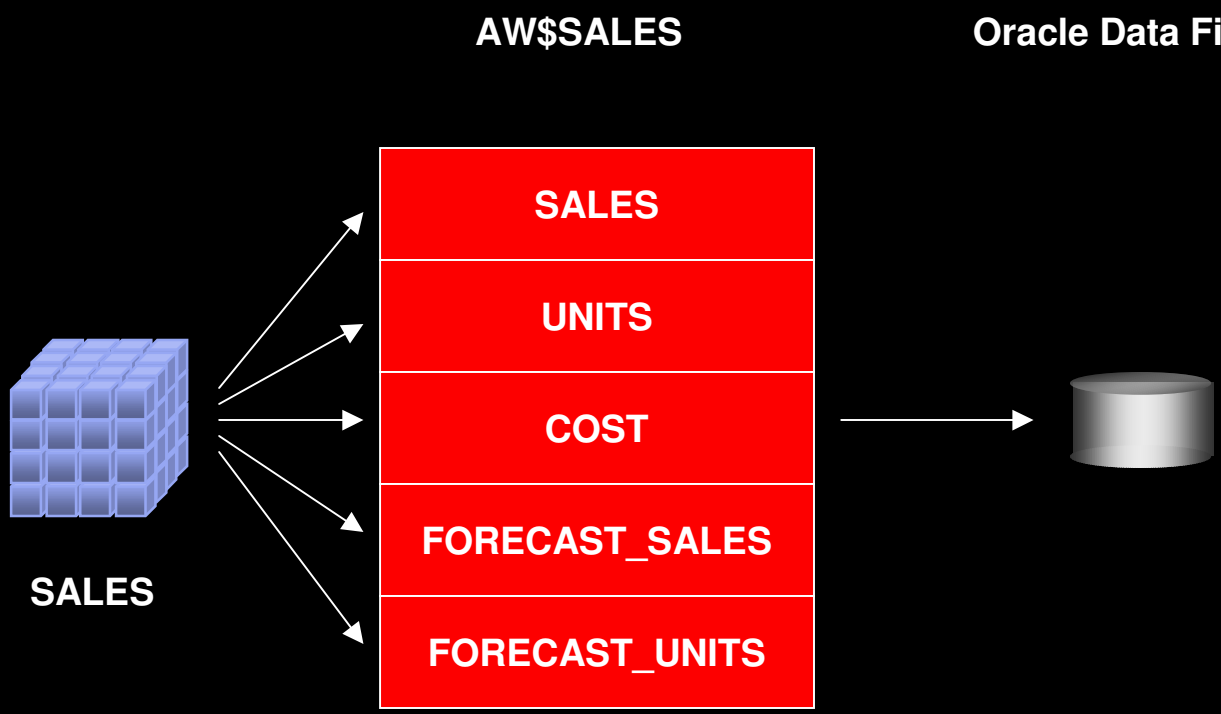
- Dynamic aggregations, measures, members, forecasts, models, etc.

Data

- Dimension members, measures, stored aggregations

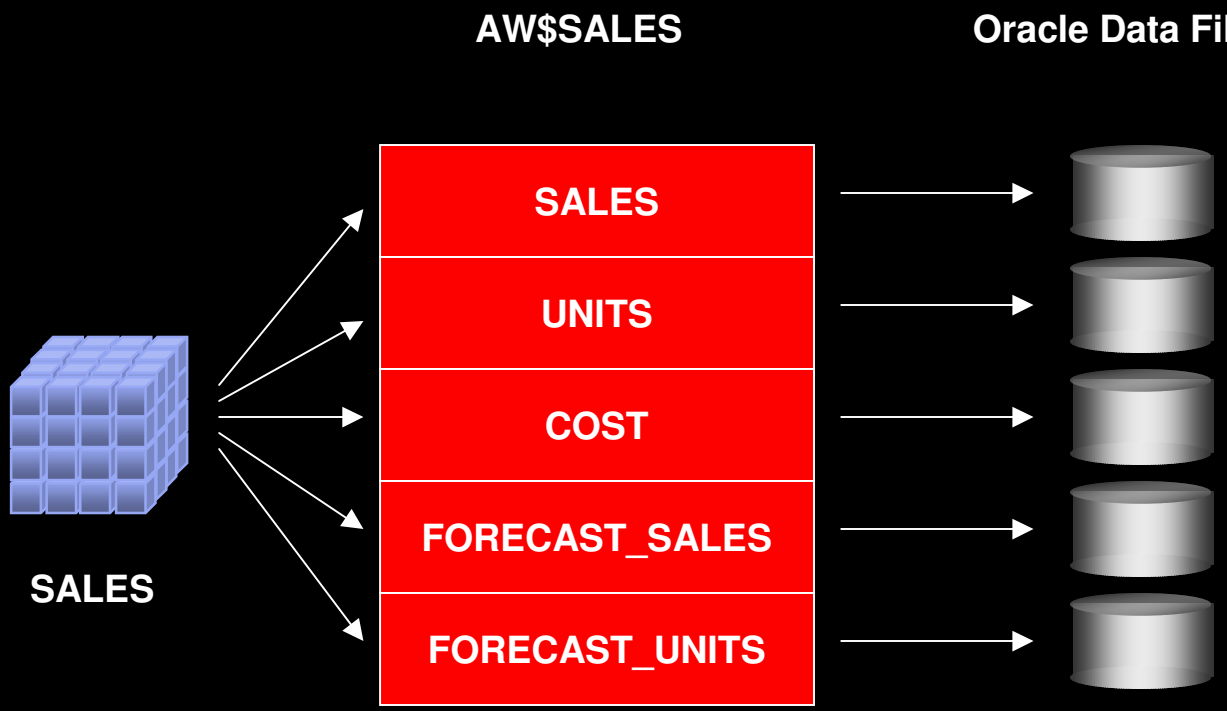
Cubes is presented to application as fully solved

Storage Model



Objects in analytic workspace are stored in separate rows in the AW$ table

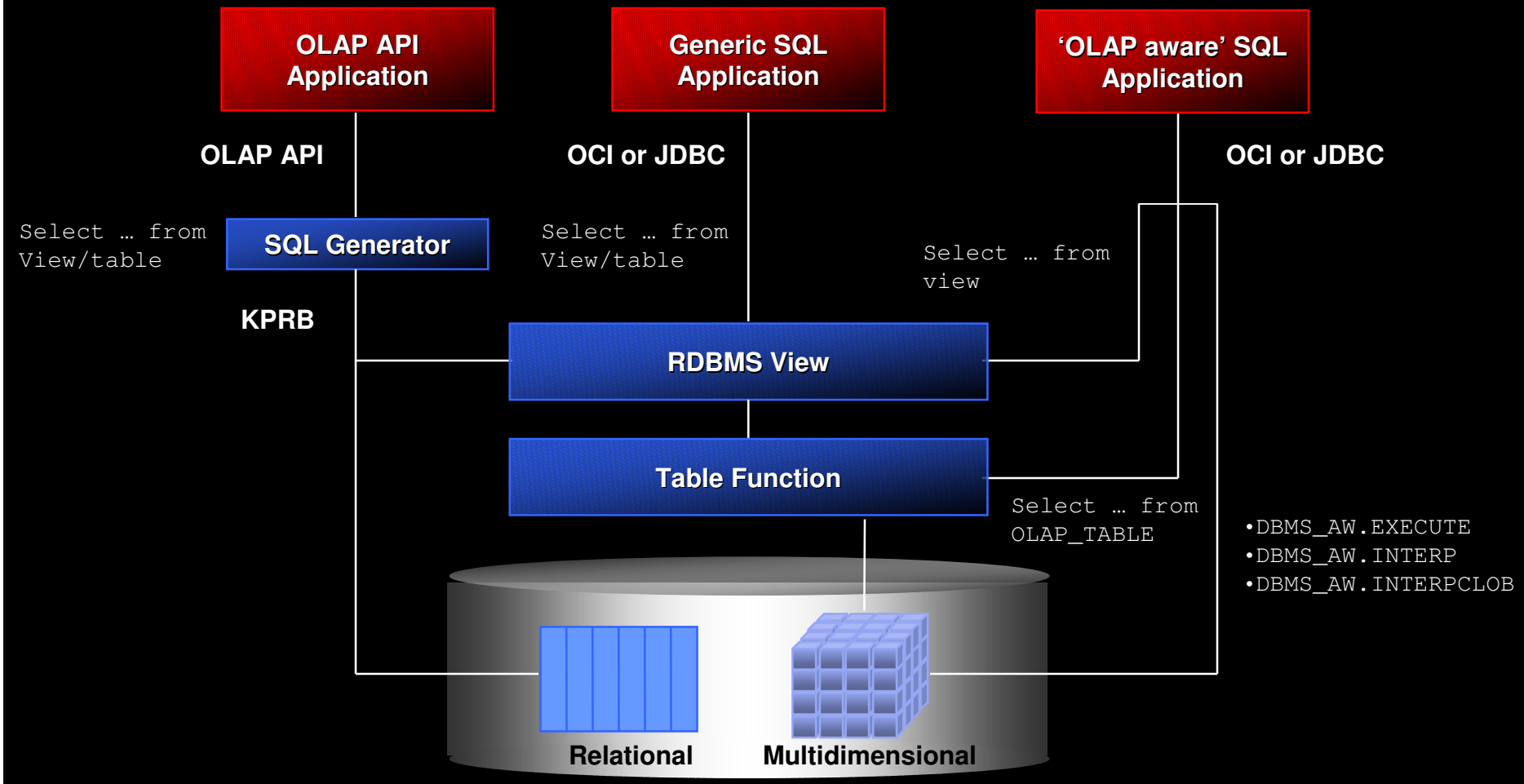
Storage Model



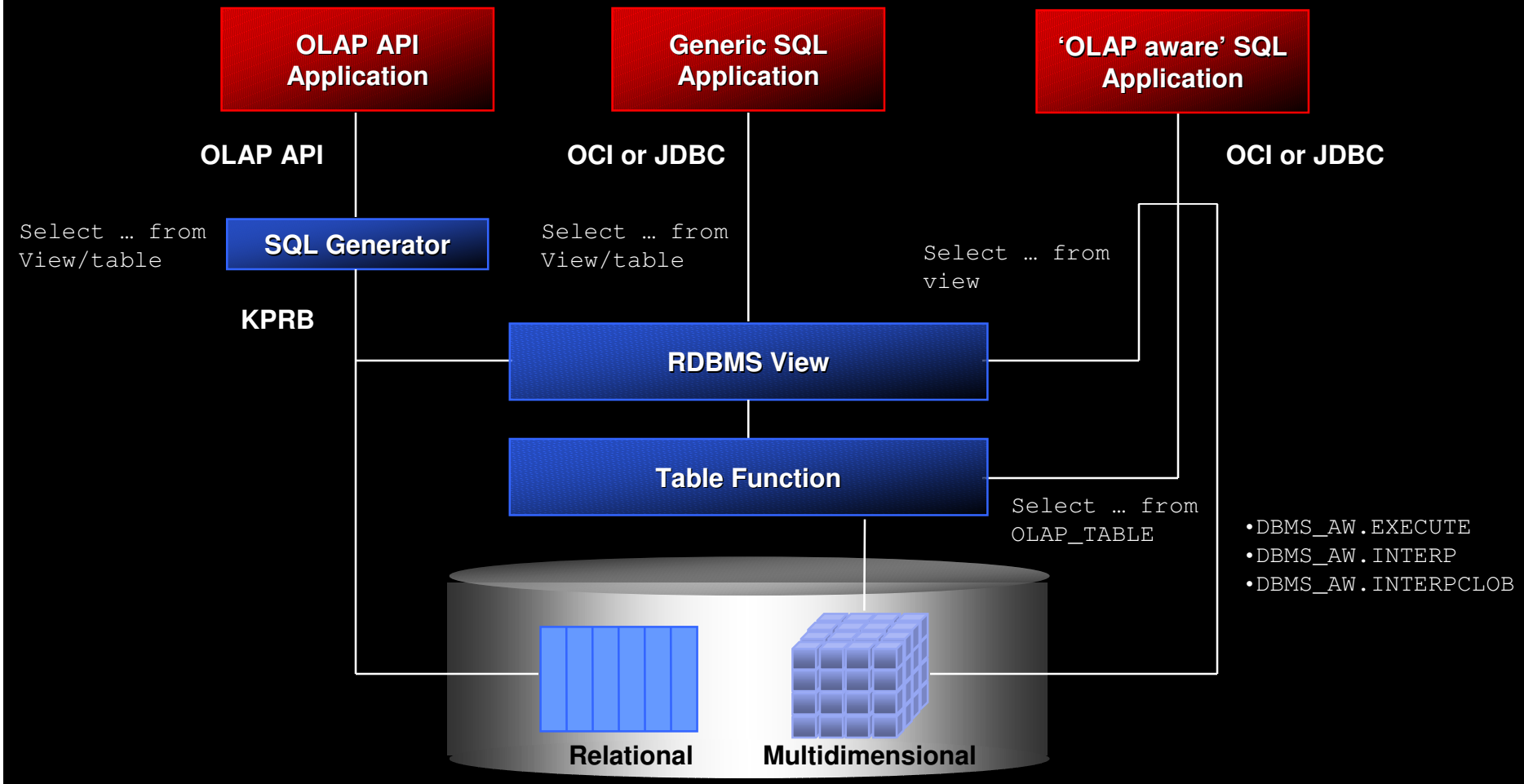
AW$ table can be partitioned using table partitioning

Query Methods …

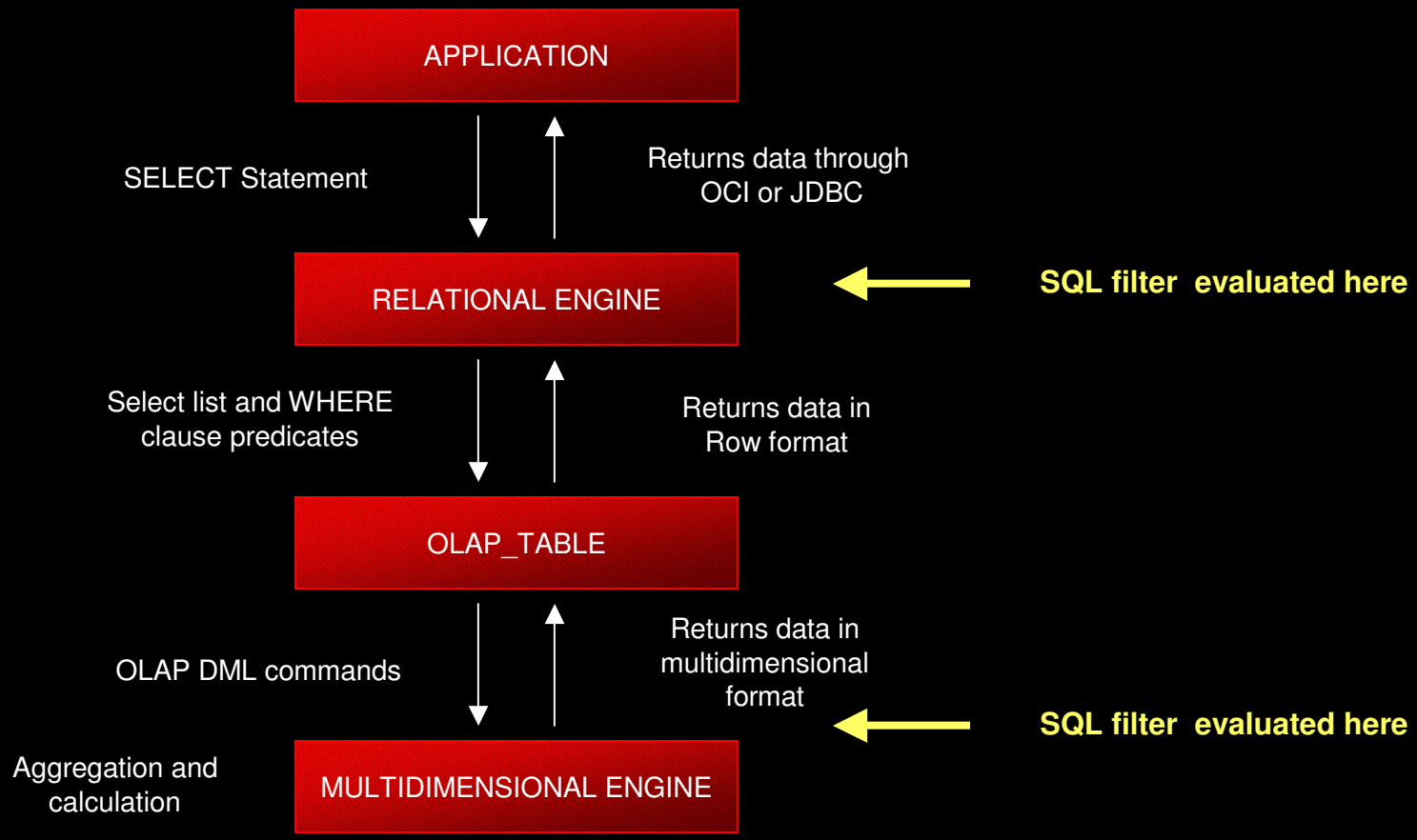
Query Methods



Query Methods



SQL Query Processing

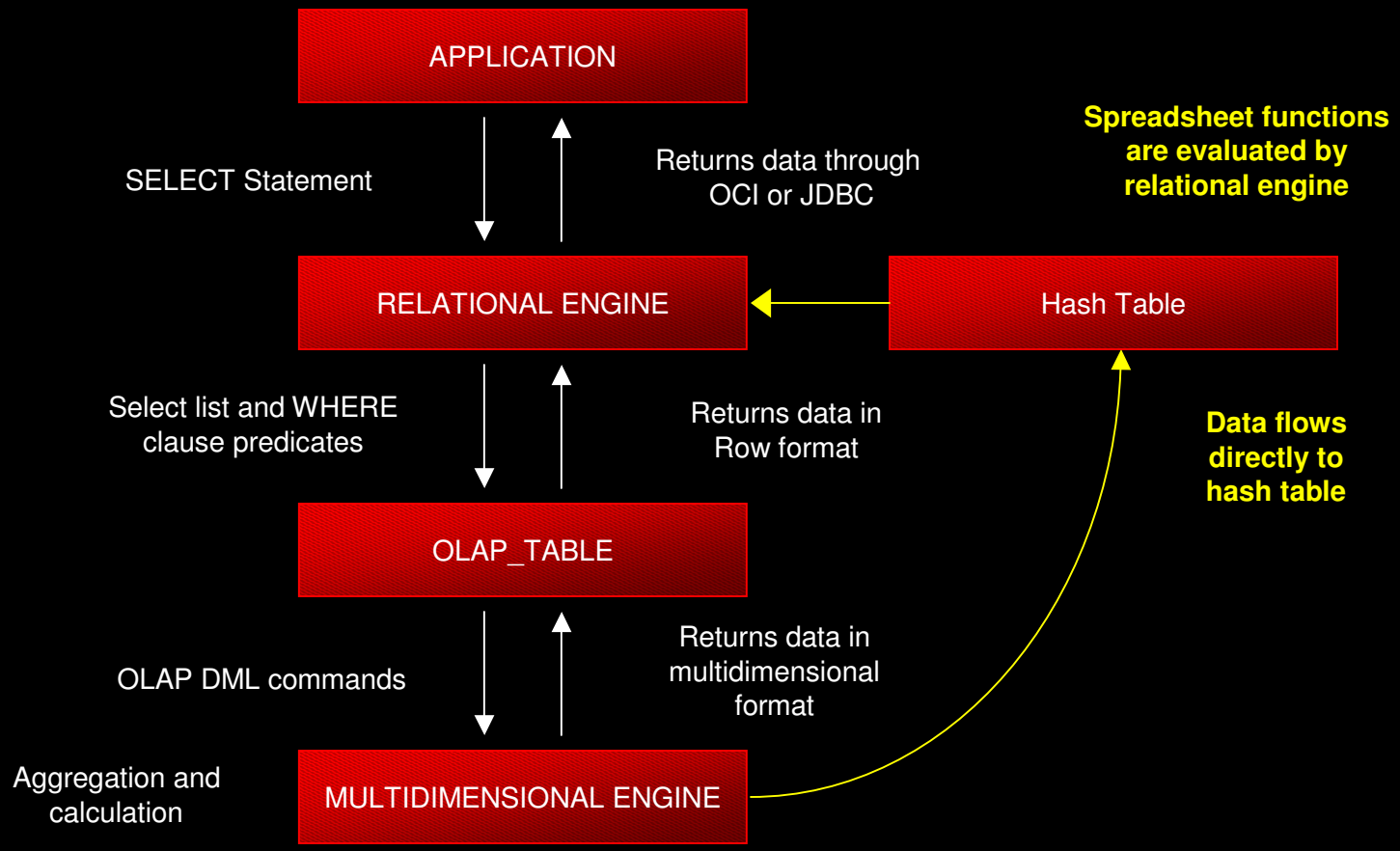


SQL MODEL

- New MODEL clause provides syntax for the “custom dimension member” like functionality

```
select prod, year, amount
from sales
model dimension by (prod, year) measures (amount)
(
   amount[any, any] = 1.1*amount[cv(prod), cv(year) - 1]
   amount['Games', 2002] = amount['Games', 2001] + amount['Games', 2000],
   amount['Accessories', 2002] = 1.2* sum(amount)['Accessories', for year in
(1998, 1999, 2000, 2001)]
)
```

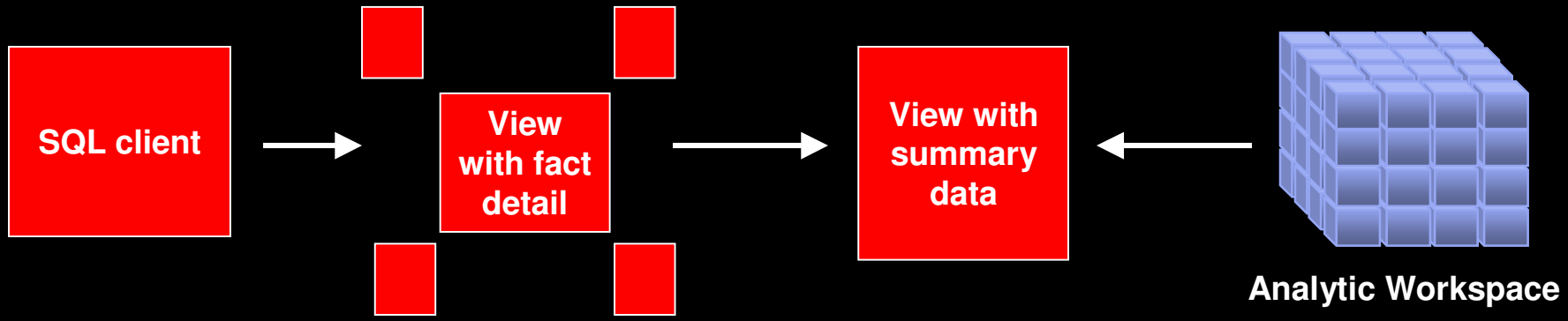
SQL Query Processing



Query Equivalency

- RDBMS feature
  - Allows DBA to indicate that a table or view *could have been* created using specified SQL
  - Allows query re-write into views over analytic workspaces
  - Simplifies administration of SQL based tools

Query Rewrite into Analytic Workspaces



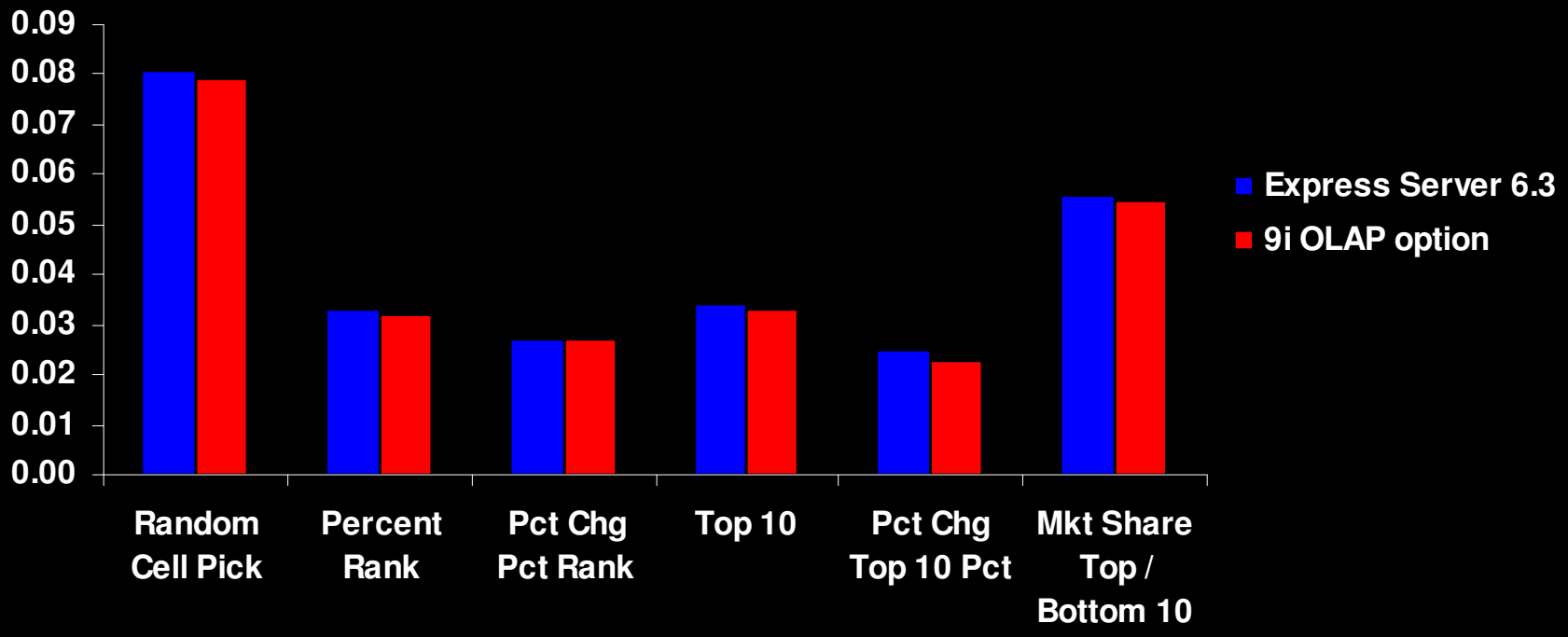
1) Application queries summary data by issue SQL against detail fact table with GROUP BY

2) Query is automatically rewritten to use summary view

3) Data is accessed from analytic workspace

Query Performance

As fast as as Express



SQL Examples …

A Time Dimension View

```
CREATE OR REPLACE VIEW time_view AS
SELECT *
FROM TABLE(OLAP_TABLE('global DURATION SESSION',
  '',
  '',
  'DIMENSION member_id as varchar2(10) from time WITH
      HIERARCHY parent_id as varchar2(10) from
            time_parentrel(time_hierlist ''CALENDAR_YEAR'')
        INHIERARCHY TIME_INHIER
        FAMILYREL
          MONTH_ID as varchar(10),
          QUARTER_ID as varchar(10),
          YEAR_ID as varchar(10)
          FROM
            time_familyrel(time_levellist ''MONTH''),
            time_familyrel(time_levellist ''QUARTER''),
            time_familyrel(time_levellist ''YEAR'')
   ATTRIBUTE short_desc as varchar2(10) FROM time_short_description
   ATTRIBUTE long_desc as varchar2(10) FROM time_long_description
   ATTRIBUTE time_span as number FROM time_time_span
   ATTRIBUTE end_date as date FROM time_end_date
   ATTRIBUTE month_of_quarter as varchar2(16) FROM time_month_of_quarter
   ATTRIBUTE month_of_year as varchar2(16) FROM time_month_of_year
   ATTRIBUTE quarter_of_year as varchar2(16) FROM time_quarter_of_year'
));
```

A Time Dimension View

The view

```
SQL> desc time_et_view;
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------------
 MEMBER_ID                                          VARCHAR2(10)
 PARENT_ID                                          VARCHAR2(10)
 MONTH_ID                                           VARCHAR2(10)
 QUARTER_ID                                         VARCHAR2(10)
 YEAR_ID                                            VARCHAR2(10)
 SHORT_DESC                                         VARCHAR2(10)
 LONG_DESC                                          VARCHAR2(10)
 TIME_SPAN                                          NUMBER
 END_DATE                                           DATE
 MONTH_OF_QUARTER                                   VARCHAR2(16)
 MONTH_OF_YEAR                                      VARCHAR2(16)
 QUARTER_OF_YEAR                                    VARCHAR2(16)
```

A Time Dimension View

Query using standard SQL

```
SQL> select member_id, long_desc, quarter_of_year
     from time_et_view where parent_id = '102‘
     and quarter_of_year in ('2','3')
     order by end_date;

MEMBER_ID  LONG_DESC  QUARTER_OF_YEAR
---------- ---------- ---------------
99         Q2-03      2
100        Q3-03      3
```

Simple Sales Fact View

```
/* A simple fact view */


CREATE OR REPLACE VIEW units_cube_star_fact_view AS
SELECT *
FROM TABLE(OLAP_TABLE('global DURATION SESSION',
  '',
  '',
  'DIMENSION time_id as varchar2(20) from time
   DIMENSION customer_id as varchar2(20) from customer
   DIMENSION product_id as varchar2(20) from product
   DIMENSION channel_id as varchar2(20) from channel
   MEASURE sales as number FROM units_cube_sales
   MEASURE units as number FROM units_cube_units
   MEASURE extended_cost as number FROM units_cube_extended_cost
   ROW2CELL olap_calc'
));
```

Simple Sales Fact View

```
/* A simple select with olap_expression */

select
  time_id,
  customer_id,
  product_id,
  channel_id,
  sales,
  units,
  extended_cost,
  olap_expression(olap_calc,'lag(units_cube_sales,1,time,status)') as SALES_PRIOR_PRIOD
from units_cube_star_fact_view
where
  time_id in ('1','2','3','4','85','102','119','145')
  and customer_id = 'TOTAL_CUSTOMER_1'
  and product_id = 'TOTAL_PRODUCT_1'
  and channel_id = 'TOTAL_CHANNEL_1';
```

Fact View with Dimensional Attributes

```
CREATE OR REPLACE VIEW units_cube_attr_fact_view AS
SELECT *
FROM TABLE(OLAP_TABLE('global DURATION SESSION',
  '',
  '',
  'DIMENSION time_id as varchar2(20) from time with
     ATTRIBUTE time_level as varchar2(15) from time_levelrel
     ATTRIBUTE time_parent as varchar2(15) from time_parentrel
     ATTRIBUTE time_dsc as varchar2(15) from time_long_description
   DIMENSION customer_id as varchar2(20) from customer with
     ATTRIBUTE customer_level as varchar2(15) from customer_levelrel
     ATTRIBUTE customer_parent as varchar2(20) from customer_parentrel
     ATTRIBUTE customer_dsc as varchar2(25) from customer_long_description
   DIMENSION product_id as varchar2(20) from product with
     ATTRIBUTE product_level as varchar2(15) from product_levelrel
     ATTRIBUTE product_parent as varchar2(15) from product_parentrel
     ATTRIBUTE product_dsc as varchar2(25) from product_long_description
   DIMENSION channel_id as varchar2(20) from channel with
     ATTRIBUTE channel_level as varchar2(15) from channel_levelrel
     ATTRIBUTE channel_parent as varchar2(15) from channel_parentrel
     ATTRIBUTE channel_dsc as varchar2(25) from channel_long_description
   MEASURE sales as number FROM units_cube_sales
   MEASURE units as number FROM units_cube_units
   MEASURE extended_cost as number FROM units_cube_extended_cost
   ROW2CELL olap_calc'
));
```

Fact View with Dimensional Attributes

```
select
  time_id,
  time_dsc,
  customer_id,
  customer_dsc,
  product_id,
  product_dsc,
  channel_id,
  channel_dsc,
  sales,
  units,
  extended_cost
from
  units_cube_attr_fact_view
where
  time_parent = '102'
  and customer_level = 'TOTAL_CUSTOMER'
  and product_level = 'TOTAL_PRODUCT'
  and channel_level = 'TOTAL_CHANNEL';
```

Fact View with Dimensional Attributes

TIME_ID	TIME_DSC	SALES	UNITS	EXTENDED_COST
100	Q3-03	33636358.5	140931	31494036.7
101	Q4-03	36446069.5	145716	34329834.6
98	Q1-03	26946411	110104	24982346.3
99	Q2-03	33247675.7	137318	30739600.7

Oracle 10g OLAP ...

Oracle10g OLAP Highlights

- Support for large multidimensional data sets
- Administration
- Query interfaces

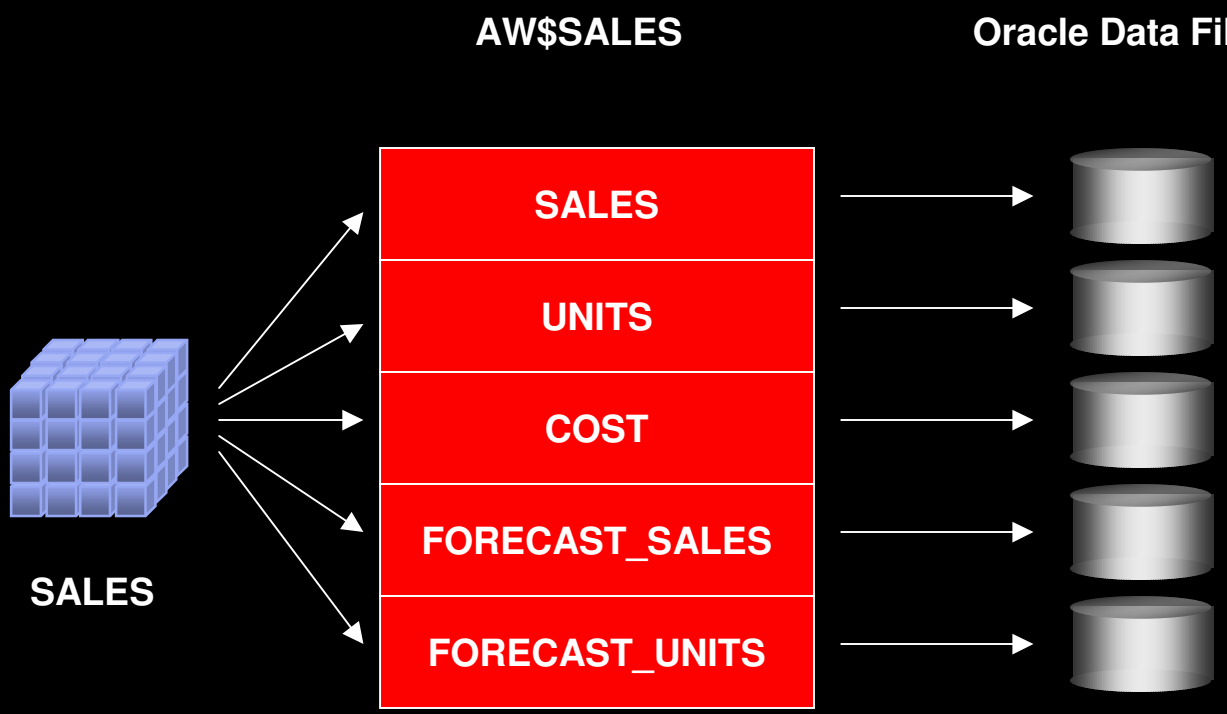
Large Multidimensional Data Sets

- Physical storage model enhancements
- MULTI attach mode
- Partitioning
- Parallelism
- Aggregation from formulas
- Indexing optimizations
- Real Application Clusters and Grid Computing

9*i* Release 2 Storage Model
AW$ table

Analytic Workspace data is stored in tables as LOB data type

Oracle10g Storage Model



AW$ table can be partitioned using table partitioning

MULTI Attach Mode

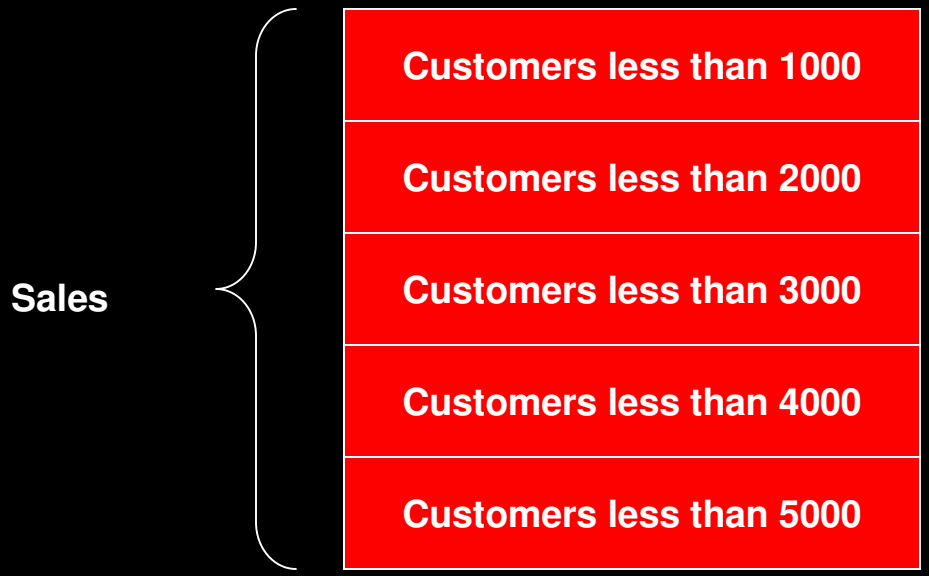
- 10*g* MULTI attach mode allows multiple sessions to attach to the analytic workspace read-write
- Use separate sessions to
  - Parallelize data loading tasks
  - Aggregate data
  - Allocate data
  - Solve models
  - Forecast data
  - Etc

Partitioned Variables

- Engine level partitioning of variable objects in the analytic workspace
- Each partition becomes a row in the AW$ table
- Partitioning methods
  - RANGE
  - LIST
  - CONCAT

Partitioned Variables

- RANGE partitioning
  - Partitions based on a range of dimension members



Partitioned Variables

- LIST partitioning
  - Partitions based on a list of named members

Compressed Cubes

- New storage format
- Optimized data storage for very sparse data sets
  - Large, highly dimensioned models
  - Extremely sparse data

Aggregation

- AGGREGATE_FROM
  - Derives aggregate level data from calculations at leaf level
  - Eliminates need to persist calculated data at leaf level

Aggregation Sources
Traditional Method

Traditional method of storing data at leaf levels from the results of a computation

Aggregation Sources
AGGREGATE_FROM

Using AGGREGATE FROM to aggregate from a formula eliminates ETL step and reduces size of database

Analytic Workspace Manger and Administration …

ORACLE

Administration

- New Java and XML based administrative interfaces
- Major upgrade to administrative GUI tools
- Simplified administrative processes

Administrative Interfaces

- Object oriented DDL/DML via Java and XML based APIs
  - Fully abstracts logical dimensional model from physical design
  - Supports “data immersive” administrative experience
  - Used by Analytic Workspace Manager and Oracle Warehouse Builder

Analytic Workspace Manager

- All new GUI for analytic workspace administration
  - Designed for the ‘data dabbler’
  - Dimensional modeling
  - Flexible data source mapping
  - Emphasis on data generation

Analytic Workspace Manager

- Purpose
  - Quickly define and build analytic workspaces
    - Appeals to DBA and LoB power user cube designers, and support SCs during PoCs
    - Quickly get SCs to the value add of embellishing the AW with computation and presentation
    - Provide all-GUI experience/demonstration
  - Support both the Database as vendor neutral BI platform and Oracle BI stack
    - Presented as general AW administrative tool

Analytic Workspace Manager

- Define logical dimensional model
- Implement physical model in analytic workspace
- Map to relational data sources
- Lifecycle management
- Templates

Analytic Workspace Manager

- Dimensional modeling from an end-user perspective
  - Dimensions
  - Cubes
  - Custom measures
  - Aggregations

Analytic Workspace Manager

- Implement model in analytic workspace
  - OLAP option automatically builds an efficient analytic workspace based on the logical model
    - Automatically uses partitioning, parallelism, compressed composite/cube, aggregations
    - Eliminates the need to program using the OLAP DML for general cube construction

Analytic Workspace Manager

- Mapping to relational sources
  - Data source can be a relational object or OLAP DML
  - Supports wider variety of relational sources as compared to AWM 9.2
    - Stars, snowflakes, network of tables
    - Tables, views, dblinks, etc.
  - Not an ETL tool, but compatible with Oracle Warehouse Builder

Analytic Workspace Manager

- Manage analytic workspace throughout its lifecycle
  - Data loading
  - Automatically aggregates and calculates measures according to calculation rules
  - Cube refresh and resolve
  - Supports Oracle Job Queue

Analytic Workspace Manager

- Templates
  - Save dimensions, cubes and measures to template files
  - Create objects and AWs from templates
  - Used to
    - Share analytic workspace designs with Oracle Warehouse Builder, other users and with other applications
    - Transfer object definitions to other schema or instances
    - Persist object definitions outside database
    - Place object definitions in source control

Analytic Workspace Manager

- Roadmap: Product Release
  - Mapping features
    - Loading data at more than one level
    - Mapping multiple fact tables to a cube
    - Multi-part key mappings

Analytic Workspace Manager

- Roadmap: Production Release
  - Aggregation features
    - Apply aggregation rules to individual measures
    - Choose aggregation hierarchy
    - Specify caching options

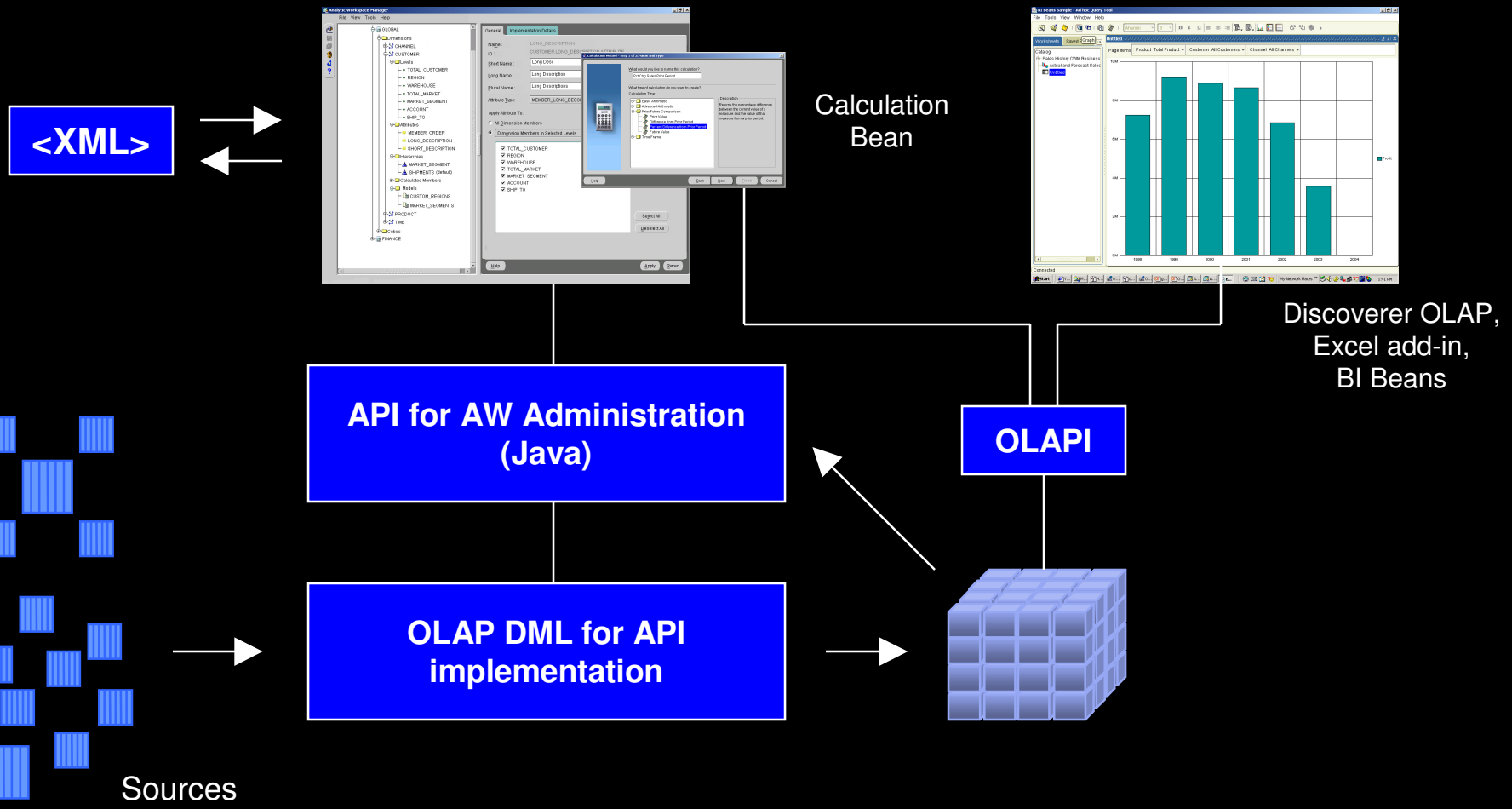
Analytic Workspace Manager

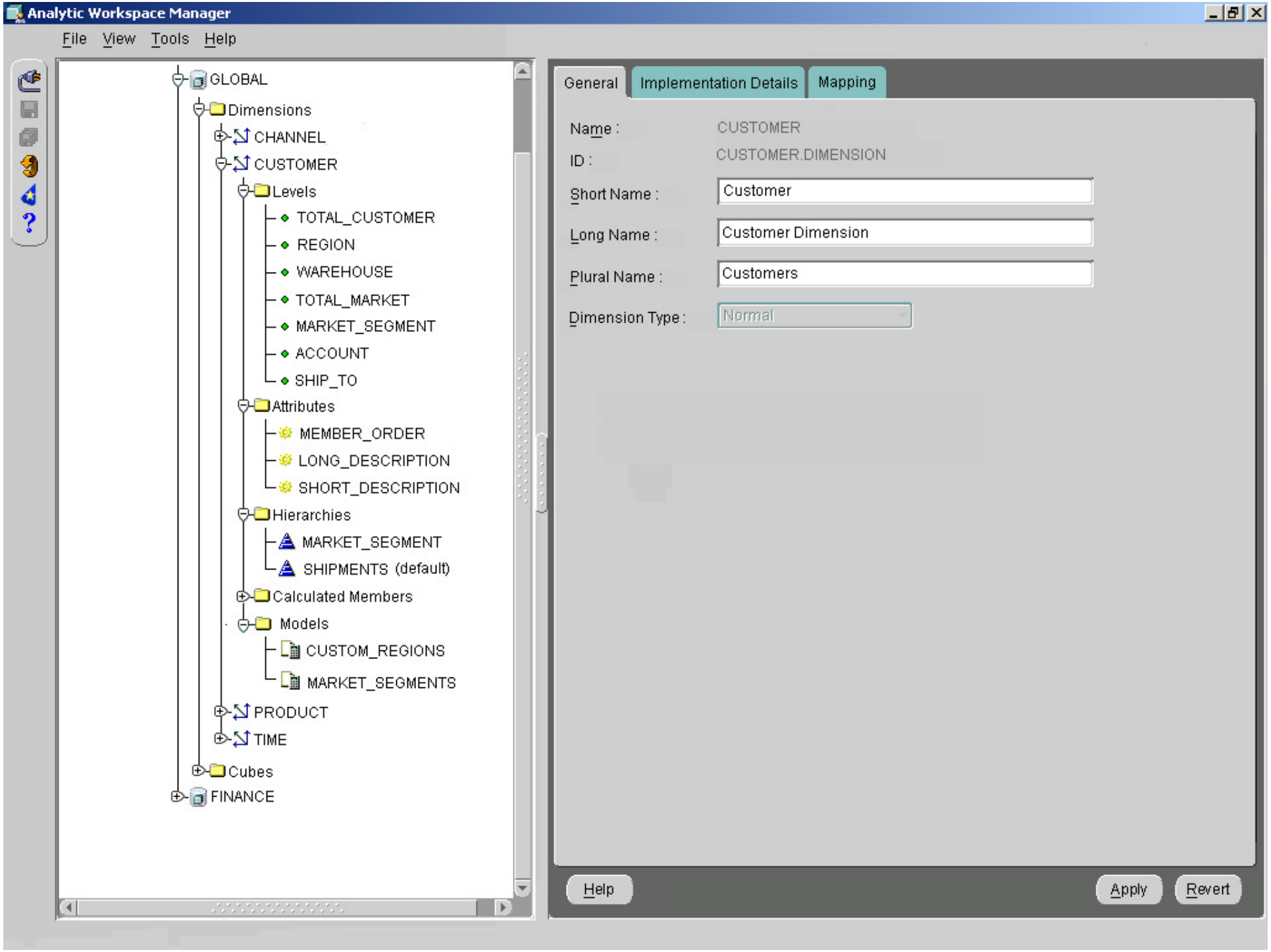
- Roadmap: Post 10.1.0.4
  - Calculation plans, with additional calculations
    - Forecasts
    - Allocations
    - Models (systems of equations)
  - View data via cross tab and graph
  - Work off-line from database
    - Design and deploy mode

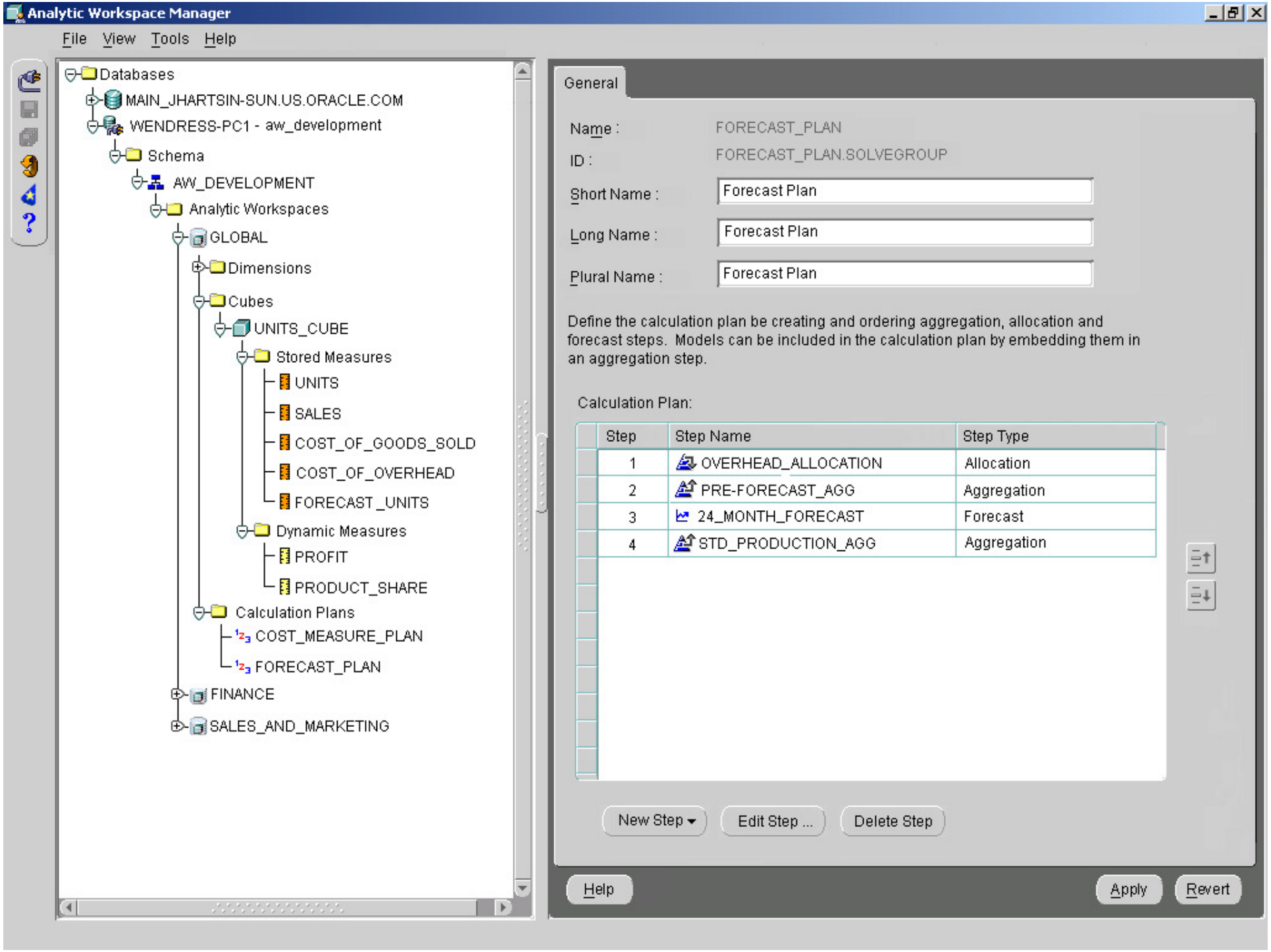
Analytic Workspace Manager

- Roadmap: Post 10.1.0.4
  - Dimension maintenance
    - Add/remove/rename dimension members
    - Custom members
    - Change child-parent relationships
  - Data write back
    - Change measure/fact data
  - Member/cell level security (“PERMIT”)
    - Can be implemented in OLAP DML

Architecture




Analytic Workspace Manager
File View Tools Help
GLOBAL
Dimensions
CHANNEL
CUSTOMER
Levels
TOTAL_CUSTOMER
REGION
WAREHOUSE
TOTAL_MARKET
MARKET_SEGMENT
ACCOUNT
SHIP_TO
Attributes
MEMBER_ORDER
LONG_DESCRIPTION
SHORT_DESCRIPTION
Hierarchies
MARKET_SEGMENT
SHIPMENTS (default)
Calculated Members
Models
CUSTOM_REGIONS
MARKET_SEGMENTS
PRODUCT
TIME
Cubes
FINANCE
General
Implementation Details
Mapping
Name : CUSTOMER
ID : CUSTOMER.DIMENSION
Short Name : Customer
Long Name : Customer Dimension
Plural Name : Customers
Dimension Type : Normal
Help
Apply
Revert


Analytic Workspace Manager
File View Tools Help
Databases
MAIN_JHARTSIN-SUN.US.ORACLE.COM
WENDRESS-PC1 - aw_development
Schema
AW_DEVELOPMENT
Analytic Workspaces
GLOBAL
Dimensions
Cubes
UNITS_CUBE
Stored Measures
UNITS
SALES
COST_OF_GOODS_SOLD
COST_OF_OVERHEAD
FORECAST _UNITS
Dynamic Measures
PROFIT
PRODUCT_SHARE
Calculation Plans
COST_MEASURE_PLAN
FORECAST_PLAN
FINANCE
SALES_AND_MARKETING
General
Name : FORECAST_PLAN
ID : FORECAST_PLAN.SOLVEGROUP
Short Name : Forecast Plan
Long Name : Forecast Plan
Plural Name : Forecast Plan
Define the calculation plan be creating and ordering aggregation, allocation and forecast steps. Models can be included in the calculation plan by embedding them in an aggregation step.
Calculation Plan:
Step
Step Name
Step Type
1 OVERHEAD_ALLOCATION Allocation
2 PRE-FORECAST_AGG Aggregation
3 24_MONTH_FORECAST Forecast
4 STD_PRODUCTION_AGG Aggregation
New Step
Edit Step ...
Delete Step
Help
Apply
Revert

Query Interfaces …

Query Interfaces

- Simplification of SQL and OLAP API access structures
- SQL interface enhancements

SQL and OLAP API Access

- Oracle10g simplified and hardens access structures
  - SQL access
    - Automatically generated ADTs
  - OLAPI API access
    - OLAPI APIs does not require predefined views
    - All metadata read directly from the analytic workspace

SQL Interface

- Application of relational filters to multidimensional data types
- SQL MODEL support
- Query rewrite over multidimensional data types

Oracle Advantages

- Industrial strength calculation engine
- Elimination/reduction of data replication
- Computational scalability
- Concurrent user scalability
- Open, modern interfaces
- High availability
- Easer to manage
- Most complete BI/DW/OLAP product line
- Platform support

Oracle is the only relational-multidimensional database

	Oracle	IBM	Hyperion	Microsoft
Single RDBMS-MDDS process	Yes	No	No	No
Single data store	Yes	No	No	No
Single metadata repository	Yes	No	No	No
Single set of management tools	Yes	No	No	No
Single security model	Yes	No	No	No
OLAP API and SQL interfaces	Yes	No	No	No


ORACLE®