Business Intelligence Forecasting and Modeling with Oracle10^{g} SQL

Anthony D. Noriega MSCS, MBA, BSSE, OCP-DBA

ORACLE CERTIFIED PROFESSIONAL

ORACLE Technology NETWORK

Speaker Qualifications

- Independent Consultant, ADN
- Over 20 years of IT experience, 16 with Oracle
- Former OCP Instructor
- Speaker at NYOUG and IOUG events
- MS Computer Science, NJIT, 1993
- PhD CIS candidate, NJIT, 1997
- MBA MIS, Montclair State University, 2006
- Forecasting projects at M&M Mars, AT&T
- Analytical/Statistical projects with FMC and AT&T
- Modeling Projects with Bowne, DB, eWayDirect

Objective

- Emphasize Oracle10g SQL Modeling and Forecasting capabilities.
- Discuss areas of current and future applications, and relevant studies.
- Introduce and expand on a frame of reference to approach BI modeling and forecasting.

Business Framework

- Traditional Methodologies
  - Statistical Analysis
    - Linear Regression
    - Multiple Regression
    - Factorial Design
    - Latin Quarters
  - Exponential Smoothing
    - Simple Exponential Smoothing.
    - Double Exponential Smoothing

Business Framework

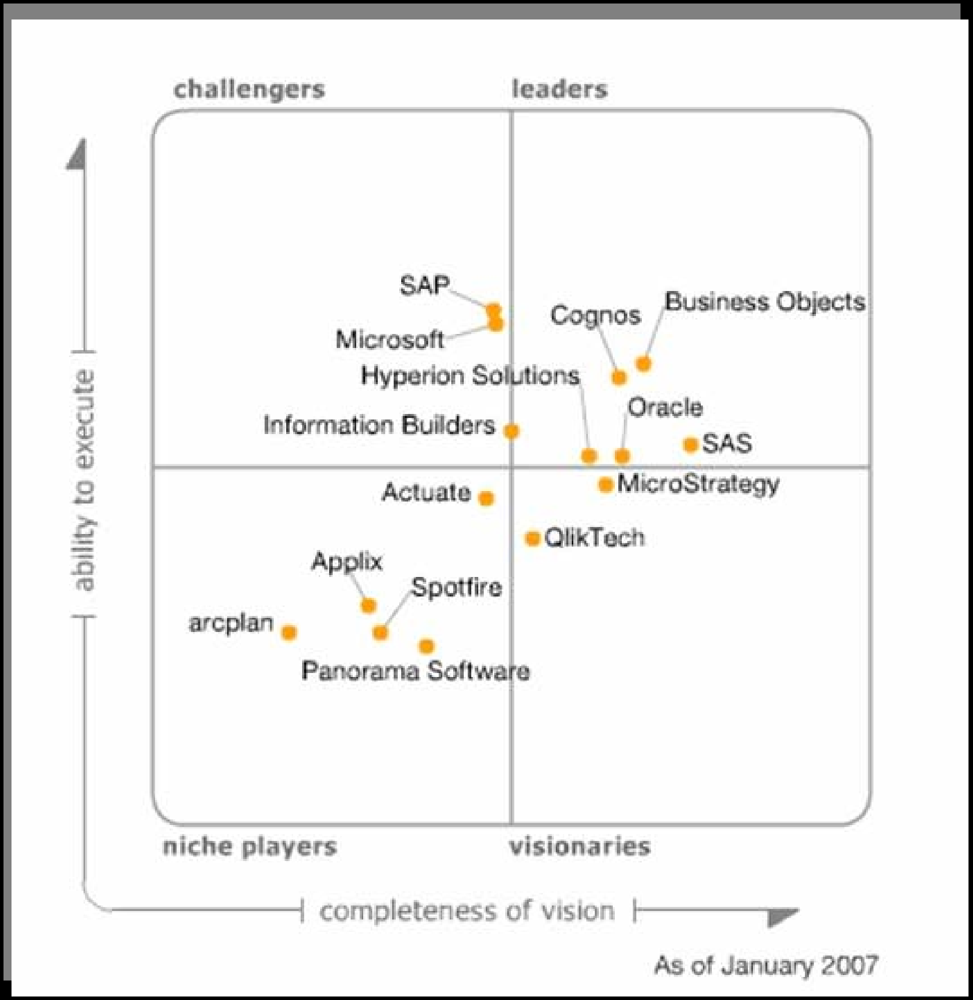
- Traditional Methodologies
  - Time Series Analysis
    - Moving Average
    - Moving Average with Seasonal Adjustments
    - ARMA (Autoregressive Moving Average) Box, Jenkins (1976)
    - ARIMA (Autoregressive Integrated Moving Average).

Business Framework

- Traditional Methodologies
  - Operating Research
    - Markov Chains (Stochastic Processes)
    - Bayesian Models (Probabilistic Models)
    - Game Theory Approaches
  - Mathematical Models
    - Linear Algebra
    - Structured Matrices and Polynomials Models

BI Software Market

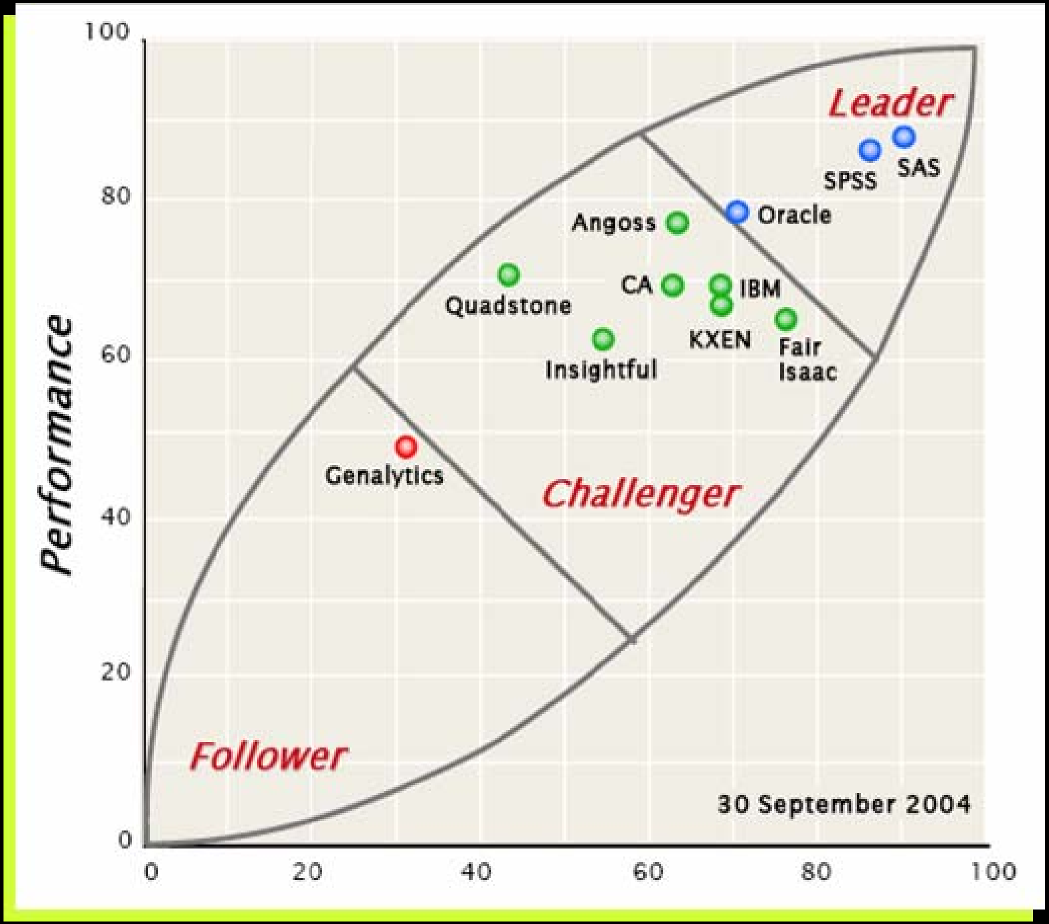
Oracle in the Leader's Quadrant



As of January 2007

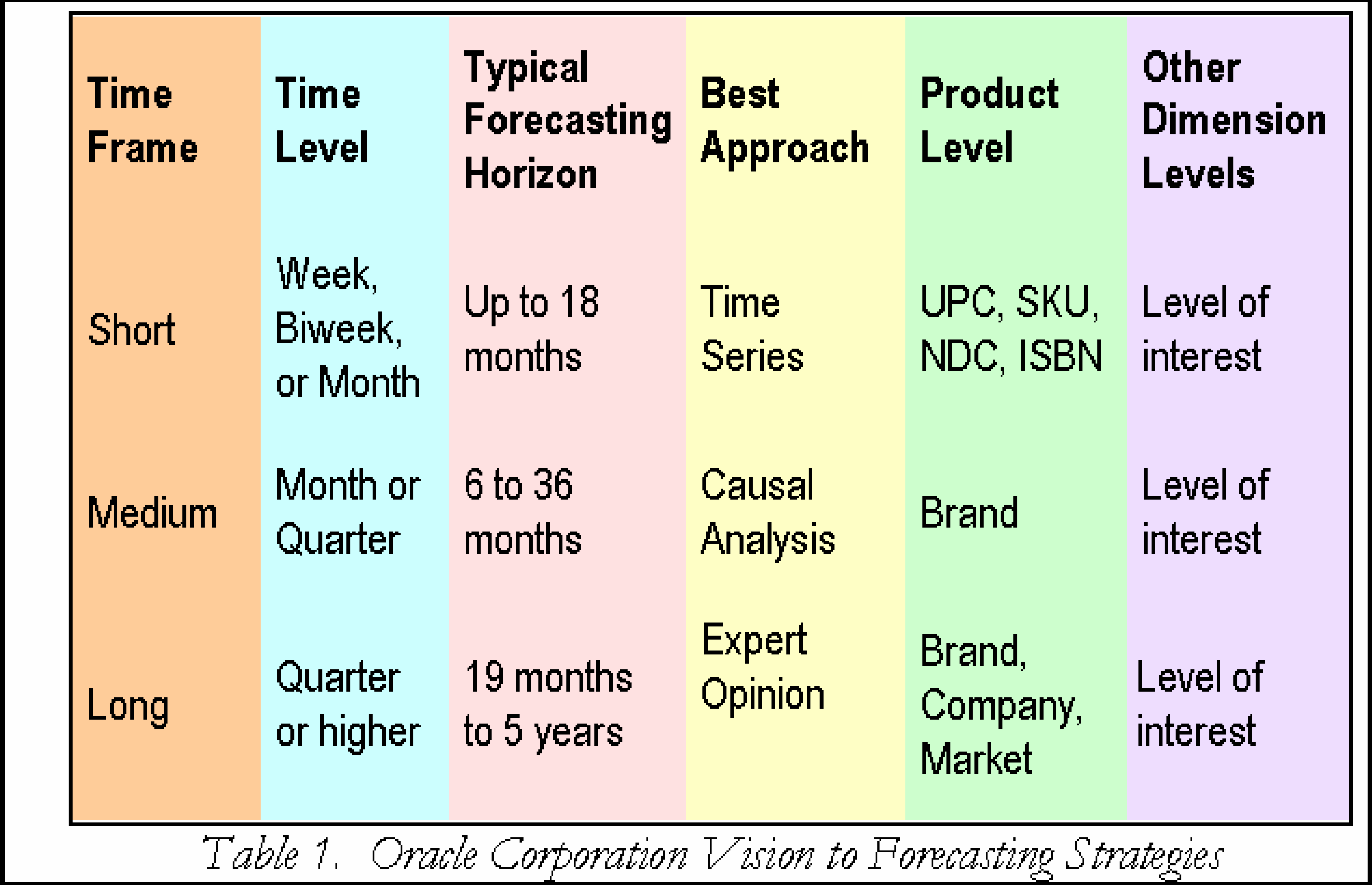
Data Mining Software Market

Oracle in the Leader's Quadrant



30 September 2004

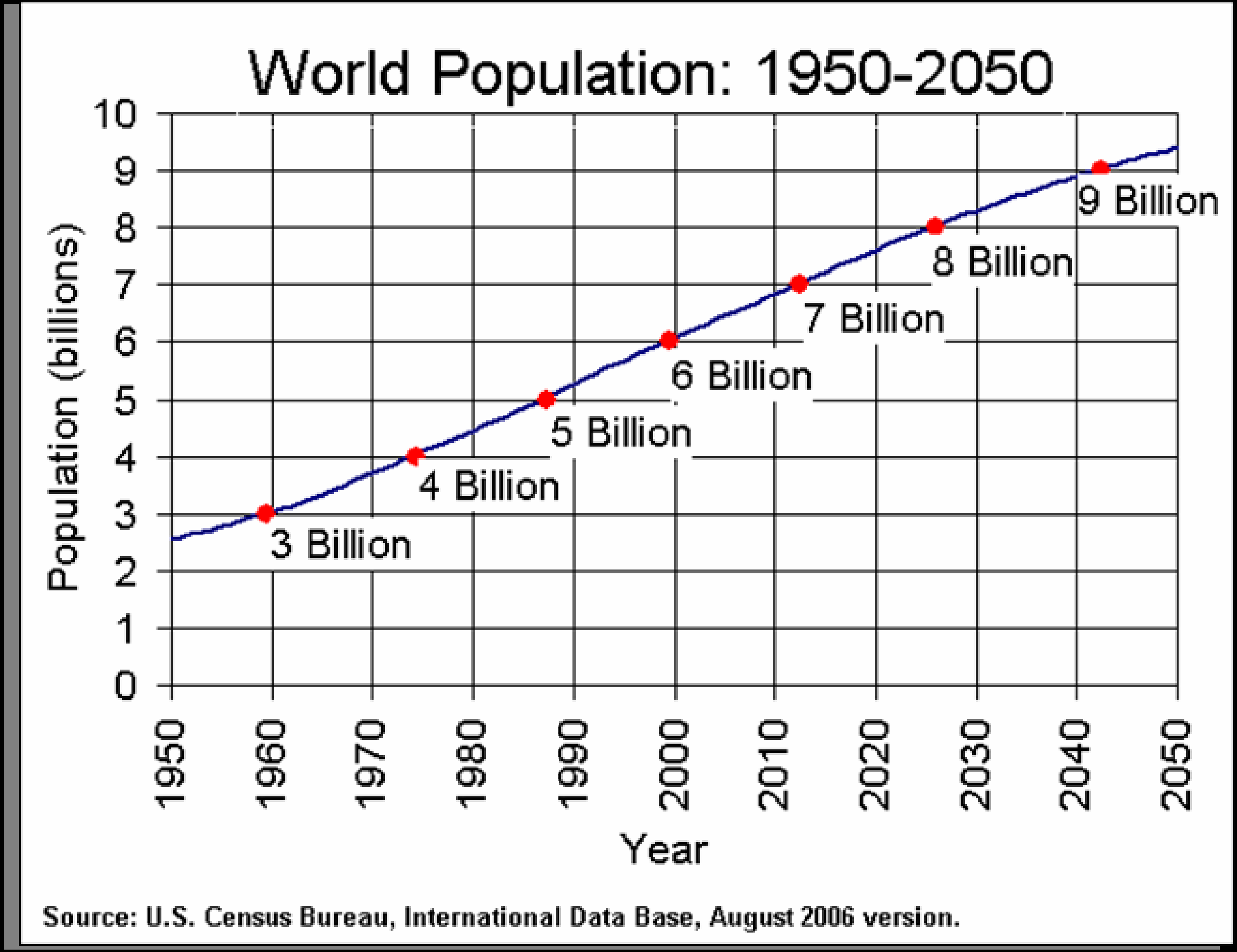
Oracle's BI Approach



Time Frame	Time Level	Typical Forecasting Horizon	Best Approach	Product Level	Other Dimension Levels
Short	Week, Biweek, or Month	Up to 18 months	Time Series	UPC, SKU, NDC, ISBN	Level of interest
Medium	Month or Quarter	6 to 36 months	Causal Analysis	Brand	Level of interest
Long	Quarter or higher	19 months to 5 years	Expert Opinion	Brand, Company, Market	Level of interest

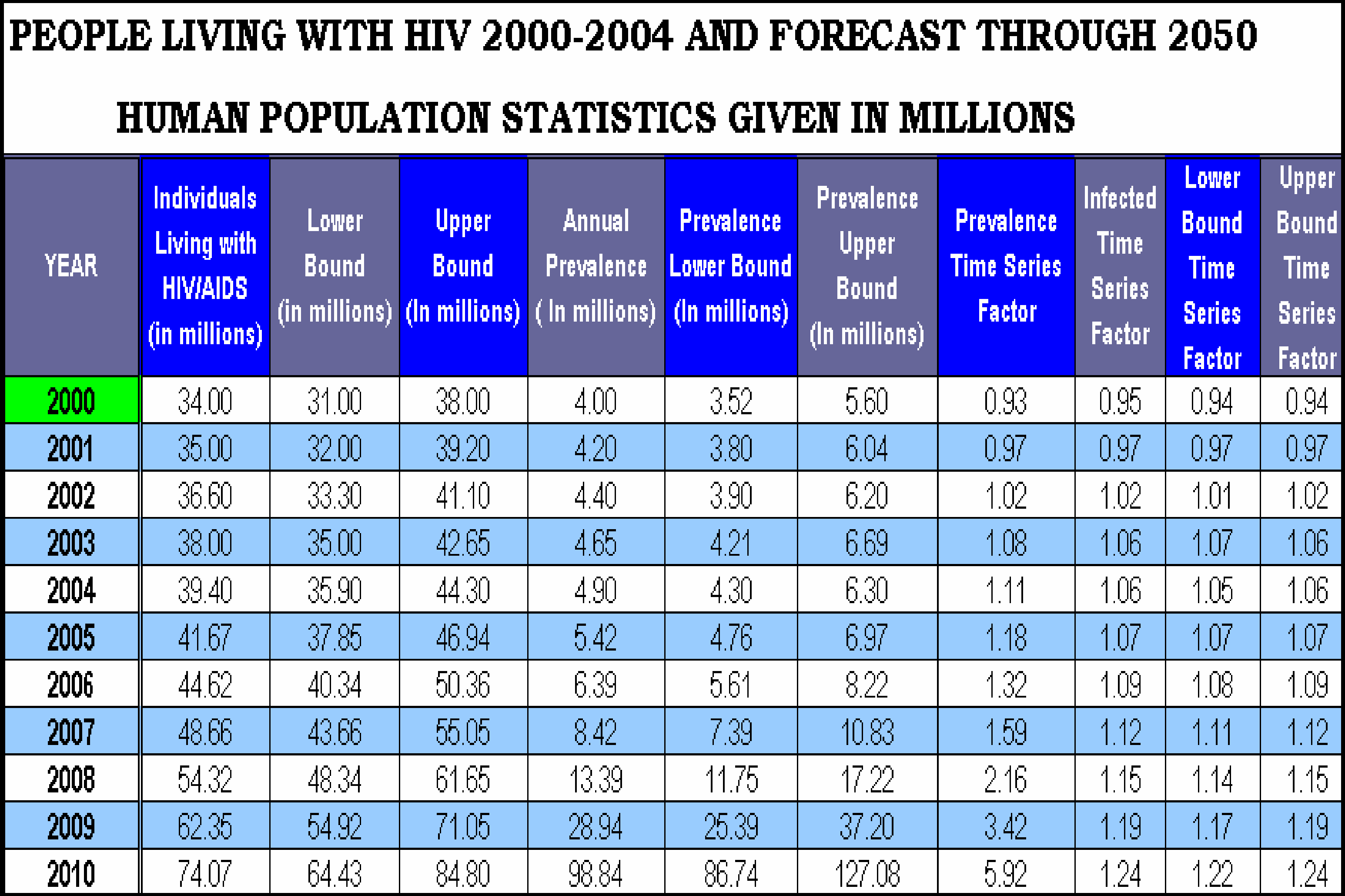
Table 1. Oracle Corporation Vision to Forecasting Strategies

What to Forecast



Source: U.S. Census Bureau, International Data Base, August 2006 version.

What to Forecast



PEOPLE LIVING WITH HIV 2000-2004 AND FORECAST THROUGH 2050

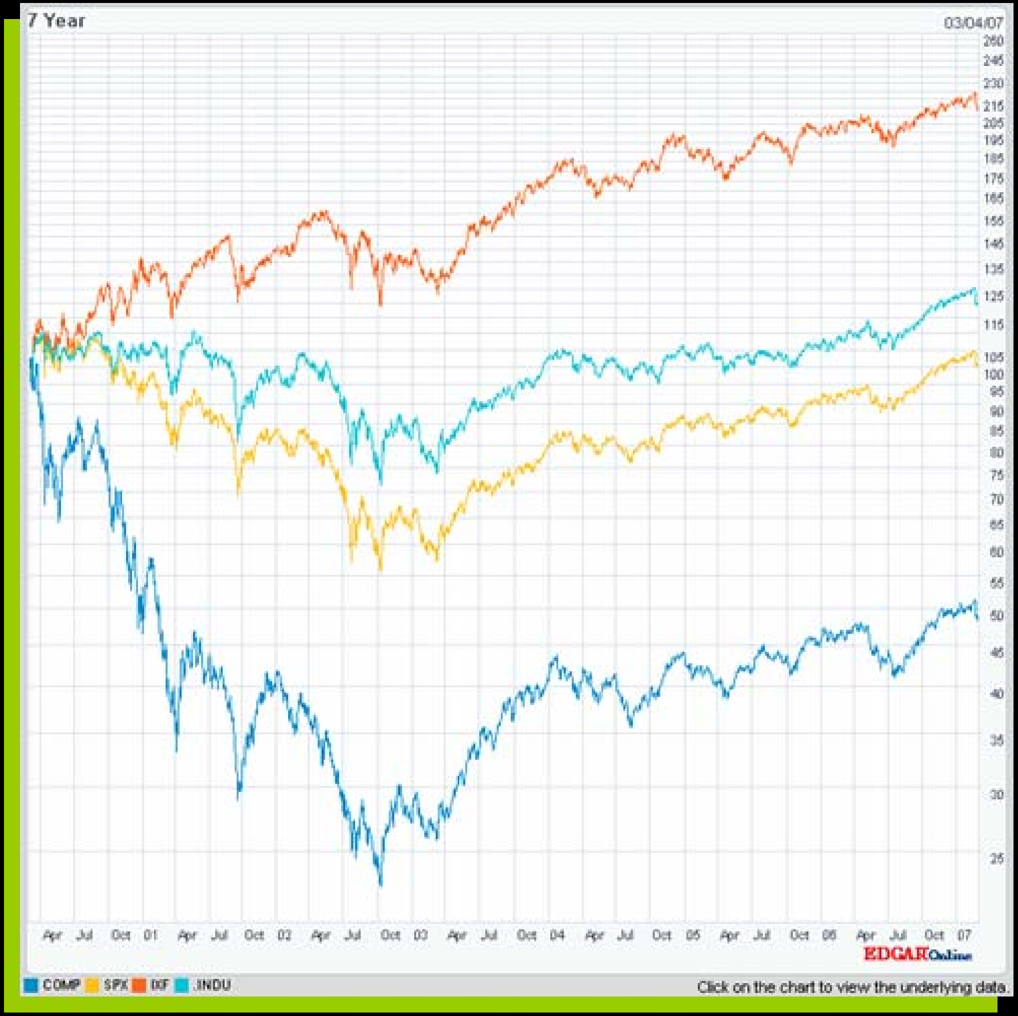
HUMAN POPULATION STATISTICS GIVEN IN MILLIONS

YEAR	Individuals Living with HIV/AIDS (in millions)	Lower Bound (in millions)	Upper Bound (In millions)	Annual Prevalence (In millions)	Prevalence Lower Bound (In millions)	Prevalence Upper Bound (In millions)	Prevalence Time Series Factor	Infected Time Series Factor	Lower Bound Time Series Factor	Upper Bound Time Series Factor
2000	34.00	31.00	38.00	4.00	3.52	5.60	0.93	0.95	0.94	0.94
2001	35.00	32.00	39.20	4.20	3.80	6.04	0.97	0.97	0.97	0.97
2002	36.60	33.30	41.10	4.40	3.90	6.20	1.02	1.02	1.01	1.02
2003	38.00	35.00	42.65	4.65	4.21	6.69	1.08	1.06	1.07	1.06
2004	39.40	35.90	44.30	4.90	4.30	6.30	1.11	1.06	1.05	1.06
2005	41.67	37.85	46.94	5.42	4.76	6.97	1.18	1.07	1.07	1.07
2006	44.62	40.34	50.36	6.39	5.61	8.22	1.32	1.09	1.08	1.09
2007	48.66	43.66	55.05	8.42	7.39	10.83	1.59	1.12	1.11	1.12
2008	54.32	48.34	61.65	13.39	11.75	17.22	2.16	1.15	1.14	1.15
2009	62.35	54.92	71.05	28.94	25.39	37.20	3.42	1.19	1.17	1.19
2010	74.07	64.43	84.80	98.84	86.74	127.08	5.92	1.24	1.22	1.24

What to Model

- Finance
- Economics

- NASDAQ
- Dow
- S&P



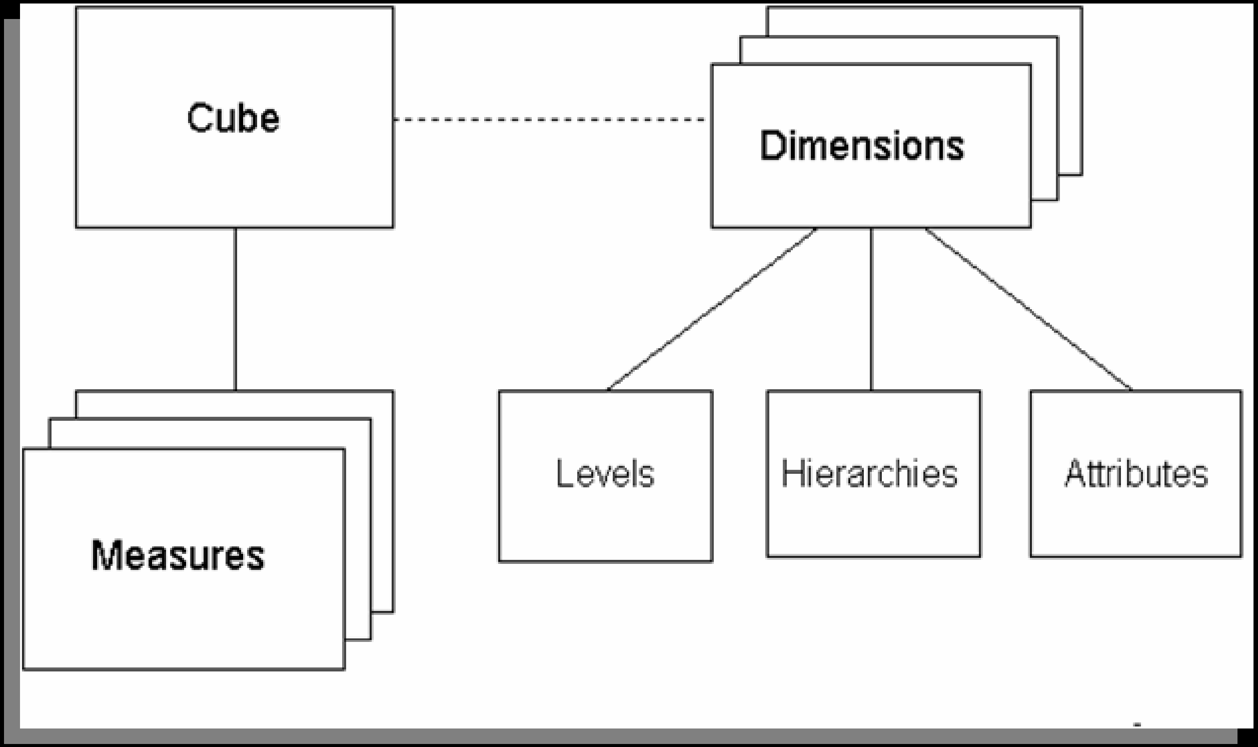
Technical Framework

- Analytical Workspace
  - Tables
  - Dimensions
    - Levels
    - Hierarchies
      - Level-Based
      - Value-Based
    - Measures

Technical Framework

- Analytical Workspace
  - Logical Cubes
  - Rules
  - Attributes

Analytical Workspace



- OLAP Registry
- Analytical Workspace Manager

Analytical SQL Syntax

ANALYTICAL SQL DML SYNTAX HIGHLIGHTS

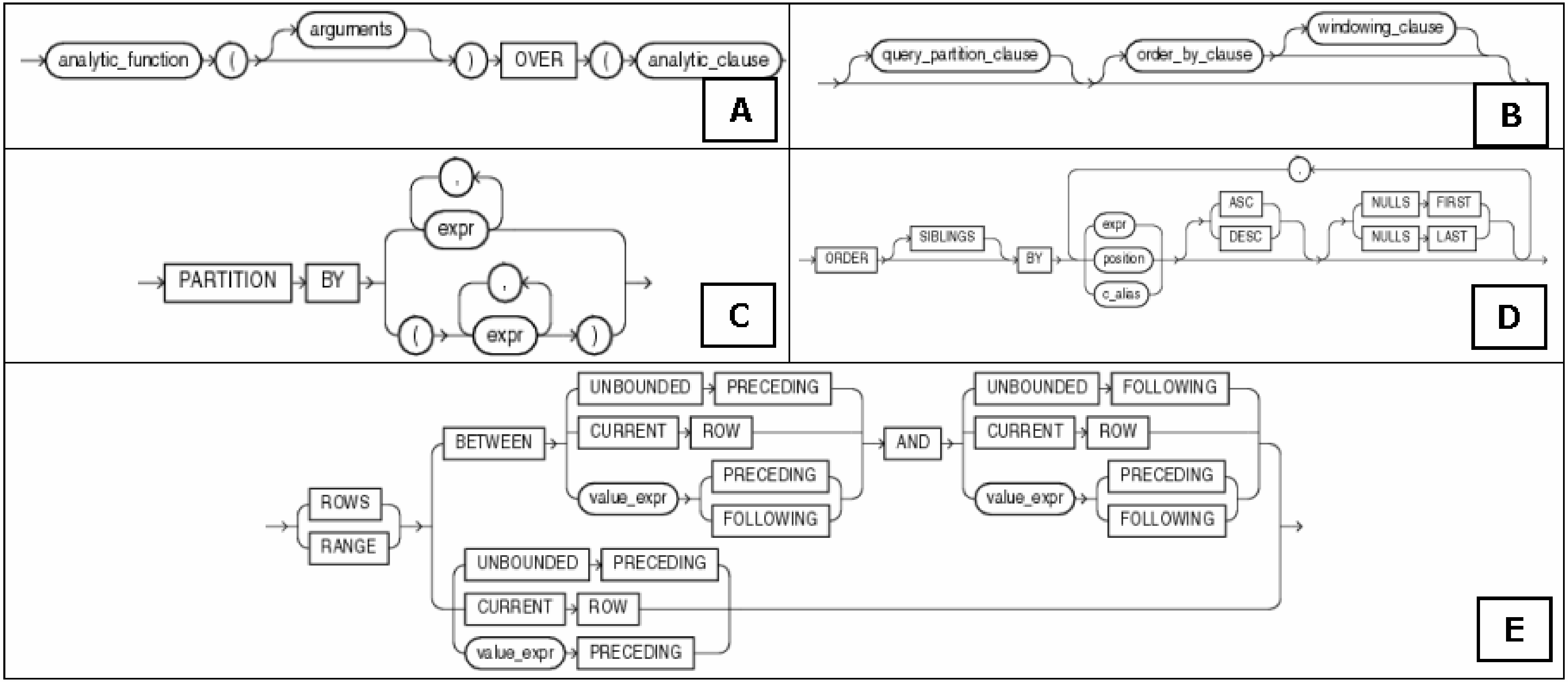
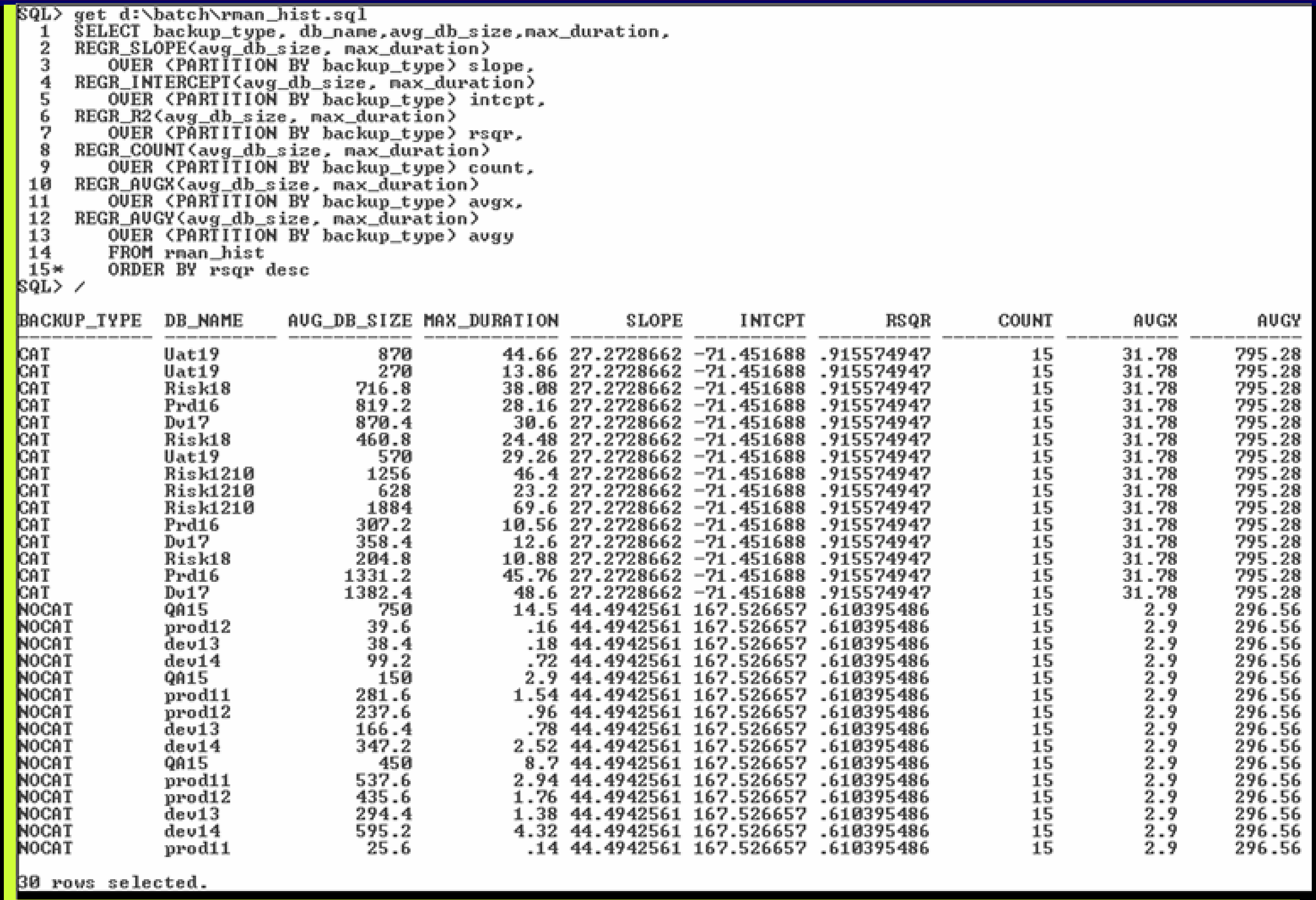


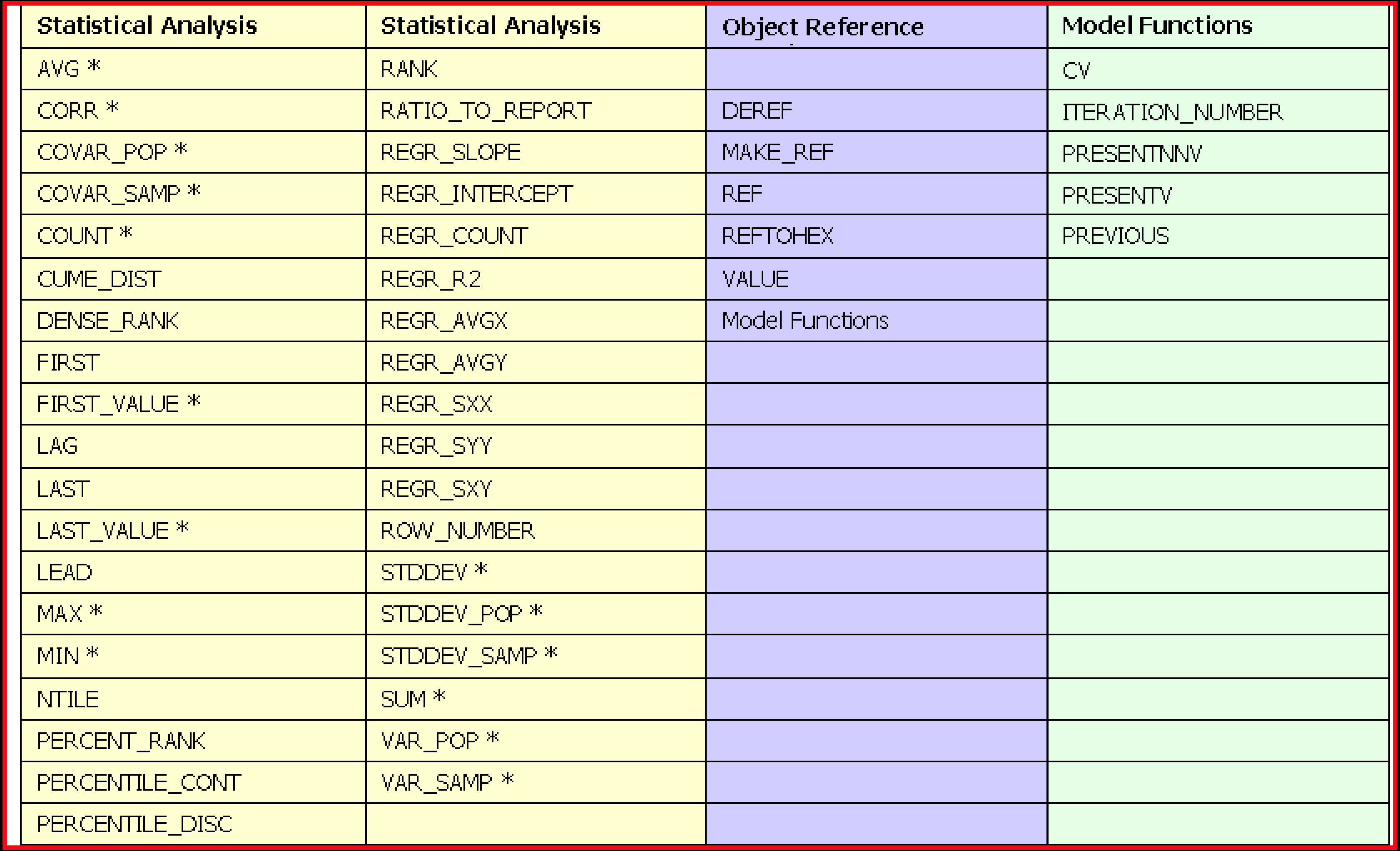
Exhibit 4. A. Analytic Function. B. Analytic Clause. C. Query Partition Clause. D. Order By Clause. E. Windowing Clause.

A Simple Regression Model



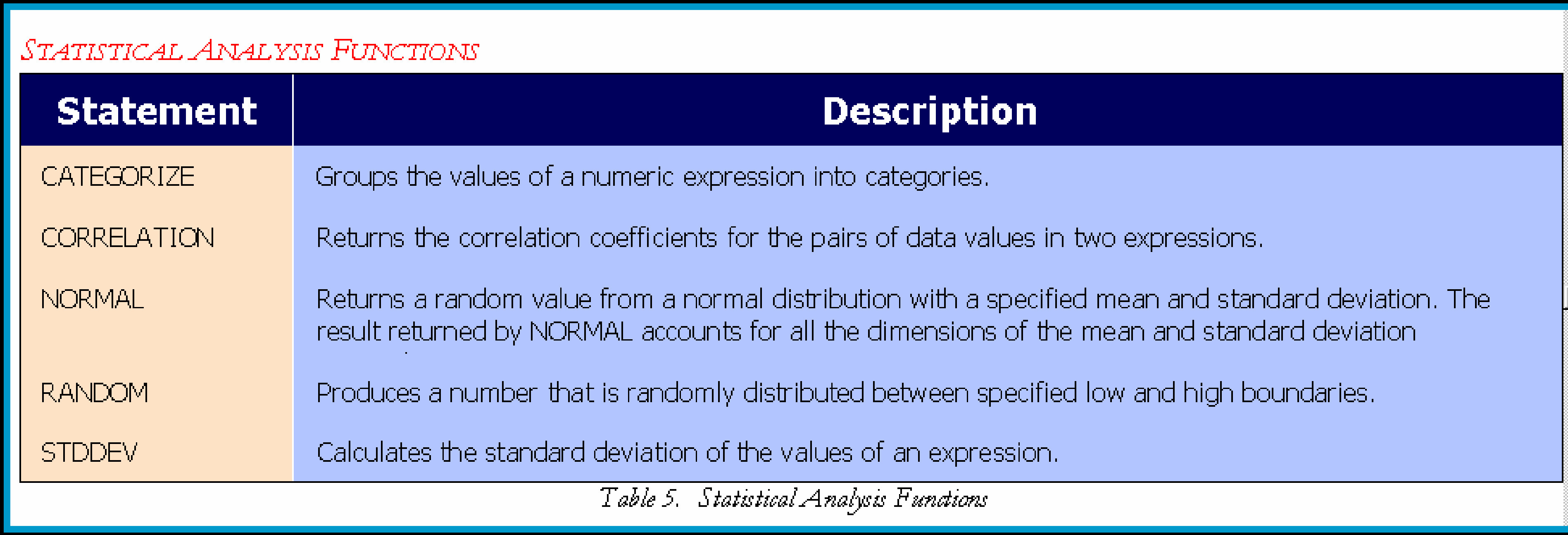
```
SQL> get d:\batch\rman_hist.sql
  SELECT backup_type, db_name,avg_db_size,max_duration,
  REGR_SLOPE(avg_db_size, max_duration)
     OVER (PARTITION BY backup_type) slope,
  REGR_INTERCEPT(avg_db_size, max_duration)
     OVER (PARTITION BY backup_type) intcpt,
  REGR_R2(avg_db_size, max_duration)
     OVER (PARTITION BY backup_type) rsqr,
  REGR_COUNT(avg_db_size, max_duration)
     OVER (PARTITION BY backup_type) count,
  REGR_AVGX(avg_db_size, max_duration)
     OVER (PARTITION BY backup_type) avgx,
  REGR_AVGY(avg_db_size, max_duration)
     OVER (PARTITION BY backup_type) avgy
     FROM rman_hist
     ORDER BY rsqr desc
SQL> /

BACKUP_TYPE  DB_NAME    AVG_DB_SIZE MAX_DURATION      SLOPE     INTCPT       RSQR      COUNT       AVGX       AVGY
------------ ---------- ----------- ------------ ---------- ---------- ---------- ---------- ---------- ----------
CAT          Uat19              870        44.66 27.2728662 -71.451688 .915574947         15      31.78     795.28
CAT          Uat19              270        13.86 27.2728662 -71.451688 .915574947         15      31.78     795.28
CAT          Risk18           716.8        38.08 27.2728662 -71.451688 .915574947         15      31.78     795.28
CAT          Prd16            819.2        28.16 27.2728662 -71.451688 .915574947         15      31.78     795.28
CAT          Dv17             870.4         30.6 27.2728662 -71.451688 .915574947         15      31.78     795.28
CAT          Risk18           460.8        24.48 27.2728662 -71.451688 .915574947         15      31.78     795.28
CAT          Uat19              570        29.26 27.2728662 -71.451688 .915574947         15      31.78     795.28
CAT          Risk1210          1256         46.4 27.2728662 -71.451688 .915574947         15      31.78     795.28
CAT          Risk1210           628         23.2 27.2728662 -71.451688 .915574947         15      31.78     795.28
CAT          Risk1210          1884         69.6 27.2728662 -71.451688 .915574947         15      31.78     795.28
CAT          Prd16            307.2        10.56 27.2728662 -71.451688 .915574947         15      31.78     795.28
CAT          Dv17             358.4         12.6 27.2728662 -71.451688 .915574947         15      31.78     795.28
CAT          Risk18           204.8        10.88 27.2728662 -71.451688 .915574947         15      31.78     795.28
CAT          Prd16           1331.2        45.76 27.2728662 -71.451688 .915574947         15      31.78     795.28
CAT          Dv17            1382.4         48.6 27.2728662 -71.451688 .915574947         15      31.78     795.28
NOCAT        QA15               750         14.5 44.4942561 167.526657 .610395486         15        2.9     296.56
NOCAT        prod12            39.6          .16 44.4942561 167.526657 .610395486         15        2.9     296.56
NOCAT        dev13             38.4          .18 44.4942561 167.526657 .610395486         15        2.9     296.56
NOCAT        dev14             99.2          .72 44.4942561 167.526657 .610395486         15        2.9     296.56
NOCAT        QA15               150          2.9 44.4942561 167.526657 .610395486         15        2.9     296.56
NOCAT        prod11           281.6         1.54 44.4942561 167.526657 .610395486         15        2.9     296.56
NOCAT        prod12           237.6          .96 44.4942561 167.526657 .610395486         15        2.9     296.56
NOCAT        dev13            166.4          .78 44.4942561 167.526657 .610395486         15        2.9     296.56
NOCAT        dev14            347.2         2.52 44.4942561 167.526657 .610395486         15        2.9     296.56
NOCAT        QA15               450          8.7 44.4942561 167.526657 .610395486         15        2.9     296.56
NOCAT        prod11           537.6         2.94 44.4942561 167.526657 .610395486         15        2.9     296.56
NOCAT        prod12           435.6         1.76 44.4942561 167.526657 .610395486         15        2.9     296.56
NOCAT        dev13            294.4         1.38 44.4942561 167.526657 .610395486         15        2.9     296.56
NOCAT        dev14            595.2         4.32 44.4942561 167.526657 .610395486         15        2.9     296.56
NOCAT        prod11            25.6          .14 44.4942561 167.526657 .610395486         15        2.9     296.56

30 rows selected.
```

Analytical Functions (SQL)



Statistical Analysis	Statistical Analysis	Object Reference	Model Functions
AVG *	RANK		CV
CORR *	RATIO_TO_REPORT	DEREF	ITERATION_NUMBER
COVAR_POP *	REGR_SLOPE	MAKE_REF	PRESENTNNV
COVAR_SAMP *	REGR_INTERCEPT	REF	PRESENTV
COUNT *	REGR_COUNT	REFTOHEX	PREVIOUS
CUME_DIST	REGR_R2	VALUE	
DENSE_RANK	REGR_AVGX	Model Functions	
FIRST	REGR_AVGY		
FIRST_VALUE *	REGR_SXX		
LAG	REGR_SYY		
LAST	REGR_SXY		
LAST_VALUE *	ROW_NUMBER		
LEAD	STDDEV *		
MAX *	STDDEV_POP *		
MIN *	STDDEV_SAMP *		
NTILE	SUM *		
PERCENT_RANK	VAR_POP *		
PERCENTILE_CONT	VAR_SAMP *		
PERCENTILE_DISC			

OLAP DML (Statistics)



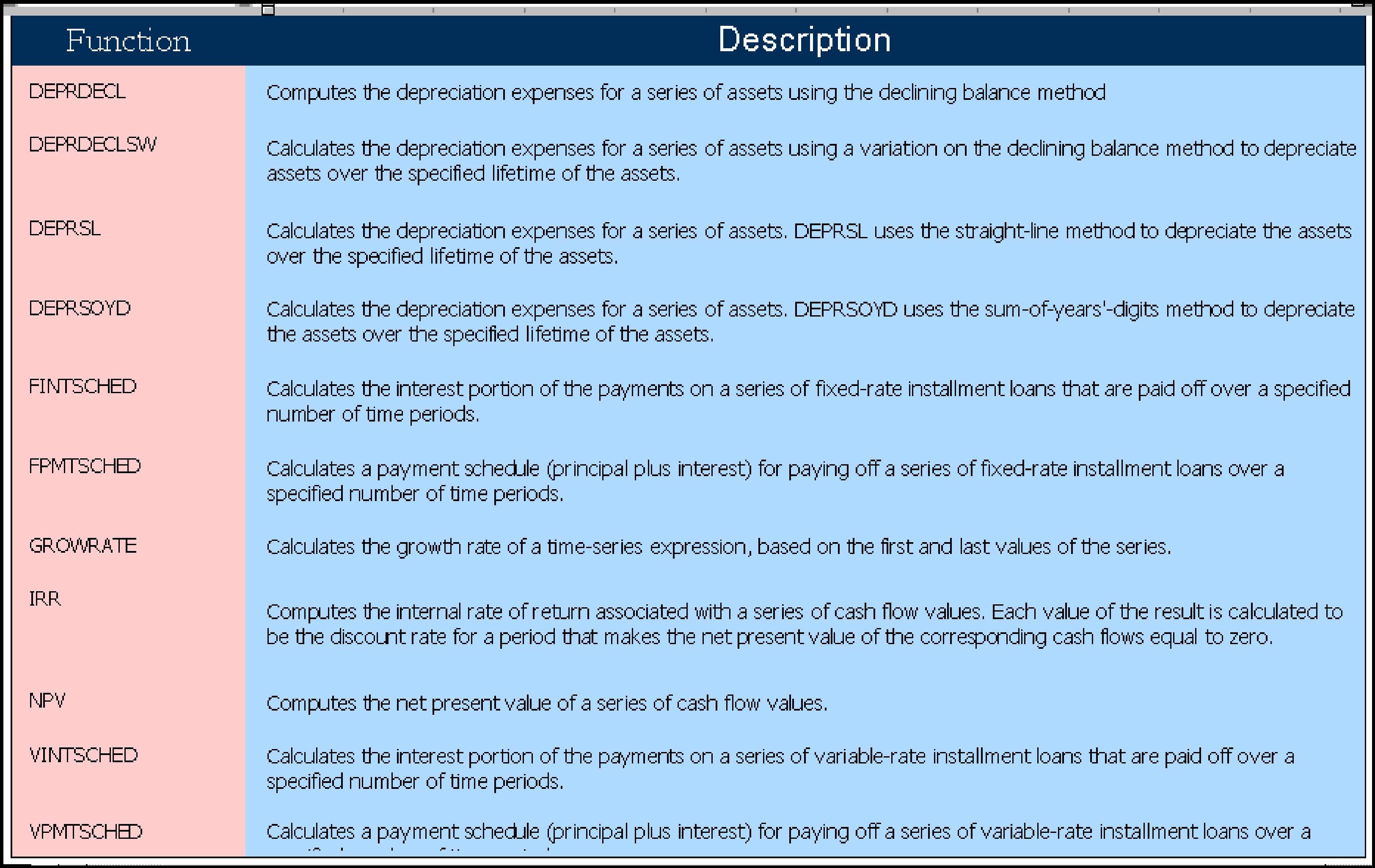
STATISTICAL ANALYSIS FUNCTIONS

Statement	Description
CATEGORIZE	Groups the values of a numeric expression into categories.
CORRELATION	Returns the correlation coefficients for the pairs of data values in two expressions.
NORMAL	Returns a random value from a normal distribution with a specified mean and standard deviation. The result returned by NORMAL accounts for all the dimensions of the mean and standard deviation
RANDOM	Produces a number that is randomly distributed between specified low and high boundaries.
STDDEV	Calculates the standard deviation of the values of an expression.

Table 5. Statistical Analysis Functions

J2EE/Java API, DBMS_AW (PL/SQL)

OLAP DML (Finance)



Function	Description
DEPRDECL	Computes the depreciation expenses for a series of assets using the declining balance method
DEPRDECLSW	Calculates the depreciation expenses for a series of assets using a variation on the declining balance method to depreciate assets over the specified lifetime of the assets.
DEPRSL	Calculates the depreciation expenses for a series of assets. DEPRSL uses the straight-line method to depreciate the assets over the specified lifetime of the assets.
DEPRSOYD	Calculates the depreciation expenses for a series of assets. DEPRSOYD uses the sum-of-years'-digits method to depreciate the assets over the specified lifetime of the assets.
FINTSCHED	Calculates the interest portion of the payments on a series of fixed-rate installment loans that are paid off over a specified number of time periods.
FPMTSCHED	Calculates a payment schedule (principal plus interest) for paying off a series of fixed-rate installment loans over a specified number of time periods.
GROWRATE	Calculates the growth rate of a time-series expression, based on the first and last values of the series.
IRR	Computes the internal rate of return associated with a series of cash flow values. Each value of the result is calculated to be the discount rate for a period that makes the net present value of the corresponding cash flows equal to zero.
NPV	Computes the net present value of a series of cash flow values.
VINTSCHED	Calculates the interest portion of the payments on a series of variable-rate installment loans that are paid off over a specified number of time periods.
VPMTSCHED	Calculates a payment schedule (principal plus interest) for paying off a series of variable-rate installment loans over a

Forecast / Regression Statements

- Simple Forecast and Regressions
  - REGRESS.report
  - SMOOTH
- Forecast and Regression Statements using Context

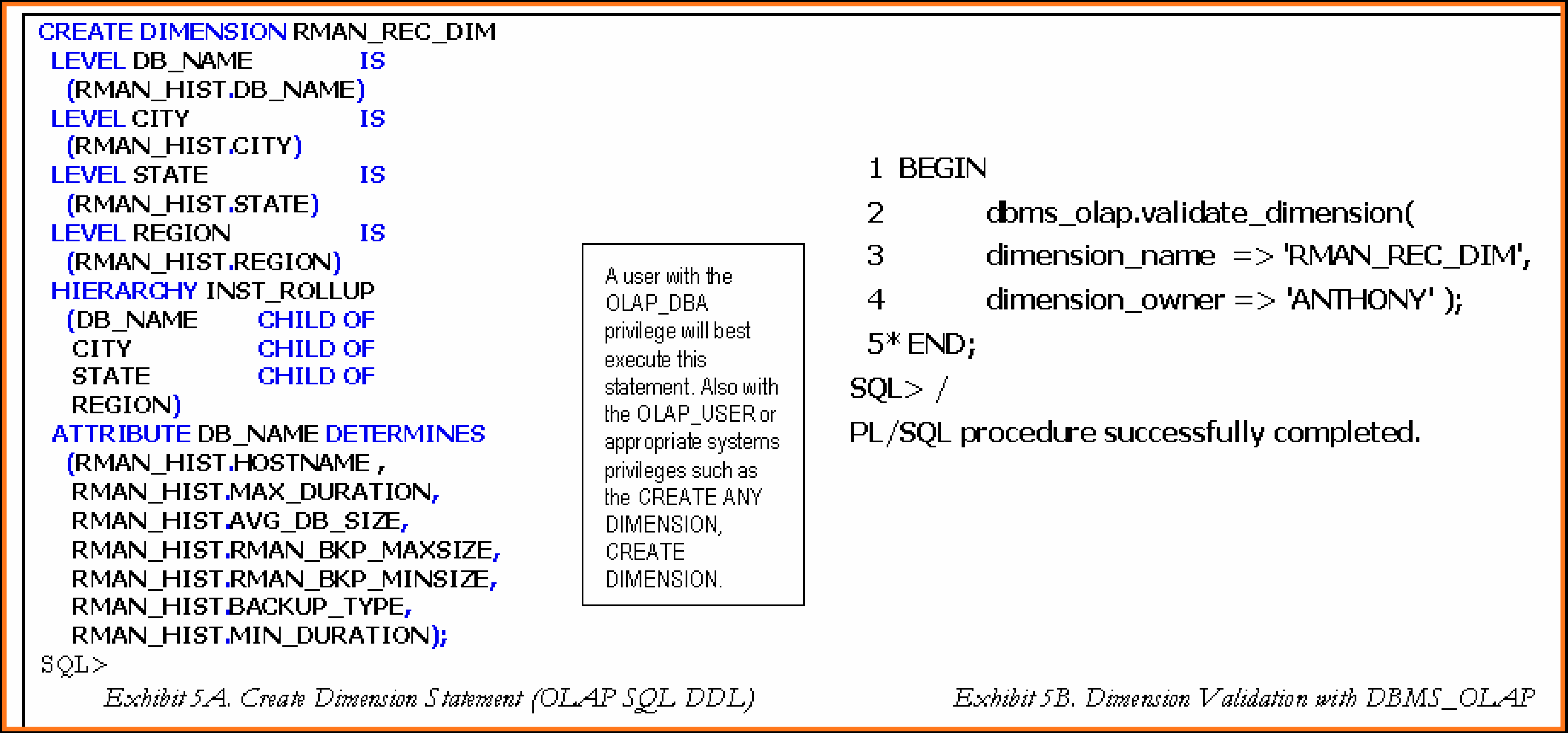
Analytical Operators

- Rollup
- Cube

Group-by Aggregation

- Aggregation using Grouping Sets
- Aggregation using Operators
- Aggregation with the Group Id and Grouping Id Differentiators.

Dimensions (MOLAP)



```
CREATE DIMENSION RMAN_REC_DIM
 LEVEL DB_NAME            IS
   (RMAN_HIST.DB_NAME)
 LEVEL CITY               IS
   (RMAN_HIST.CITY)
 LEVEL STATE              IS
   (RMAN_HIST.STATE)
 LEVEL REGION             IS
   (RMAN_HIST.REGION)
 HIERARCHY INST_ROLLUP
   (DB_NAME       CHILD OF
    CITY          CHILD OF
    STATE         CHILD OF
    REGION)
 ATTRIBUTE DB_NAME DETERMINES
   (RMAN_HIST.HOSTNAME ,
    RMAN_HIST.MAX_DURATION,
    RMAN_HIST.AVG_DB_SIZE,
    RMAN_HIST.RMAN_BKP_MAXSIZE,
    RMAN_HIST.RMAN_BKP_MINSIZE,
    RMAN_HIST.BACKUP_TYPE,
    RMAN_HIST.MIN_DURATION);
SQL>
```

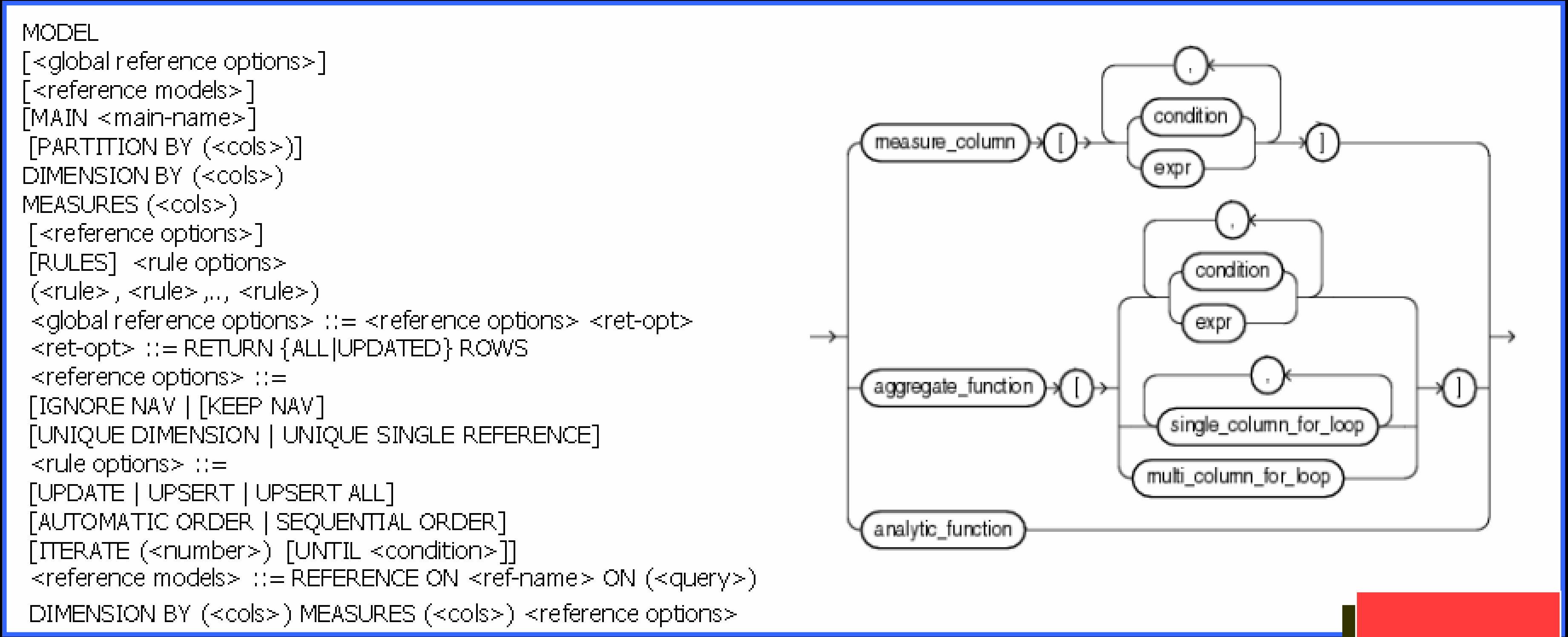
Exhibit 5A. Create Dimension Statement (OLAP SQL DDL)

A user with the OLAP_DBA privilege will best execute this statement. Also with the OLAP_USER or appropriate systems privileges such as the CREATE ANY DIMENSION, CREATE DIMENSION.

```
  1  BEGIN
  2          dbms_olap.validate_dimension(
  3          dimension_name  => 'RMAN_REC_DIM',
  4          dimension_owner => 'ANTHONY' );
  5* END;
SQL> /
PL/SQL procedure successfully completed.
```

Exhibit 5B. Dimension Validation with DBMS_OLAP

MODEL Clause



```
MODEL
[<global reference options>]
[<reference models>]
[MAIN <main-name>]
 [PARTITION BY (<cols>)]
DIMENSION BY (<cols>)
MEASURES (<cols>)
 [<reference options>]
 [RULES] <rule options>
 (<rule>, <rule>,.., <rule>)
 <global reference options> ::= <reference options> <ret-opt>
 <ret-opt> ::= RETURN {ALL|UPDATED} ROWS
 <reference options> ::=
 [IGNORE NAV | [KEEP NAV]
 [UNIQUE DIMENSION | UNIQUE SINGLE REFERENCE]
 <rule options> ::=
 [UPDATE | UPSERT | UPSERT ALL]
 [AUTOMATIC ORDER | SEQUENTIAL ORDER]
 [ITERATE (<number>) [UNTIL <condition>]]
 <reference models> ::= REFERENCE ON <ref-name> ON (<query>)
 DIMENSION BY (<cols>) MEASURES (<cols>) <reference options>
```

MODEL-Related Clauses

- Partition by
- Dimension by
- Measures
- Rules
  - Rules Options
    - Update, Upsert, Upsert all

Model Conditions in Rules

In addition to matching rules these conditions can filter the rule application:

- IS ANY
- IS PRESENT

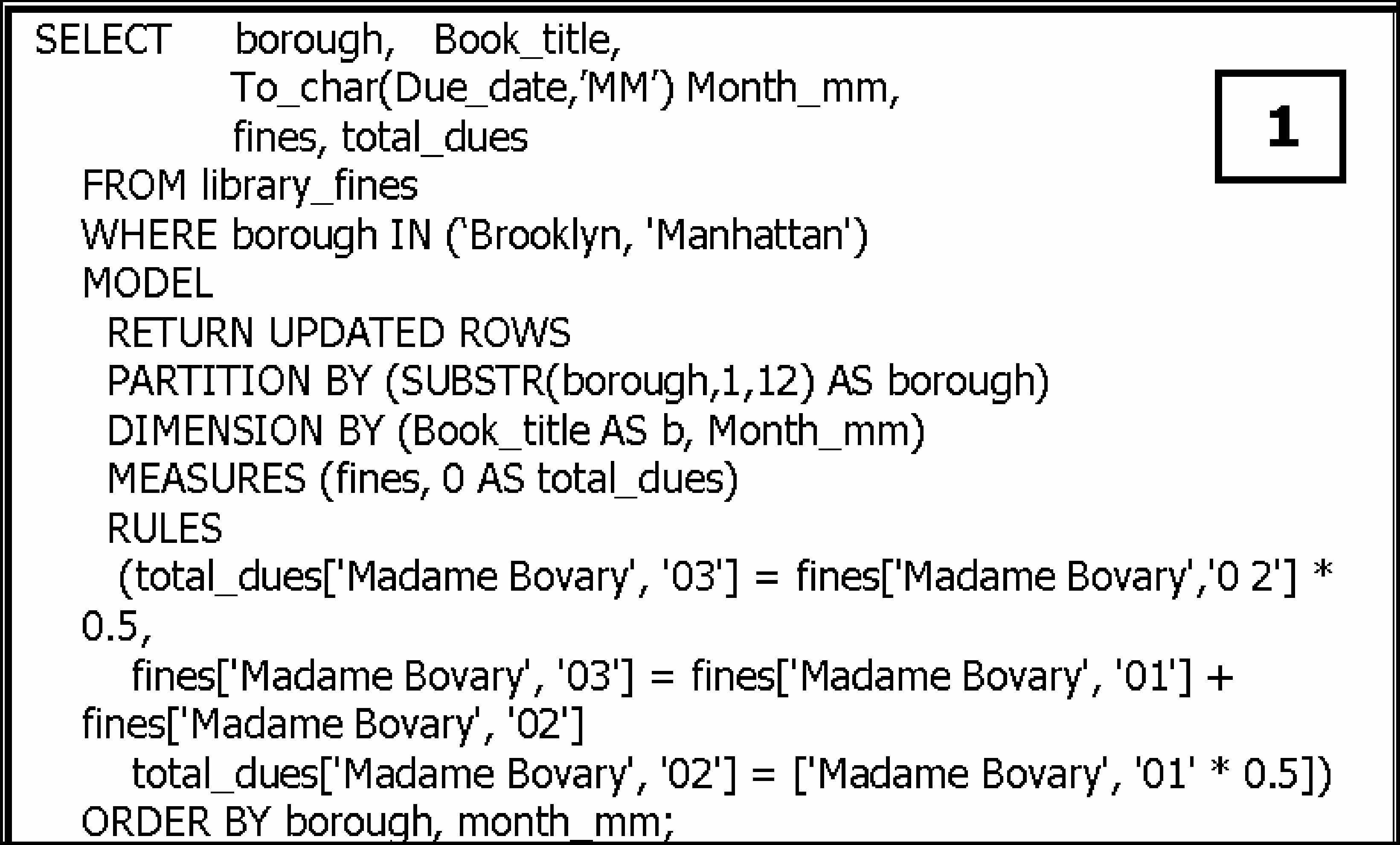
Unified Perspective

- SQL Functions and API
- OLAP DML API
  - Functions
  - Statements
- J2EE
- XML
- SOA

Current Benefits

- Analytic Workspace Support
- OLAP Registry
- Oracle’s OLAP DML J2EE/Java and PL/SQL Integration
- E-Business Support
- Hot Plug-in
- Enhanced OLAP DBA, DBA Capabilities.

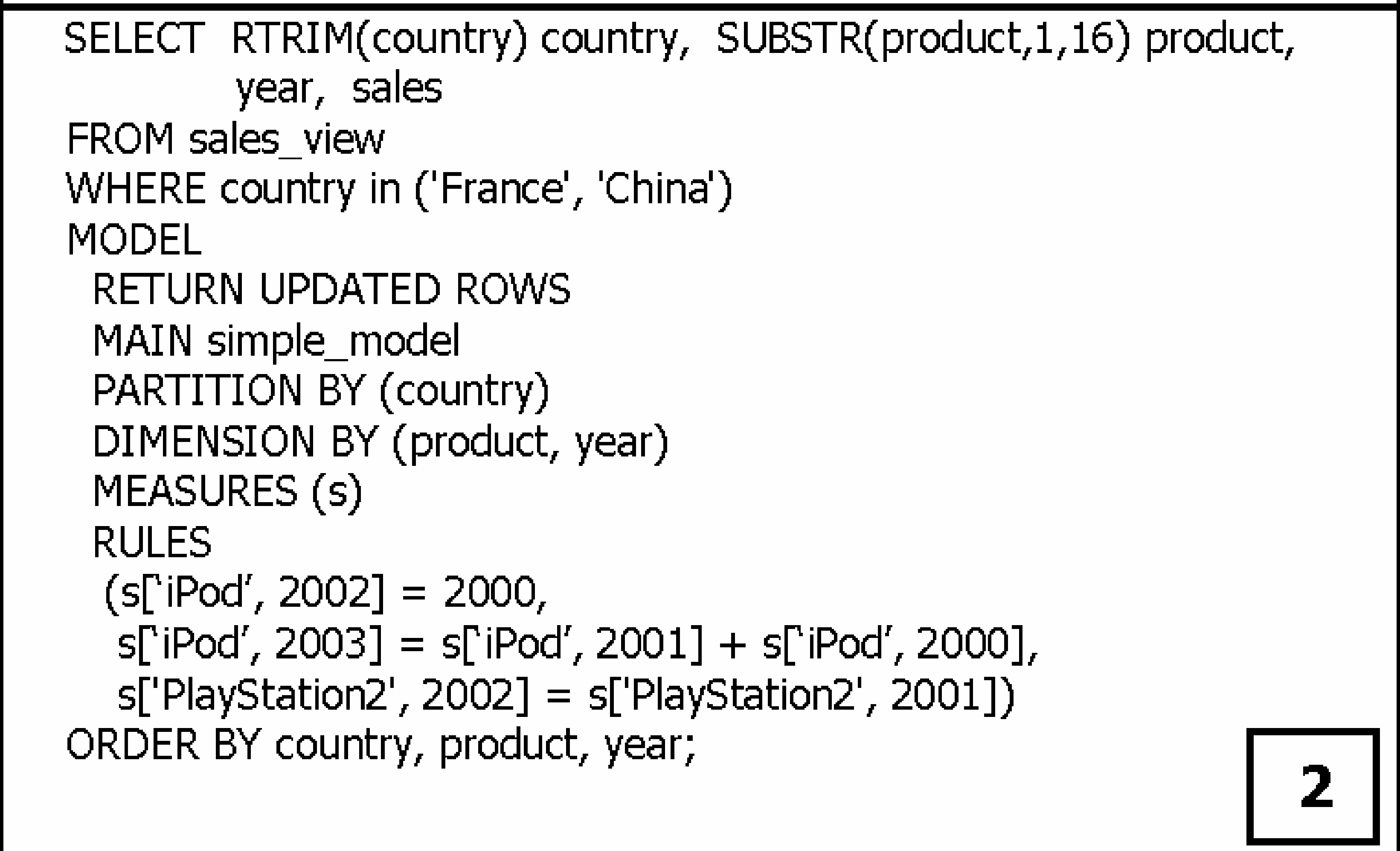
SQL Examples



```
SELECT     borough,   Book_title,
           To_char(Due_date,‘MM’) Month_mm,
           fines, total_dues
   FROM library_fines
   WHERE borough IN (‘Brooklyn, 'Manhattan')
   MODEL
     RETURN UPDATED ROWS
     PARTITION BY (SUBSTR(borough,1,12) AS borough)
     DIMENSION BY (Book_title AS b, Month_mm)
     MEASURES (fines, 0 AS total_dues)
     RULES
      (total_dues['Madame Bovary', '03'] = fines['Madame Bovary','0 2'] *
   0.5,
       fines['Madame Bovary', '03'] = fines['Madame Bovary', '01'] +
   fines['Madame Bovary', '02']
       total_dues['Madame Bovary', '02'] = ['Madame Bovary', '01' * 0.5])
   ORDER BY borough, month_mm;
```

1

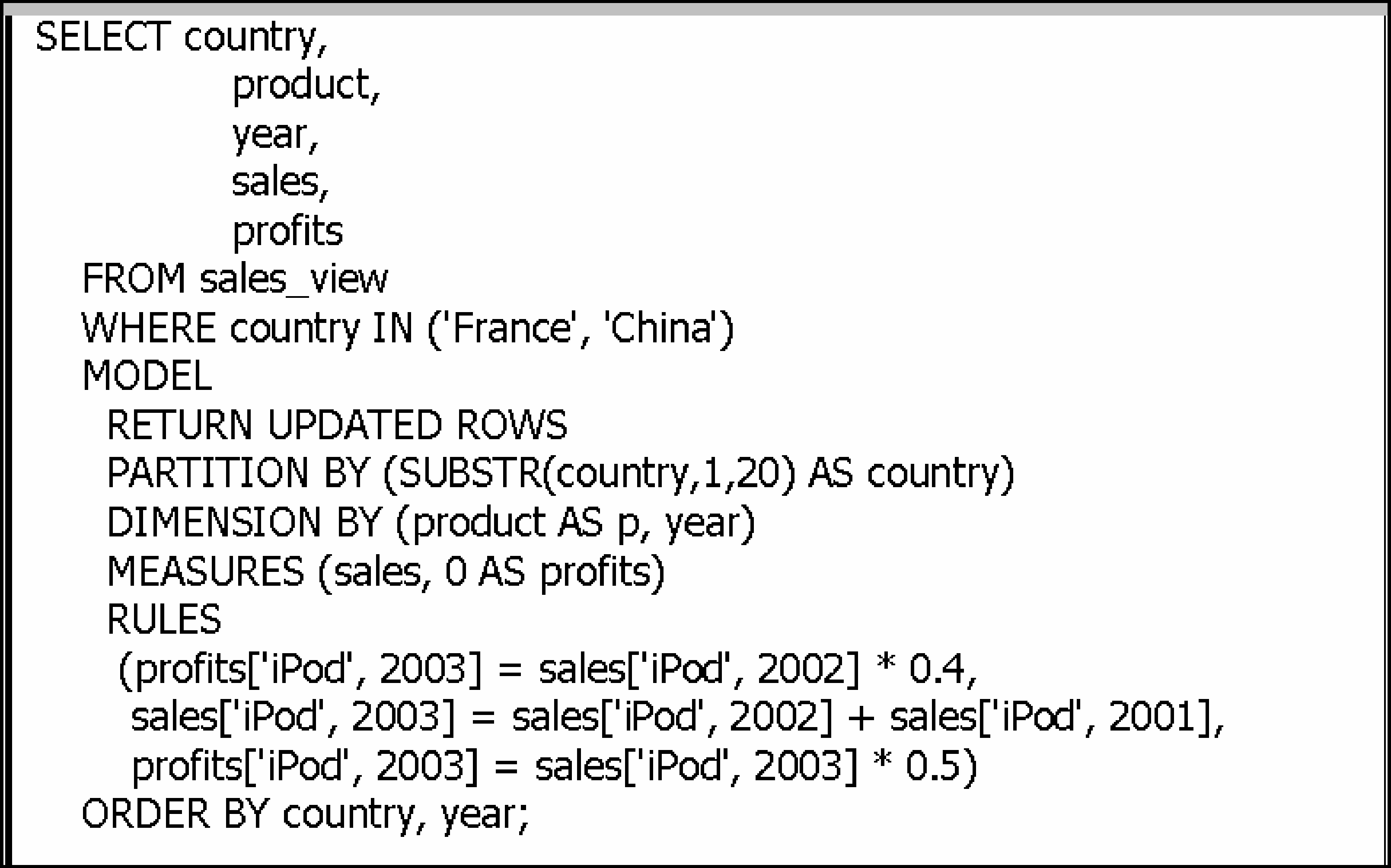
SQL Examples



```
SELECT  RTRIM(country) country,  SUBSTR(product,1,16) product,
              year,  sales
FROM sales_view
WHERE country in ('France', 'China')
MODEL
  RETURN UPDATED ROWS
  MAIN simple_model
  PARTITION BY (country)
  DIMENSION BY (product, year)
  MEASURES (s)
  RULES
   (s[‘iPod’, 2002] = 2000,
    s[‘iPod’, 2003] = s[‘iPod’, 2001] + s[‘iPod’, 2000],
    s['PlayStation2', 2002] = s['PlayStation2', 2001])
ORDER BY country, product, year;
```

2

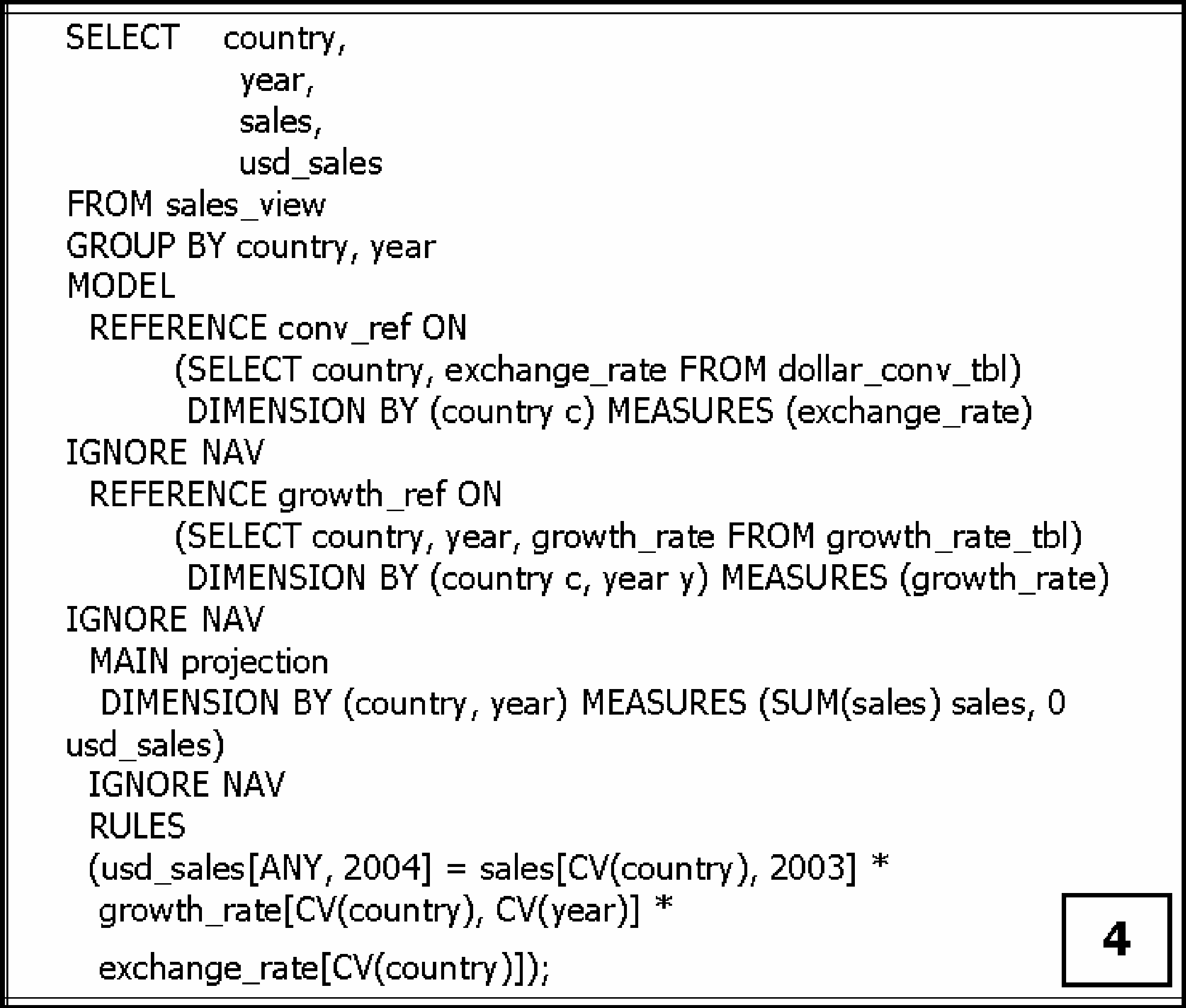
SQL Examples



```
SELECT country,
               product,
               year,
               sales,
               profits
   FROM sales_view
   WHERE country IN ('France', 'China')
   MODEL
     RETURN UPDATED ROWS
     PARTITION BY (SUBSTR(country,1,20) AS country)
     DIMENSION BY (product AS p, year)
     MEASURES (sales, 0 AS profits)
     RULES
      (profits['iPod', 2003] = sales['iPod', 2002] * 0.4,
       sales['iPod', 2003] = sales['iPod', 2002] + sales['iPod', 2001],
       profits['iPod', 2003] = sales['iPod', 2003] * 0.5)
   ORDER BY country, year;
```

3

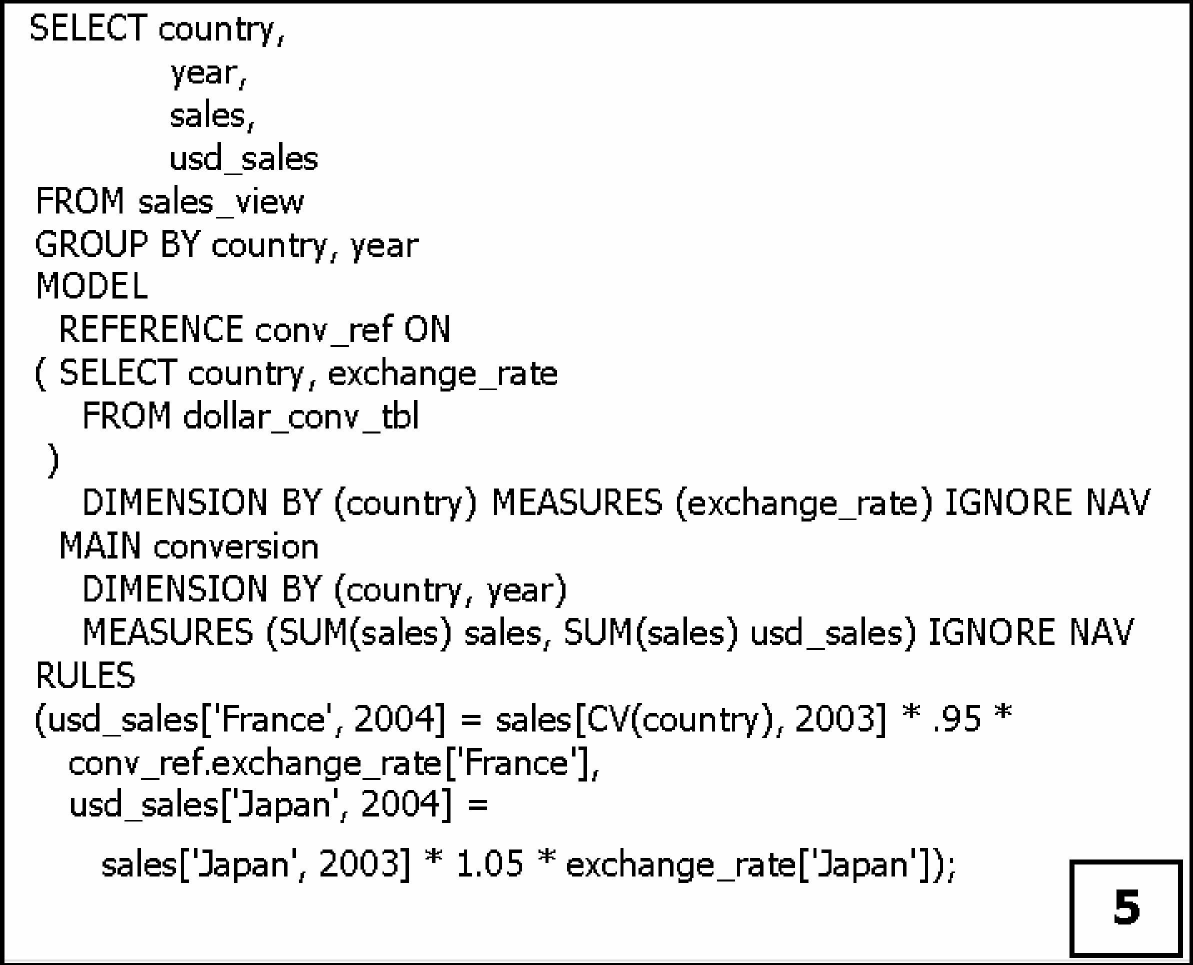
SQL Examples



```
SELECT   country,
          year,
          sales,
          usd_sales
FROM sales_view
GROUP BY country, year
MODEL
  REFERENCE conv_ref ON
          (SELECT country, exchange_rate FROM dollar_conv_tbl)
           DIMENSION BY (country c) MEASURES (exchange_rate)
IGNORE NAV
  REFERENCE growth_ref ON
          (SELECT country, year, growth_rate FROM growth_rate_tbl)
           DIMENSION BY (country c, year y) MEASURES (growth_rate)
IGNORE NAV
  MAIN projection
   DIMENSION BY (country, year) MEASURES (SUM(sales) sales, 0
usd_sales)
  IGNORE NAV
  RULES
  (usd_sales[ANY, 2004] = sales[CV(country), 2003] *
   growth_rate[CV(country), CV(year)] *
   exchange_rate[CV(country)]);
```

4

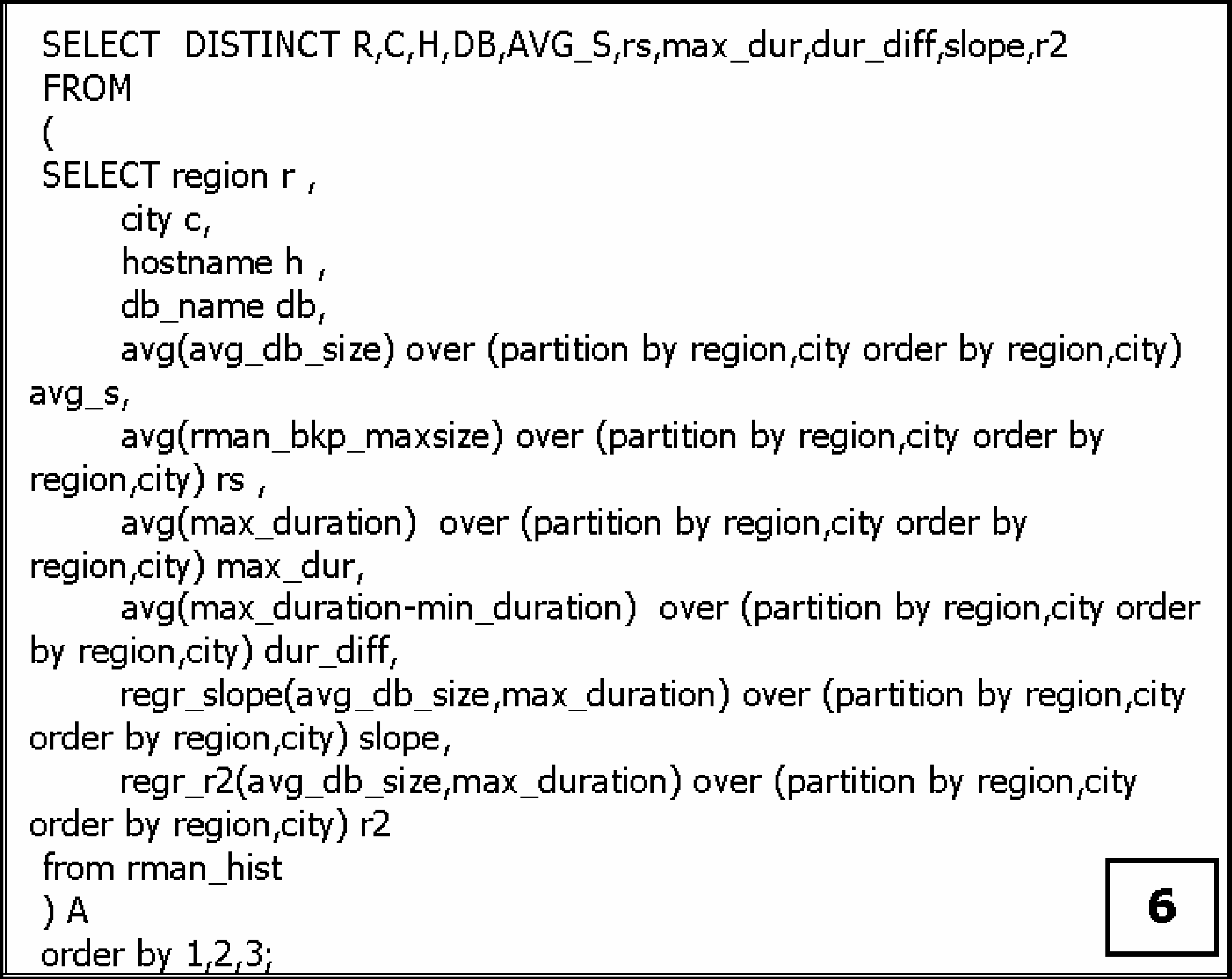
SQL Examples



```
SELECT country,
              year,
              sales,
              usd_sales
 FROM sales_view
 GROUP BY country, year
 MODEL
   REFERENCE conv_ref ON
 ( SELECT country, exchange_rate
     FROM dollar_conv_tbl
  )
     DIMENSION BY (country) MEASURES (exchange_rate) IGNORE NAV
   MAIN conversion
     DIMENSION BY (country, year)
     MEASURES (SUM(sales) sales, SUM(sales) usd_sales) IGNORE NAV
 RULES
 (usd_sales['France', 2004] = sales[CV(country), 2003] * .95 *
    conv_ref.exchange_rate['France'],
    usd_sales['Japan', 2004] =

       sales['Japan', 2003] * 1.05 * exchange_rate['Japan']);
```

5

SQL Examples



```
 SELECT  DISTINCT R,C,H,DB,AVG_S,rs,max_dur,dur_diff,slope,r2
 FROM
 (
 SELECT region r ,
        city c,
        hostname h ,
        db_name db,
        avg(avg_db_size) over (partition by region,city order by region,city)
avg_s,
        avg(rman_bkp_maxsize) over (partition by region,city order by
region,city) rs ,
        avg(max_duration)  over (partition by region,city order by
region,city) max_dur,
        avg(max_duration-min_duration)  over (partition by region,city order
by region,city) dur_diff,
        regr_slope(avg_db_size,max_duration) over (partition by region,city
order by region,city) slope,
        regr_r2(avg_db_size,max_duration) over (partition by region,city
order by region,city) r2
 from rman_hist
 ) A
 order by 1,2,3;
```

6

Concluding Remarks

- Oracle SQL analytical framework has attained an outstanding level of maturity deserving a leadership positioning.
- Oracle SQL and OLAP API can integrate with custom J2EE, XML, and .net applications, and for a key reporting Data Warehousing capability with Text and related tools.

Concluding Remarks

- SQL Model capabilities have achieved a transactional and reporting level which allows the visualization of modeling and forecasting in query-like mode, with greater flexibility than conventional statistical analysis and modeling tools.
- Oracle10g support through both OLAP DML and API and SQL is essentially solid for conventional forecasting methods such as Time Series (Moving Average and ARIMA) and Exponential Smoothing, among others.

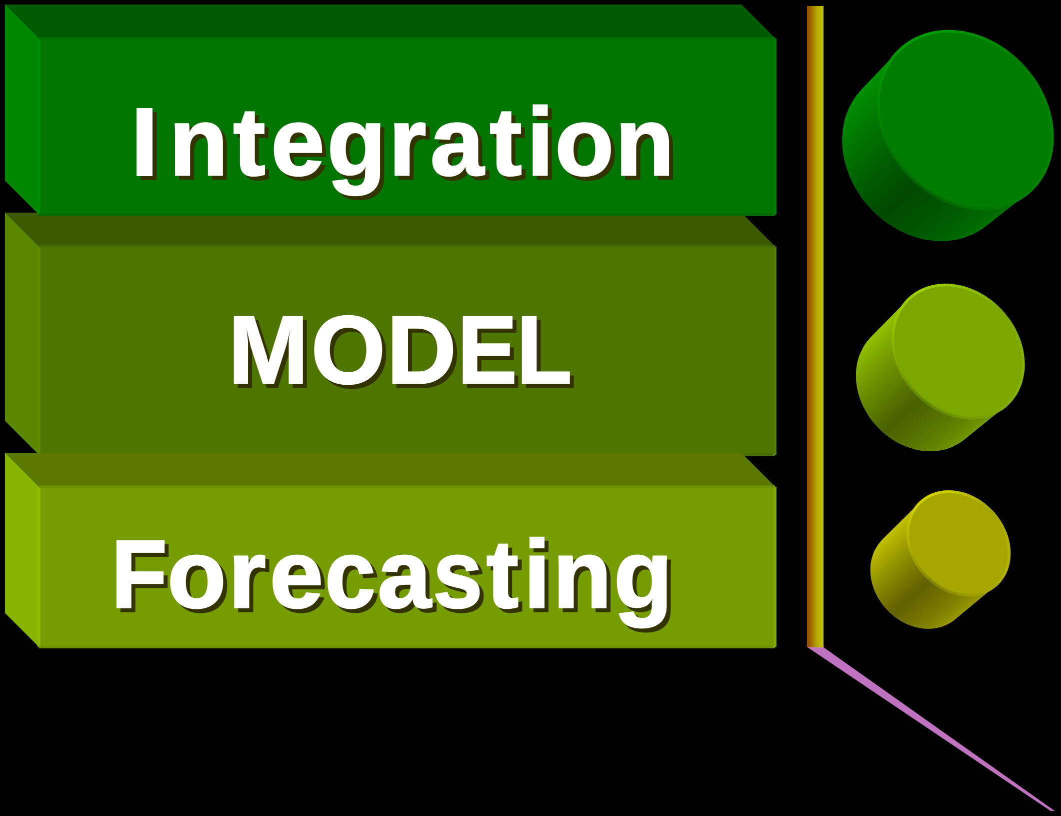
Future Forecasting Models

- Who
- Where
- When
- Extent
- How
- How much

Case Studies

- Health (HIV, AIDS)
- Demographics (Census)
- Financial Markets (Nasdaq)
- Econometrics (Income/Gender)
- Marketing Research (4Ps)

Concluding Remarks



Integration

MODEL

Forecasting