DW & Big Data on your smartphone

Smartphone app helping you get the most from this year’s OpenWorld

Access to all the most important information
• Presenter profiles
• Must-see sessions
• Must-attend hands-on labs
• Useful links

http://tinyurl.com/kmbsxbu
Using Analytical SQL to Intelligently Explore Big Data

Joerg Otto, Head of Database Engineering
IDS GmbH - Analysis and Reporting Services

Marty Gubar
Director Product Manager

Keith Laker, Oracle
Senior Principal Product Manager
Safe Harbor Statement

The following is intended to outline our general product direction. It is intended for information purposes only, and may not be incorporated into any contract. It is not a commitment to deliver any material, code, or functionality, and should not be relied upon in making purchasing decisions. The development, release, and timing of any features or functionality described for Oracle’s products remains at the sole discretion of Oracle.
Agenda

1. Analytical SQL – simplicity and sophistication
2. Using analytical SQL to solve complex business problems
3. 12c – faster and smarter
4. Big Data SQL – analyze all your data
5. Summary
Analytical SQL – simplicity and sophistication
The On-Going Evolution of SQL

- **8i**
  - Introduction of "window" functions

- **9i**
  - Enhanced window functions (percentile, etc)
  - Rollup, grouping sets, cube

- **10g**
  - Statistical functions
  - SQL model clause
  - Partition Outer Join
  - Data mining

- **11g**
  - SQL Pivot
  - Recursive WITH
  - ListAgg, Nth value window

- **12c**
  - Pattern matching
  - Top N clause
  - Approx Count distinct
  - JSON support
The On-Going Evolution of SQL

- Introduction of “window” functions
- Enhanced window functions (percentile, etc)
- Rollup, grouping sets, cube
- Statistical functions
- Partition Outer Join
- Data mining
- SQL Pivot
- Recursive WITH
- ListAgg, Nth value window
- Pattern matching
- Top N clause
- Approx Count distinct
- JSON support

Oracle Big Data SQL
Using analytical SQL to solve complex business problems

Joerg Otto,
Head of Database Engineering
IDS GmbH – Analysis and Reporting Services
Allianz Group At A Glance

- Insurance, Asset Management and Banking
- Founded 1890, headquarters in Munich / Germany
- 78 Million Customers in more than 70 countries
- > 147,000 Employees in (2013)
- €110 Billion Revenue (2013)
- €6.3 Billion Net income (2012)
Analytical Functions – How we use it

- What is most relevant investment metadata information?
  - Rank

- What is the 3-month moving average of stock price?
  - Moving Window

- What is the percentage growth of 2010 premiums written over 2009?
  - Period-over-period comparisons

- What are January’s net income as a percentage of the entire year’s?
  - Compare aggregates on different levels

- What are deviations between delivered and market data?
  - with_bucket, standard deviation

- What is the ratio of an e.g. investment’s market value in a fund?
  - Share holdings calculations
Analytical Functions – Types

• LAG/LEAD functions
  • Direct inter-row reference using offsets

• Ranking
  • cume_dist, rank, dense_rank, percent_rank, ntile

• Reporting Aggregate
  • sum, avg, min, max, variance, stddev, count, ratio_to_report

• Statistical Aggregates
  • correlation, linear regression, covariance

• Window Aggregate
  • min, max, count, avg, sum, variance, stddev, first_value, last_value
Analytical Functions – LEAD example

```
SELECT imdi.imdi_anchor_id,
    imdi.imdi_valuation_date AS valid_from,
    LEAD(imdi.imdi_valuation_date)
    OVER(PARTITION BY imdi.imdi_anchor_id
            ORDER BY imdi.imdi_valuation_date) - 1 AS valid_until,
    imdi.imdi_f_asset_val_net AS net_asset_val
FROM ids.inmd_market_data_inv imdi
WHERE imdi.imdi_anchor_id = 'AN00054467'
```

Example:
- delivers a time series for validity date
Analytical Functions – RANK

- Prioritization of rows and rows in data sets (in window functions)
- Used to build waterfalls for views

SELECT *
FROM (SELECT imdi.imdi_anchor_id,
         imdi.imdi_valuation_date,
         RANK() OVER(PARTITION BY imdi.imdi_anchor_id
                    ORDER BY imdi.imdi_valuation_date DESC) AS rk,
         imdi.imdi_f_asset_val_net
     FROM ids.inmd_market_data_inv imdi
     WHERE imdi.imdi_anchor_id = 'AN00054467'
     )
WHERE rk = 1;

Example: delivers latest net asset value of an investment

<table>
<thead>
<tr>
<th>Description</th>
<th>Cost</th>
<th>Object owner</th>
<th>Object name</th>
<th>Cardinality</th>
<th>Bytes</th>
<th>CP</th>
</tr>
</thead>
<tbody>
<tr>
<td>SELECT STATEMENT, GOAL = ALL_ROWS</td>
<td>39</td>
<td></td>
<td></td>
<td>103</td>
<td>7519</td>
<td>14</td>
</tr>
<tr>
<td>VIEW</td>
<td>39</td>
<td>SYS</td>
<td></td>
<td>103</td>
<td>7519</td>
<td>14</td>
</tr>
<tr>
<td>WINDOW SORT PUSHED RANK</td>
<td>39</td>
<td></td>
<td></td>
<td>103</td>
<td>2163</td>
<td>14</td>
</tr>
<tr>
<td>TABLE ACCESS BY INDEX ROWID</td>
<td>38</td>
<td>IDS</td>
<td>INMD_MARKET_DATA_INV</td>
<td>103</td>
<td>2163</td>
<td>50</td>
</tr>
<tr>
<td>INDEX RANGE SCAN</td>
<td>3</td>
<td>IDS</td>
<td>IMDI_UK</td>
<td>103</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Analytical Functions – RATIO_TO_REPORT

- Computes the ratio of a value to the sum of a data set of values
- Used to calculate breakdowns and distributions of investment classes

WITH
summe AS (  
SELECT valid_until, cons_unit,  
    SUM(exposure_eur) exp_eur,  
    SUM(exposure_por) exp_por,  
    FROM vo_a_positions_aggregated  
WHERE ...  
GROUP BY ...)

SELECT e.valid_until,  
e.cons_unit,  
e.scur_code,  
e.val,  
100 * SUM(e.exposure_eur) / s.exp_eur  
100 * SUM(e.exposure_por) / s.exp_por  
FROM vo_a_positions_aggregated e,  
summe s  
WHERE e.valid_until = s.valid_until  
AND e.cons_unit = s.cons_unit  
GROUP BY ...

Example:
- delivers latest exposure distribution of subfonds

<table>
<thead>
<tr>
<th>VAL_DATE</th>
<th>CONS_UNIT</th>
<th>SUB_UNIT</th>
<th>%</th>
<th>EXPOR_EUR</th>
</tr>
</thead>
<tbody>
<tr>
<td>30.06.2012</td>
<td>Tb2</td>
<td>Ixxx2</td>
<td>100</td>
<td>18361518</td>
</tr>
<tr>
<td>30.06.2012</td>
<td>Tb3</td>
<td>FJP</td>
<td>3,8045</td>
<td>694416</td>
</tr>
<tr>
<td>30.06.2012</td>
<td>Tb3</td>
<td>AUD</td>
<td>11,1122</td>
<td>2028351</td>
</tr>
<tr>
<td>30.06.2012</td>
<td>Tb3</td>
<td>RUD</td>
<td>28,4839</td>
<td>5199641</td>
</tr>
<tr>
<td>30.06.2012</td>
<td>Tb3</td>
<td>ACH</td>
<td>3,2193</td>
<td>387632</td>
</tr>
<tr>
<td>30.06.2012</td>
<td>Tb3</td>
<td>RCH</td>
<td>29,4063</td>
<td>5367654</td>
</tr>
<tr>
<td>30.06.2012</td>
<td>Tb3</td>
<td>EUV</td>
<td>23,5506</td>
<td>4298778</td>
</tr>
<tr>
<td>30.06.2012</td>
<td>Tb3</td>
<td>AJP</td>
<td>0,4213</td>
<td>76006</td>
</tr>
<tr>
<td>30.06.2012</td>
<td>Tb3</td>
<td>ITL</td>
<td>100</td>
<td>3122327</td>
</tr>
</tbody>
</table>
Analytical Functions – Moving time window example

SELECT AVG(last_price) over (PARTITION BY inv_id, exchange, vendor ORDER BY val_date RANGE BETWEEN INTERVAL '3' MONTH PRECEDING AND INTERVAL '1' DAY PRECEDING) avg_price_month...
Analytical Functions – Aggregates example

### ROLLUP

<table>
<thead>
<tr>
<th>SPOK ID</th>
<th>Attr. Type</th>
<th>Marketvalue Sum</th>
</tr>
</thead>
<tbody>
<tr>
<td>4547</td>
<td>A</td>
<td>-2,434,690</td>
</tr>
<tr>
<td>4547</td>
<td>C</td>
<td>-13,638,163</td>
</tr>
<tr>
<td>4547</td>
<td>F</td>
<td>783,300.073</td>
</tr>
<tr>
<td>4547</td>
<td>S</td>
<td>2,165,730.474</td>
</tr>
<tr>
<td>4547</td>
<td>X</td>
<td>-943,969</td>
</tr>
<tr>
<td></td>
<td>Total</td>
<td>2,227,013.724</td>
</tr>
</tbody>
</table>

### CUBE

<table>
<thead>
<tr>
<th>Attr. Type</th>
<th>4551</th>
<th>4547</th>
<th>Total</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>-1,971,228</td>
<td>-2,434,690</td>
<td>-4,405,918</td>
</tr>
<tr>
<td>C</td>
<td>32,572,552</td>
<td>-13,638,163</td>
<td>18,934,389</td>
</tr>
<tr>
<td>F</td>
<td></td>
<td>78,300.073</td>
<td>78,300.073</td>
</tr>
<tr>
<td>S</td>
<td>714,799.433</td>
<td>2,165,730.474</td>
<td>2,880,529.907</td>
</tr>
<tr>
<td>X</td>
<td>-24.653</td>
<td>-943.969</td>
<td>-968.622</td>
</tr>
<tr>
<td>Total</td>
<td>745,376.105</td>
<td>2,227,013.724</td>
<td>2,972,389.829</td>
</tr>
</tbody>
</table>

- Used to classify/aggregate data for types of investments (e.g., Stocks, fund, derivatives etc.) / “Excel” in the database
- Used to aggregate data for portfolio hierarchies
12c – faster and smarter

- SQL Pattern Matching
- Approximate Count Distinct
Pattern Recognition In Sequences of Rows

SQL Pattern Matching - Concepts

• Recognize patterns in sequences of events using SQL
  – Sequence is a stream of rows
  – Event equals a row in a stream

• New SQL construct MATCH_RECOGNIZE
  – Logically partition and order the data
  – Pattern defined with regular expressions via variables
    • Regular expression matched against a sequence of rows (forwards/backwards)
    • Each pattern variable is defined using conditions on rows and aggregates
Pattern Matching With Oracle SQL
Simplifies Development and Application Code: SQL vs. Java

Searching for double bottom (w-style) patterns in stock market data

```
SELECT first_x, last_z
FROM ticker
MATCH_RECOGNIZE (
    PARTITION BY stock_id ORDER BY timestamp
    MEASURES FIRST(x.time) AS first_x,
    LAST(z.time) AS last_z
    ONE ROW PER MATCH
    PATTERN (X+ Y+ W+ Z+)
    DEFINE X AS (price < PREV(price)),
    Y AS (price > PREV(price)),
    W AS (price < PREV(price)),
    Z AS (price > PREV(price)) AND
    z.time - FIRST(x.time) <= 7 )
```

250+ Lines of Java UDF
20x less code
Getting An *Approximate* Answer From a Query

*When “good enough” is good enough*

- **Business Problems**
  - Not every query requires a completely accurate result
    - Trending, data discovery, social analysis
  - Exact processing of large data sets not economical
  - For interactive analysis, sufficient accuracy of a query not known at start
    - Accuracy dependent on result because of “Think time” before next analysis step

- **Solutions**
  - Provide “approximate result” capabilities in SQL
    - Guided by user-controllable intentions: maximum runtime, maximum accuracy, number of iterations
  - Framework allows data analysis accuracy to be refined progressively
Big Data SQL – Analyze All Data

- Oracle SQL Applied to All Data
- Oracle Database, Hadoop and NoSQL*

* NoSQL Coming Soon!
Oracle Big Data SQL

Oracle SQL on all your data.

Oracle SQL on Hadoop and beyond
  • With a Smart Scan service inspired by Exadata
  • With native SQL operators
  • With the security of Oracle Database
Intelligent Query Optimization

Exadata: Applies SmartScan Close to the Data

Query Data in RDBMS

Oracle SQL

Exadata

Oracle Exadata Storage Server

Oracle Exadata Storage Server
Intelligent Query Optimization

Exadata & Big Data SQL: Applies SmartScan Close to All Data

Query Data in RDBMS and Hadoop

Oracle SQL

Exadata

Big Data Appliance

Fast

Massive Parallelism

Filtered Locally

Minimized Data Movement
Oracle Big Data SQL Demonstration

Securely analyze customer behavior and sales transactions for targeted marketing

Customer behavioral data from weblogs stored as JSON documents

WEBLOG
{"logid":"L23588999",
"customerid":"richard.obrien@mail.com",
"platform":"Desktop",
"eventdate":"11/6/12",
"httpcode":"200",
"brandname":"Coca-Cola",
"ipaddress":"167.134.198.24",
"qty":1}
Demonstration

Easily identify customers for a targeted marketing campaign: those who are active users of our site but not spending money

• Parse complex JSON documents
• Preserve anonymity of customer identities
• Applied Analytic SQL over data sourced from both Oracle Database tables and Hadoop
• **Fast!** Pushed JSON parsing and row filtering close to the data – using Smart Scan on the Big Data Appliance
Summary
Key Benefits of Analyzing Big Data with SQL

**Highlights**

Key benefits provided by Oracle’s analytical SQL for Big Data:
1. Increased performance
2. Enhanced developer productivity
3. Improved manageability
4. Minimized learning effort
5. Investment protection (through ANSI SQL compliance)
Where to get more information

- SQL Analytics Home Page on OTN
- Big Data Home Page on OTN
- Oracle Learning Library

- Big Data, Data Warehouse and SQL Analytics blog

Follow us on social media
**DW and Big Data Demo Booths**

Come and visit us in the Moscone South Exhibition Hall

<table>
<thead>
<tr>
<th></th>
<th>Regular Hours</th>
<th>Dedicated Hours</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Monday</strong></td>
<td>9:45 a.m.–6:00 p.m.</td>
<td>9:45 a.m.–10:15 a.m.</td>
</tr>
<tr>
<td><strong>Tuesday</strong></td>
<td>10:00 a.m.–6:00 p.m.</td>
<td>10:00 a.m.–10:45 a.m.</td>
</tr>
<tr>
<td><strong>Wednesday</strong></td>
<td>9:30 a.m.–3:30 p.m.</td>
<td>9:30 a.m.–10:15 a.m.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>2:45 p.m.–3:30 p.m.</td>
</tr>
</tbody>
</table>
### DW and Big Data Sessions @OOW2014

#### Monday

<table>
<thead>
<tr>
<th>Time</th>
<th>Session Title</th>
<th>Location</th>
</tr>
</thead>
<tbody>
<tr>
<td>2:45pm</td>
<td><em>Oracle Big Data: Strategy and Roadmap</em></td>
<td>South 104</td>
</tr>
<tr>
<td>4:00pm</td>
<td><em>Using Analytical SQL to Intelligently Explore Big Data</em></td>
<td>North 131</td>
</tr>
<tr>
<td>5:15pm</td>
<td><em>Oracle Big Data Appliance: Deep Dive and Roadmap for Customers and Partners</em></td>
<td>South 104</td>
</tr>
</tbody>
</table>

#### Tuesday

<table>
<thead>
<tr>
<th>Time</th>
<th>Session Title</th>
<th>Location</th>
</tr>
</thead>
<tbody>
<tr>
<td>10:45pm</td>
<td><em>Data Warehousing and Big Data Customer Panel</em></td>
<td>South 302</td>
</tr>
<tr>
<td>12:00pm</td>
<td><em>Top Five Things to Know About Oracle Database In-Memory</em></td>
<td>South 104</td>
</tr>
<tr>
<td>6:00pm</td>
<td><em>Meet the Experts - Oracle’s Big Data Management System</em></td>
<td>South 303</td>
</tr>
</tbody>
</table>
# DW and Big Data Sessions @OOW2014

## Wednesday

<table>
<thead>
<tr>
<th>Time</th>
<th>Session Title</th>
<th>Location</th>
</tr>
</thead>
<tbody>
<tr>
<td>10:15pm</td>
<td>Big Data &amp; Predictive Analytics: Fiserv Data Mining Case Study</td>
<td>South 301</td>
</tr>
<tr>
<td>12:45pm</td>
<td>If You Think Partitioning Is Only for Performance, Think Again</td>
<td>South 103</td>
</tr>
<tr>
<td>3:45pm</td>
<td>Oracle Big Data SQL: Deep Dive (SQL over Relational, NoSQL, and Hadoop)</td>
<td>South 103</td>
</tr>
<tr>
<td>4:45pm</td>
<td>Parallel Execution and Resource Management in Concurrent Environments</td>
<td>North 131</td>
</tr>
</tbody>
</table>

## Thursday

<table>
<thead>
<tr>
<th>Time</th>
<th>Session Title</th>
<th>Location</th>
</tr>
</thead>
<tbody>
<tr>
<td>12:00pm</td>
<td>Oracle Database In-Memory Meets Oracle Optimizer</td>
<td>South 104</td>
</tr>
</tbody>
</table>
# DW and Big Data Hands-on Lab @OOW2014

Oracle Big Data SQL: Unified SQL Analysis Across the Big Data Platform

<table>
<thead>
<tr>
<th>Date</th>
<th>Time</th>
<th>Location</th>
</tr>
</thead>
<tbody>
<tr>
<td>Monday</td>
<td>11:45am - 12:45pm</td>
<td>Hotel Nikko - Peninsula</td>
</tr>
<tr>
<td>Tuesday</td>
<td>3:45pm – 4:45pm</td>
<td>Hotel Nikko - Peninsula</td>
</tr>
<tr>
<td>Wednesday</td>
<td>1:15pm – 2:15pm</td>
<td>Hotel Nikko - Peninsula</td>
</tr>
<tr>
<td>Thursday</td>
<td>11:30am – 12:30pm</td>
<td>Hotel Nikko - Peninsula</td>
</tr>
</tbody>
</table>
THANK YOU FOR JOINING US TODAY

ENJOY YOUR OPENWORLD
Hardware and Software
Engineered to Work Together