Enabling Self-Service Analytics with Analytic Views & Data Visualization from Cloud to Desktop

Francesco Tisiot

BI Tech Lead at Rittman Mead

- Verona, Italy
- Rittman Mead Blog
- 10 Years Experience in BI/Analytics
- francesco.tisiot@rittmanmead.com
- @FTisiot
- Oracle ACE

About Rittman Mead

Rittman Mead is a **data and analytics company** who specialise in data visualisation, predictive analytics, enterprise reporting and data engineering.

We use our skill, experience and know-how to work with organisations across the world to interpret their data. We enable the business, the consumers, the data providers and IT to work towards a common goal, **delivering innovative and cost-effective solutions** based on our core values of thought leadership, hard work and honesty.

We work across **multiple verticals** on projects that range from mature, large scale implementations to proofs of concept and can provide skills in **development, architecture, delivery, training and support**.

You are DBAs right?

Photo by Marcelo Vaz on Unsplash

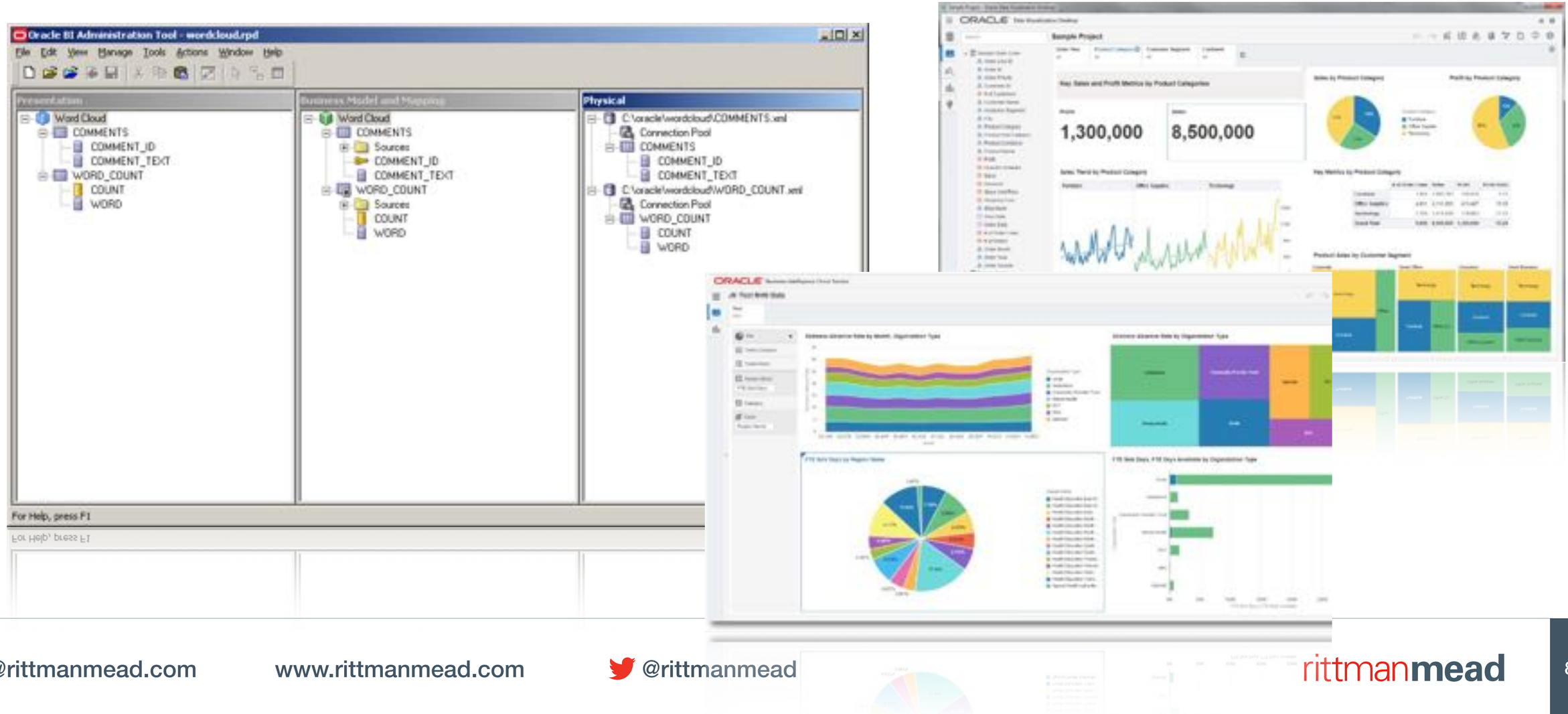
Inexperienced Users Writing SQL

Corporate IT-Driven BI Tools

Photo by Yuriy Kovalev on Unsplash

Business Intelligence Tools

Oracle Analytics



The Era of Peace is Over...

Self-Service Analytics

Photo by Connor Jalbert on Unsplash

Back in Feb 2016…

ORACLE?????

Bimodal IT = Marathon Runners + Sprinters. Deeply Different, Both Essential

Think Marathon Runner

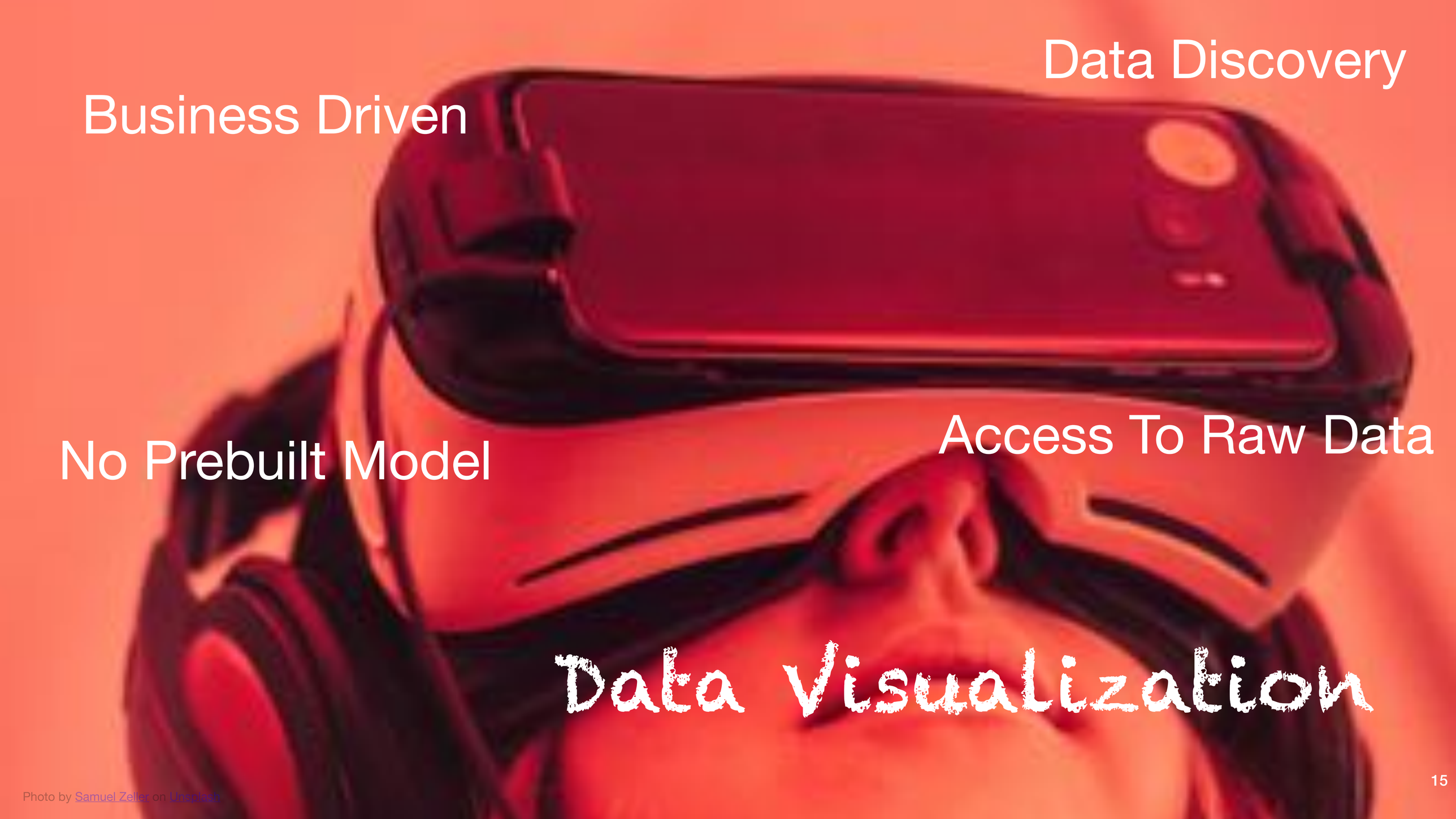
Mode 1		Mode 2
Reliability	**Goal**	Agility
Price for performance	**Value**	Revenue, brand, customer experience
Waterfall, V-Model, high-ceremony IID	**Approach**	Agile, kanban, low-ceremony IID
Plan-driven, approval-based	**Governance**	Empirical, continuous, process-based
Enterprise suppliers, long-term deals	**Sourcing**	Small, new vendors, short-term deals
Good at conventional process, projects	**Talent**	Good at new and uncertain projects
IT-centric, removed from customer	**Culture**	Business-centric, close to customer
Long (months)	**Cycle Times**	Short (days, weeks)

Think Sprinter

Gartner

"The evolution and sophistication of the self-service data preparation and data discovery capabilities in the market have shifted the focus of buyers in the BI and analytics market -- toward **easy-to-use tools** that support a full range of analytic workflow capabilities and **do not require significant involvement from IT** to predefine data models up front as a prerequisite to analysis,"

Photo by Tiago Muraro on Unsplash

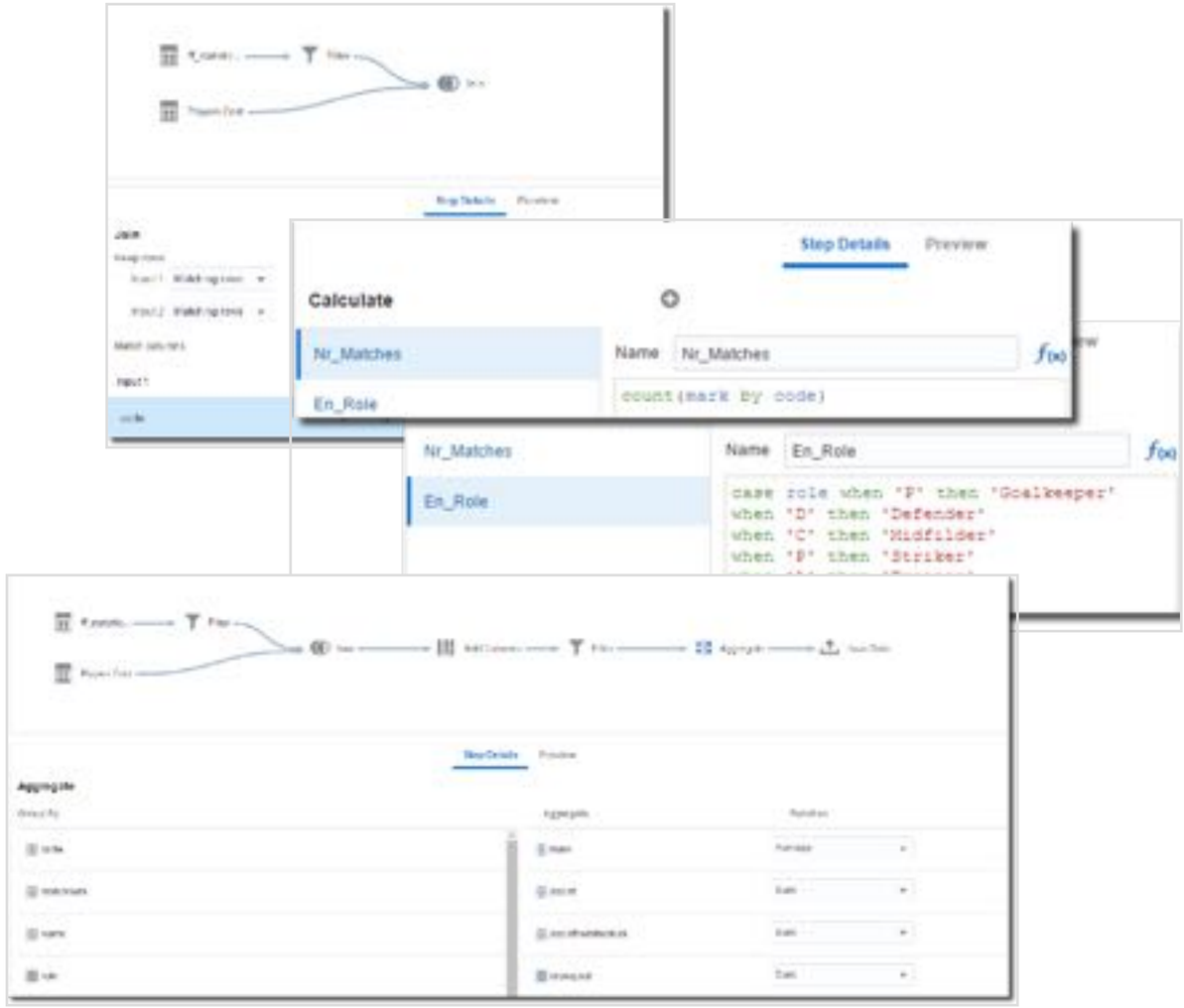


Data Visualization

- Information Exploration and Discovery
  - Single Panel Analytics
  - Data Mashup
  - Integrated with OBIEE
  - DataFlow Component

DataFlow Component

- Transform/Enrich Data
  - Filter
  - Aggregate
  - Join
  - Store Locally or Push Back
  - V4 Release
    - ML
    - Essbase Cube



In 2017


CHALLENGERS
LEADERS
Tableau
Microsoft
Qlik
Alteryx
SAP
SAS
IBM
TIBCO Software
Oracle
Birst
Salesforce
Board International
MicroStrategy
Sisense
ClearStory Data
Domo
ThoughtSpot
Information Builders
Yellowfin
Logi Analytics
Pyramid Analytics
Zoomdata
Pentaho
Datameer
NICHE PLAYERS
VISIONARIES
ABILITY TO EXECUTE
COMPLETENESS OF VISION
As of February 2017
Source: Gartner (February 2017)

Data Layer

Organisation Layer

Visualization Layer

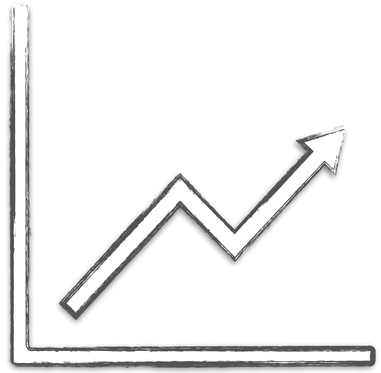
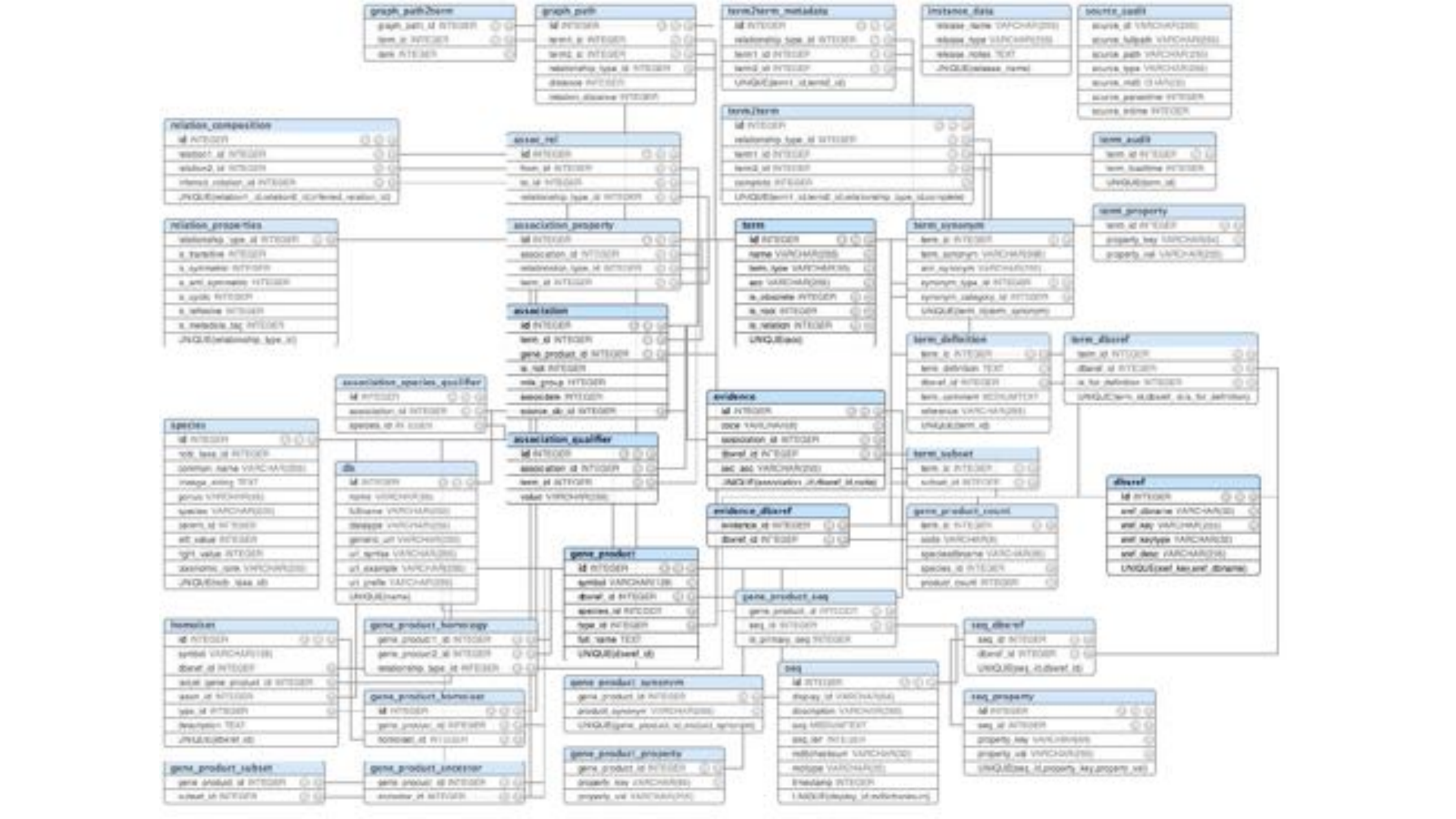
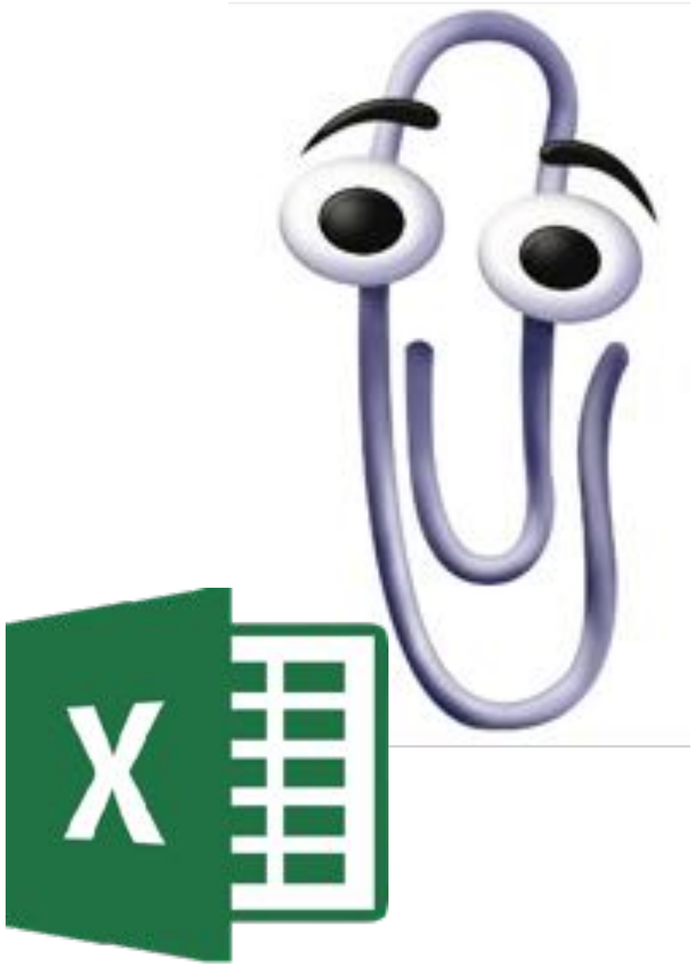


Photo by Laura Kranz on Unsplash


graph_path2term
graph_path_id INTEGER
term_id INTEGER
dark INTEGER
graph_path
id INTEGER
term1_id INTEGER
term2_id INTEGER
relationship_type_id INTEGER
distance INTEGER
relation_distance INTEGER
term2term_metadata
id INTEGER
relationship_type_id INTEGER
term1_id INTEGER
term2_id INTEGER
UNIQUE(term1_id,term2_id)
instance_data
release_name VARCHAR(255)
release_type VARCHAR(255)
release_notes TEXT
UNIQUE(release_name)
source_cadd
source_id VARCHAR(255)
source_fullpath VARCHAR(255)
source_path VARCHAR(255)
source_type VARCHAR(255)
source_md5 CHAR(32)
source_parseline INTEGER
source_edate INTEGER
relation_composition
id INTEGER
relation1_id INTEGER
relation2_id INTEGER
inferred_relation_id INTEGER
UNIQUE(relation1_id,relation2_id,inferred_relation_id)
assoc_rel
id INTEGER
from_id INTEGER
to_id INTEGER
relationship_type_id INTEGER
term2term
id INTEGER
relationship_type_id INTEGER
term1_id INTEGER
term2_id INTEGER
complete INTEGER
UNIQUE(term1_id,term2_id,relationship_type_id,complete)
term_audit
term_id INTEGER
term_lastdbx INTEGER
UNIQUE(term_id)
relation_properties
relationship_type_id INTEGER
is_transitive INTEGER
is_symmetric INTEGER
is_anti_symmetric INTEGER
is_cyclic INTEGER
is_reflexive INTEGER
is_metadata_tag INTEGER
UNIQUE(relationship_type_id)
association_property
id INTEGER
association_id INTEGER
relationship_type_id INTEGER
term_id INTEGER
term
id INTEGER
name VARCHAR(255)
term_type VARCHAR(55)
acc VARCHAR(255)
is_obsolete INTEGER
is_root INTEGER
is_relation INTEGER
UNIQUE(acc)
term_synonym
term_id INTEGER
term_synonym VARCHAR(255)
acc_synonym VARCHAR(255)
synonym_type_id INTEGER
synonym_category_id INTEGER
UNIQUE(term_id,term_synonym)
term_property
term_id INTEGER
property_key VARCHAR(64)
property_val VARCHAR(255)
association
id INTEGER
term_id INTEGER
gene_product_id INTEGER
is_not INTEGER
role_group INTEGER
assocdate INTEGER
source_db_id INTEGER
association_species_qualifier
id INTEGER
association_id INTEGER
species_id INTEGER
association_qualifier
id INTEGER
association_id INTEGER
term_id INTEGER
value VARCHAR(255)
evidence
id INTEGER
code VARCHAR(20)
association_id INTEGER
dbxref_id INTEGER
seq_acc VARCHAR(255)
UNIQUE(association_id,dbxref_id,rank)
term_definition
term_id INTEGER
term_definition TEXT
dbxref_id INTEGER
term_comment BIGINT(21)
reference VARCHAR(255)
UNIQUE(term_id)
term_dbxref
term_id INTEGER
dbxref_id INTEGER
is_for_definition INTEGER
UNIQUE(term_id,dbxref_id,is_for_definition)
species
id INTEGER
ncbi_taxa_id INTEGER
common_name VARCHAR(255)
mnemonic TEXT
genus VARCHAR(255)
species VARCHAR(255)
taxon_id INTEGER
left_value INTEGER
right_value INTEGER
taxonomic_rank VARCHAR(255)
UNIQUE(ncbi_taxa_id)
db
id INTEGER
name VARCHAR(55)
fullname VARCHAR(255)
datatype VARCHAR(255)
generic_url VARCHAR(255)
url_syntax VARCHAR(255)
url_example VARCHAR(255)
uri_prefix VARCHAR(255)
UNIQUE(name)
evidence_dbxref
evidence_id INTEGER
dbxref_id INTEGER
term_subset
term_id INTEGER
subset_id INTEGER
gene_product_count
term_id INTEGER
code VARCHAR(20)
speciesdbname VARCHAR(55)
species_id INTEGER
product_count INTEGER
dbxref
id INTEGER
xref_dbname VARCHAR(32)
xref_key VARCHAR(255)
xref_keytype VARCHAR(32)
xref_desc VARCHAR(255)
UNIQUE(xref_key,xref_dbname)
gene_product
id INTEGER
symbol VARCHAR(128)
dbxref_id INTEGER
species_id INTEGER
type_id INTEGER
full_name TEXT
UNIQUE(dbxref_id)
gene_product_seq
gene_product_id INTEGER
seq_id INTEGER
is_primary_seq INTEGER
seq_dbxref
seq_id INTEGER
dbxref_id INTEGER
UNIQUE(seq_id,dbxref_id)
homolset
id INTEGER
symbol VARCHAR(128)
dbxref_id INTEGER
assoc_gene_product_id INTEGER
taxon_id INTEGER
type_id INTEGER
description TEXT
UNIQUE(dbxref_id)
gene_product_homology
gene_product1_id INTEGER
gene_product2_id INTEGER
relationship_type_id INTEGER
gene_product_homolset
id INTEGER
gene_product_id INTEGER
homolset_id INTEGER
gene_product_synonym
gene_product_id INTEGER
product_synonym VARCHAR(255)
UNIQUE(gene_product_id,product_synonym)
gene_product_property
gene_product_id INTEGER
property_key LONGVARBIN
property_val VARCHAR(64)
seq
id INTEGER
display_id VARCHAR(64)
description VARCHAR(255)
seq MEDIUMTEXT
seq_len INTEGER
md5checksum VARCHAR(32)
moltype VARCHAR(16)
timestamp INTEGER
UNIQUE(display_id,md5checksum)
seq_property
id INTEGER
seq_id INTEGER
property_key VARCHAR(64)
property_val VARCHAR(255)
UNIQUE(seq_id,property_key,property_val)
gene_product_subset
gene_product_id INTEGER
subset_id INTEGER
gene_product_ancestor
gene_product_id INTEGER
ancestor_id INTEGER



I'm Back Nerds!

Analytic Views

New in 12.2

DB Metadata Objects

Joins

Hierarchies

Aggregations

Analytic Views

Photo by Domenico Loia on Unsplash

Components

Datasource
Attributes
Levels

Attribute Dimension

Levels
Relationship

Hierarchies

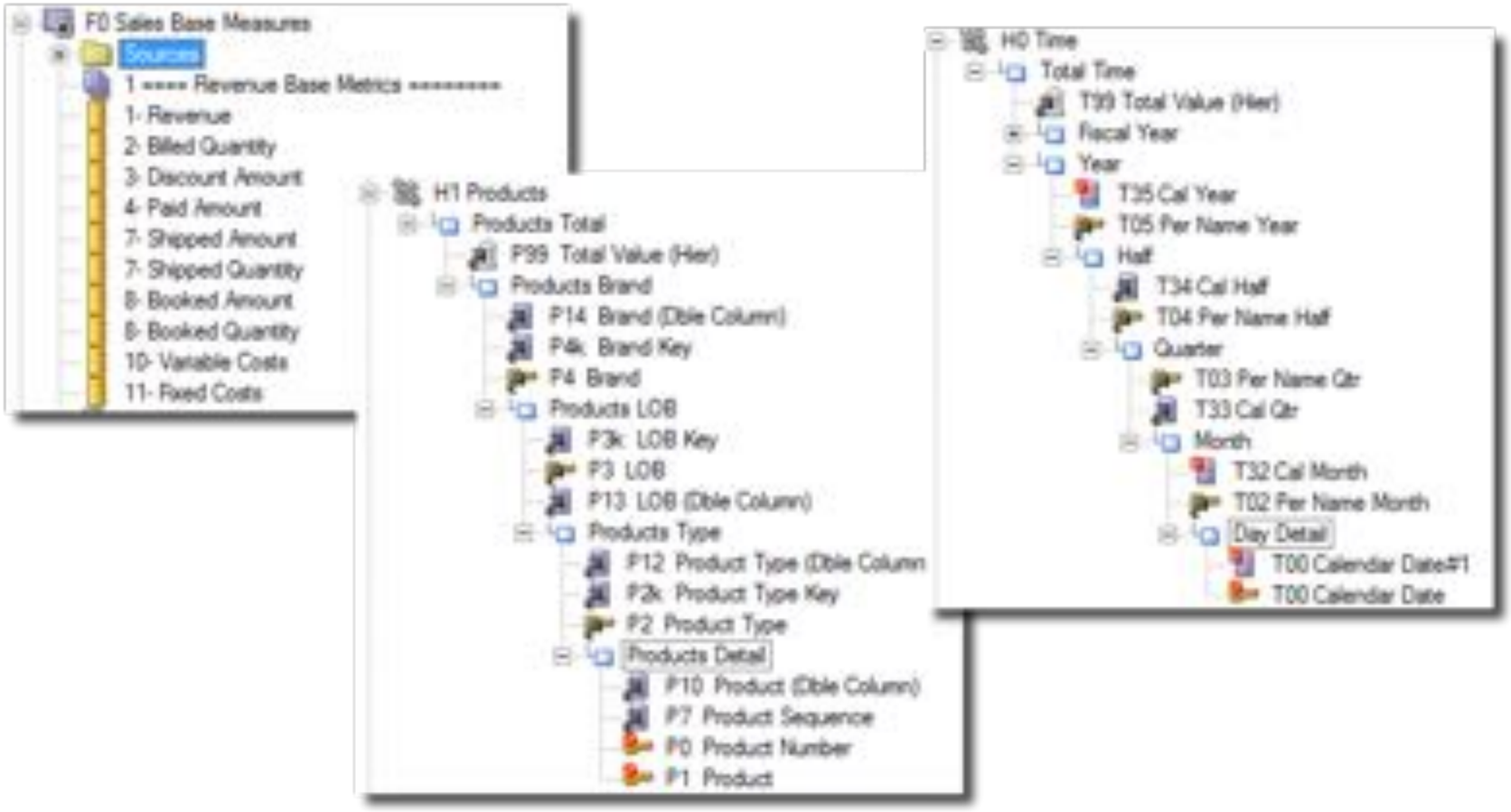
Dimensions
Measures
Aggregations
Windowing

Analytic View

Caching
Materialised
Views

Cache Group

The Model


F0 Sales Base Measures
Sources
1 ==== Revenue Base Metrics ========
1- Revenue
2- Billed Quantity
3- Discount Amount
4- Paid Amount
7- Shipped Amount
7- Shipped Quantity
8- Booked Amount
8- Booked Quantity
10- Variable Costs
11- Fixed Costs
H1 Products
Products Total
P99 Total Value (Hier)
Products Brand
P14 Brand (Dble Column)
P4k Brand Key
P4 Brand
Products LOB
P3k LOB Key
P3 LOB
P13 LOB (Dble Column)
Products Type
P12 Product Type (Dble Column
P2k Product Type Key
P2 Product Type
Products Detail
P10 Product (Dble Column)
P7 Product Sequence
P0 Product Number
P1 Product
H0 Time
Total Time
T99 Total Value (Hier)
Fiscal Year
Year
T35 Cal Year
T05 Per Name Year
Half
T34 Cal Half
T04 Per Name Half
Quarter
T03 Per Name Qtr
T33 Cal Qtr
Month
T32 Cal Month
T02 Per Name Month
Day Detail
T00 Calendar Date#1
T00 Calendar Date

Attribute Dimension

ATTRIBUTE DIMENSION

USING

```
CREATE OR REPLACE ATTRIBUTE DIMENSION D1_DIM_PRODUCT
USING SAMP_PRODUCTS_D
ATTRIBUTES
 (PROD_KEY as P0_Product_Number
    CLASSIFICATION caption VALUE 'P0 Product Number',
  PROD_DSC as P1_Product
    CLASSIFICATION caption VALUE 'P1 Product',
  TYPE as P2_Product_Type
    CLASSIFICATION caption VALUE 'P2 Product Type',
  TYPE_KEY as P2k_Product_Type
    CLASSIFICATION caption VALUE 'P2k Product Type',
  LOB as P3_LOB
    CLASSIFICATION caption VALUE 'P3 LOB',
  …
  SEQUENCE as P7_Product_Sequence
    CLASSIFICATION caption VALUE 'P7 Product Sequence',
  TOTAL_VALUE as P99_Total_Value
    CLASSIFICATION caption VALUE 'P99 Total Value')
```

Logical Column	Expression		Physical Table
P0 Product Number	Prod_Key		D10 Product (Dynam
P1 Product	Prod_Dsc		D10 Product (Dynam
P10 Product (Dble Column)	Prod_Dsc		D10 Product (Dynam
P12 Product Type (Dble Co	Type		D10 Product (Dynam
P13 LOB (Dble Column)	LOB		D10 Product (Dynam
P14 Brand (Dble Column)	Brand		D10 Product (Dynam
P2 Product Type	Type		D10 Product (Dynam
P2k Product Type Key	Type_Key		D10 Product (Dynam
P3 LOB	LOB		D10 Product (Dynam
P3k LOB Key	LOB_Key		D10 Product (Dynam
P4 Brand	Brand		D10 Product (Dynam
P4k Brand Key	Brand_Key		D10 Product (Dynam
P5 Attribute 1	Attribute_1		D10 Product (Dynam
P6 Attribute 2	Attribute_2		D10 Product (Dynam
P7 Product Sequence	Sequence		D10 Product (Dynam
P8 Product Image			
P9 Product Image Id			
P99 Total Value (Hier)	TOTAL_VALUE		D10 Product (Dynam

Attribute Dimension

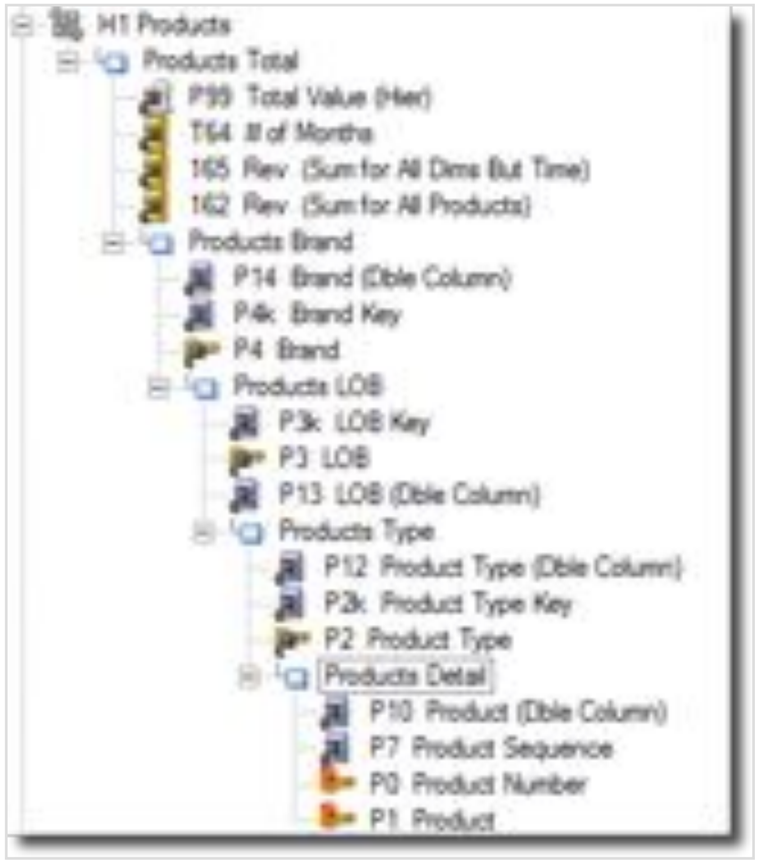
```
LEVEL BRAND
  CLASSIFICATION caption VALUE 'BRAND'
  CLASSIFICATION description VALUE 'Brand'
  KEY P4k_Brand
  MEMBER NAME P4_Brand
  MEMBER CAPTION P4_Brand
  ORDER BY P4_Brand
LEVEL Product_LOB
  CLASSIFICATION caption VALUE 'LOB'
  CLASSIFICATION description VALUE 'Lob'
  KEY P3k_LOB
  MEMBER NAME P3_LOB
  MEMBER CAPTION P3_LOB
  ORDER BY P3_LOB
  DETERMINES(P4k_Brand)
LEVEL Product_Type
  CLASSIFICATION caption VALUE 'Type'
  CLASSIFICATION description VALUE 'Type'
  KEY P2k_Product_Type
  MEMBER NAME P2_Product_Type
  MEMBER CAPTION P2_Product_Type
  ORDER BY P2_Product_Type
  DETERMINES(P3k_LOB,P4k_Brand)
LEVEL Product_Details
  CLASSIFICATION caption VALUE 'Detail'
  CLASSIFICATION description VALUE 'Detail'
  KEY P0_Product_Number
  MEMBER NAME P1_Product
  MEMBER CAPTION P1_Product
  ORDER BY P1_Product
  DETERMINES(P2k_Product_Type,P3k_LOB,P4k_Brand)
ALL MEMBER NAME 'ALL PRODUCTS';
```

LEVEL

KEY

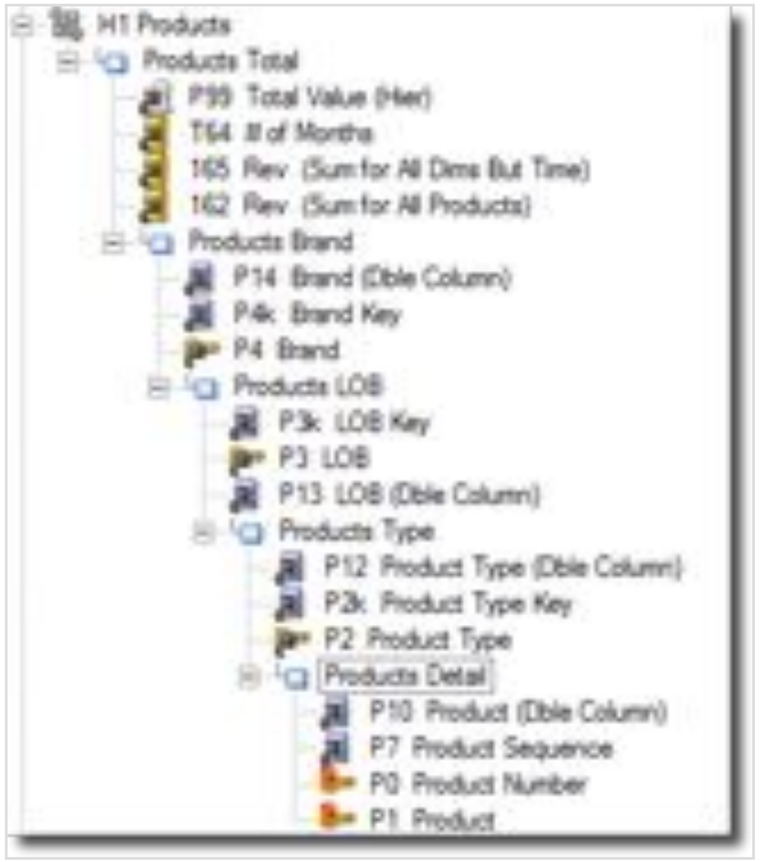
CAPTION

DETERMINES



Dimension Hierarchy

```
CREATE OR REPLACE HIERARCHY PRODUCT_HIER
  CLASSIFICATION caption VALUE 'Products Hierarchy'
USING D1_DIM_PRODUCT
  (Product_Details CHILD OF
   Product_Type CHILD OF
   Product_LOB CHILD OF
   BRAND);
```



Dimension Hierarchy

P0_PRODUCT...	P1_PRODUCT	P2_PRODUCT_...	P2K_PRO...	P3_LOB	P3K_LOB	P4_BRAND	P4K_BRAND	MEMBER_NAME	MEMBER_UNIQUE_NAME
(null)	(null)	(null)	(null)	(null)	(null)	(null)	(null)	ALL PRODUCTS	[ALL].[ALL PRODUCTS]
(null)	(null)	(null)	(null)	(null)	(null)	BizTech	10001	BizTech	[BRAND].&[10001]
(null)	(null)	(null)	(null)	Communication	1001	BizTech	10001	Communication	[PRODUCT_LOB].&[1001]
(null)	(null)	Cell Phones	101	Communication	1001	BizTech	10001	Cell Phones	[PRODUCT_TYPE].&[101]
17	CompCell RX3	Cell Phones	101	Communication	1001	BizTech	10001	CompCell RX3	[PRODUCT_DETAILS].&[17]
8	V5x Flip Phone	Cell Phones	101	Communication	1001	BizTech	10001	V5x Flip Phone	[PRODUCT_DETAILS].&[8]
(null)	(null)	Smart Phones	102	Communication	1001	BizTech	10001	Smart Phones	[PRODUCT_TYPE].&[102]
10	KeyMax S-Phone	Smart Phones	102	Communication	1001	BizTech	10001	KeyMax S-Phone	[PRODUCT_DETAILS].&[10]

- For Each Level
  - Key
  - Name
  - Ordering
  - Level
- Member Unique Name

Time Dimension

```
CREATE OR REPLACE ATTRIBUTE DIMENSION D0_DIM_DATE
DIMENSION TYPE TIME
USING SAMP_TIME_DAY_D
ATTRIBUTES
 (CALENDAR_DATE AS T00_CALENDAR_DATE,
  PER_NAME_MONTH AS T02_PER_NAME_MONTH,
  PER_NAME_QTR AS T03_PER_NAME_QTR,
  PER_NAME_YEAR AS T04_PER_NAME_YEAR,
  DAY_KEY AS T06_ROW_WID,
  BEG_OF_MTH_WID AS T22_BEG_OF_MTH_WID,
  BEG_OF_QTR_WID AS T23_BEG_OF_QTR_WID
  )
    LEVEL CAL_DAY
      LEVEL TYPE DAYS
      KEY T00_CALENDAR_DATE
      ORDER BY T00_CALENDAR_DATE
      DETERMINES(T22_BEG_OF_MTH_WID, T23_BEG_OF_QTR_WID,T04_PER_NAME_YEAR)
    LEVEL CAL_MONTH
      LEVEL TYPE MONTHS
      KEY T22_BEG_OF_MTH_WID
      MEMBER NAME T02_PER_NAME_MONTH
      ORDER BY T22_BEG_OF_MTH_WID
      DETERMINES(T23_BEG_OF_QTR_WID,T04_PER_NAME_YEAR)
    LEVEL CAL_QUARTER
      LEVEL TYPE QUARTERS
      KEY T23_BEG_OF_QTR_WID
      MEMBER NAME T03_PER_NAME_QTR
      ORDER BY T23_BEG_OF_QTR_WID
      DETERMINES(T04_PER_NAME_YEAR)
    LEVEL CAL_YEAR
      LEVEL TYPE YEARS
      KEY T04_PER_NAME_YEAR
      MEMBER NAME T04_PER_NAME_YEAR
      ORDER BY T04_PER_NAME_YEAR
    ALL MEMBER NAME 'ALL TIMES';
```

DIMENSION TYPE TIME

LEVEL TYPE

Logical Column	Expression		Physical Table
T00 Calendar Date	Calendar_Date		D01 Time Day Grain
T01 Per Name Week	Per_Name_Week		D01 Time Day Grain
T02 Per Name Month	Per_Name_Month		D01 Time Day Grain
T03 Per Name Qtr	Per_Name_Qtr		D01 Time Day Grain
T04 Per Name Half	Per_Name_Half		D01 Time Day Grain
T05 Per Name Year	Per_Name_Year		D01 Time Day Grain
T06 Row Wid	Day_Key		D01 Time Day Grain
T11 Day Of Week	Day_Of_Week		D01 Time Day Grain
T12 Day Of Month	Day_Of_Month		D01 Time Day Grain
T15 Day Of Year	Day_Of_Year		D01 Time Day Grain
T21 Beg of Week Wid	Beg_of_Week_Wid		D01 Time Day Grain
T22 Beg of Mth Wid	Beg_of_Mth_Wid		D01 Time Day Grain
T23 Beg of Qtr Wid	Beg_of_Qtr_Wid		D01 Time Day Grain

Time Dimension

```
CREATE OR REPLACE HIERARCHY TIME_HIER
USING D0_DIM_DATE
  (CAL_DAY CHILD OF
   CAL_MONTH CHILD OF
   CAL_QUARTER CHILD OF
   CAL_YEAR);
```

Logical Column	Expression		Physical Table
T00 Calendar Date	Calendar_Date		D01 Time Day Grain
T01 Per Name Week	Per_Name_Week		D01 Time Day Grain
T02 Per Name Month	Per_Name_Month		D01 Time Day Grain
T03 Per Name Qtr	Per_Name_Qtr		D01 Time Day Grain
T04 Per Name Half	Per_Name_Half		D01 Time Day Grain
T05 Per Name Year	Per_Name_Year		D01 Time Day Grain
T06 Row Wid	Day_Key		D01 Time Day Grain
T11 Day Of Week	Day_Of_Week		D01 Time Day Grain
T12 Day Of Month	Day_Of_Month		D01 Time Day Grain
T15 Day Of Year	Day_Of_Year		D01 Time Day Grain
T21 Beg of Week Wid	Beg_of_Week_Wid		D01 Time Day Grain
T22 Beg of Mth Wid	Beg_of_Mth_Wid		D01 Time Day Grain
T23 Beg of Qtr Wid	Beg_of_Qtr_Wid		D01 Time Day Grain

Analytic View Definition

```
CREATE OR REPLACE ANALYTIC VIEW F0_SALES_BASE_MEASURES
USING SAMP_REVENUE_F
DIMENSION BY
  (D0_DIM_DATE
    KEY BILL_DAY_DT REFERENCES TOO_CALENDAR_DATE
    HIERARCHIES (
      TIME_HIER DEFAULT),
   D1_DIM_PRODUCT
    KEY PROD_KEY REFERENCES P0_Product_Number
    HIERARCHIES (
      PRODUCT_HIER DEFAULT)
   )
MEASURES
 (F1_REVENUE FACT REVENUE AGGREGATE BY SUM,
  F10_VARIABLE_COST FACT COST_VARIABLE AGGREGATE BY SUM,
  F11_FIXED_COST FACT COST_FIXED AGGREGATE BY SUM,
  F2_BILLED_QTY FACT UNITS,
  F3_DISCOUNT_AMOUNT FACT DISCNT_VALUE AGGREGATE BY SUM,
  F4_AVG_REVENUE FACT REVENUE AGGREGATE BY AVG,
  F21_REVENUE_AGO AS (LAG(F1_REVENUE) OVER (HIERARCHY TIME_HIER OFFSET 1))
  )
DEFAULT MEASURE F1_REVENUE;
```

USING

LIST OF DIMENSIONS AND HIERARCHIES

Logical Column	Expression		Physical Table
1- Revenue	Revenue		F10 Billed Rev
10- Variable Costs	Cost_Variable		F10 Billed Rev
11- Fixed Costs	Cost_Fixed		F10 Billed Rev
2- Billed Quantity	Units		F10 Billed Rev
3- Discount Amount	Discnt_Value		F10 Billed Rev

AGGREGATED BY

COMPLEX FORMULAS

Complex Formulas

- AGO

```
LAG(SALES) OVER (HIERARCHY time_hier OFFSET 1)
```

- Difference from AGO

```
LAG_DIFF(SALES) OVER (HIERARCHY time_hier OFFSET 1)
```

- AGO Fixed Level

```
LAG(SALES) OVER (HIERARCHY time_hier OFFSET 1
ACROSS ANCESTOR AT LEVEL quarter)
```

- Percentage

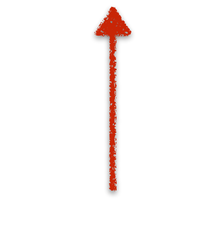
```
SHARE_OF(sales HIERARCHY product_hier LEVEL department)
```

- Fix Points

```
QUALIFY (sales, time_hier = year['11'])
```

Formatting

```
sales FACT sales
    CLASSIFICATION caption VALUE 'Sales'
    CLASSIFICATION description VALUE 'Sales'
    CLASSIFICATION format_string VALUE '$9,999.99',
```



Cache Group

```
CACHE
      -- The list of measures in the MV.
      MEASURE GROUP (
          amount_sold,
          quantity_sold)
        -- Levels that match the GROUP BY clause of the materialized view.
        LEVELS (
          sh_times_calendar_hier.calendar_year,
          sh_products_hier.category,
          sh_customers_hier.country,
          sh_channels_hier.channel_class,
          sh_promotions_hier.category)
        MATERIALIZED
```

MATERIALIZED VIEW

Using Analytics Views

Using Analytic Views

Original SQL

```
SELECT D.CAL_MONTH,
  D.BEG_OF_MTH_WID,
  P.BRAND,
  SUM(F.REVENUE) AS F01_REVENUE,
  SUM(F.UNITS)   AS F02_BILLED_QTY
FROM SAMP_REVENUE_F F
JOIN SAMP_PRODUCTS_D P
ON (F.PROD_KEY = P.PROD_KEY)
JOIN SAMP_TIME_DAY_D D
ON (F.BILL_DAY_DT = D.CALENDAR_DATE)
GROUP BY D.CAL_MONTH,
  D.BEG_OF_MTH_WID,
  P.BRAND
ORDER BY D.BEG_OF_MTH_WID,
  P.BRAND;
```

VS

AV SQL

```
SELECT TIME_HIER.MEMBER_NAME AS TIME_SLICE,
  PRODUCT_HIER.MEMBER_NAME   AS PRODUCT_SLICE,
  TIME_HIER.LEVEL_NAME AS TIME_LEVEL,
  PRODUCT_HIER.LEVEL_NAME AS PRODUCT_LEVEL,
  TIME_HIER.HIER_ORDER AS TIME_HIER_ORDER,
  PRODUCT_HIER.HIER_ORDER AS PRODUCT_HIER_ORDER,
  F1_REVENUE,
  F2_BILLED_QTY
FROM F0_SALES_BASE_MEASURES
ORDER BY TIME_HIER.HIER_ORDER,
  PRODUCT_HIER.HIER_ORDER;
```

Using Analytic Views

Original SQL

```
SELECT D.CAL_MONTH,
  D.BEG_OF_MTH_WID,
  P.BRAND,
  SUM(F.REVENUE) AS F01_REVENUE,
  SUM(F.UNITS)   AS F02_BILLED_QTY
FROM SAMP_REVENUE_F F
JOIN SAMP_PRODUCTS_D P
ON (F.PROD_KEY = P.PROD_KEY)
JOIN SAMP_TIME_DAY_D D
ON (F.BILL_DAY_DT = D.CALENDAR_DATE)
GROUP BY D.CAL_MONTH,
  D.BEG_OF_MTH_WID,
  P.BRAND
ORDER BY D.BEG_OF_MTH_WID,
  P.BRAND;
```

- Fields
- Join
- Aggregation
- Complex Formulas
- Group By
- Order By

Using Analytic Views

AV SQL

```
SELECT TIME_HIER.MEMBER_NAME AS TIME_SLICE,
  PRODUCT_HIER.MEMBER_NAME   AS PRODUCT_SLICE,
  TIME_HIER.LEVEL_NAME AS TIME_LEVEL,
  PRODUCT_HIER.LEVEL_NAME AS PRODUCT_LEVEL,
  TIME_HIER.HIER_ORDER AS TIME_HIER_ORDER,
  PRODUCT_HIER.HIER_ORDER AS PRODUCT_HIER_ORDER,
  F1_REVENUE,
  F2_BILLED_QTY
FROM F0_SALES_BASE_MEASURES
WHERE TIME_HIER.LEVEL_NAME  IN ('CAL_MONTH')
AND PRODUCT_HIER.LEVEL_NAME IN ('BRAND')
ORDER BY TIME_HIER.HIER_ORDER,
  PRODUCT_HIER.HIER_ORDER;
```

- Standard SQL
- ~~Joins~~
- ~~Complex Formulas~~
- ~~Aggregation~~
- ~~Order By~~

Standard SQL

```
SELECT TIME_HIER.MEMBER_NAME AS TIME_SLICE,
  PRODUCT_HIER.MEMBER_NAME   AS PRODUCT_SLICE,
  TIME_HIER.LEVEL_NAME AS TIME_LEVEL,
  PRODUCT_HIER.LEVEL_NAME AS PRODUCT_LEVEL,
  TIME_HIER.HIER_ORDER AS TIME_HIER_ORDER,
  PRODUCT_HIER.HIER_ORDER AS PRODUCT_HIER_ORDER,
  F1_REVENUE,
  F2_BILLED_QTY
FROM F0_SALES_BASE_MEASURES
ORDER BY TIME_HIER.HIER_ORDER,
  PRODUCT_HIER.HIER_ORDER;
```

Standard SQL

```
SELECT TIME_HIER.MEMBER_NAME AS TIME_SLICE,
  PRODUCT_HIER.MEMBER_NAME   AS PRODUCT_SLICE,
  TIME_HIER.LEVEL_NAME AS TIME_LEVEL,
  PRODUCT_HIER.LEVEL_NAME AS PRODUCT_LEVEL,
  TIME_HIER.HIER_ORDER AS TIME_HIER_ORDER,
  PRODUCT_HIER.HIER_ORDER AS PRODUCT_HIER_ORDER,
  F1_REVENUE,
  F2_BILLED_QTY
FROM F0_SALES_BASE_MEASURES
ORDER BY TIME_HIER.HIER_ORDER,
  PRODUCT_HIER.HIER_ORDER;
```

Reference to Analytic View

Optional reference to hierarchies

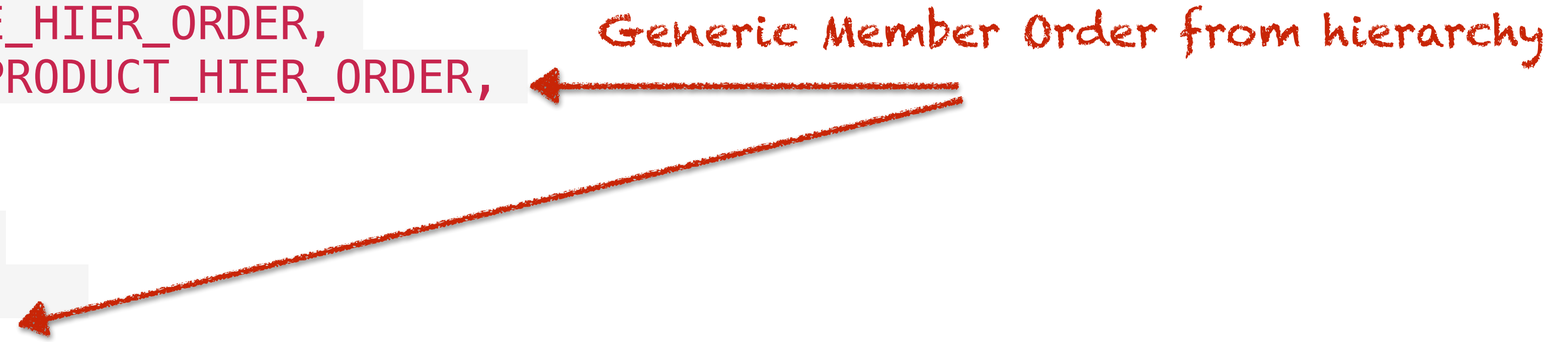
Standard SQL

Generic Member Name from hierarchy

```
SELECT TIME_HIER.MEMBER_NAME AS TIME_SLICE,
  PRODUCT_HIER.MEMBER_NAME   AS PRODUCT_SLICE,
  TIME_HIER.LEVEL_NAME AS TIME_LEVEL,
  PRODUCT_HIER.LEVEL_NAME AS PRODUCT_LEVEL,
  TIME_HIER.HIER_ORDER AS TIME_HIER_ORDER,
  PRODUCT_HIER.HIER_ORDER AS PRODUCT_HIER_ORDER,
  F1_REVENUE,
  F2_BILLED_QTY
FROM F0_SALES_BASE_MEASURES
ORDER BY TIME_HIER.HIER_ORDER,
  PRODUCT_HIER.HIER_ORDER;
```

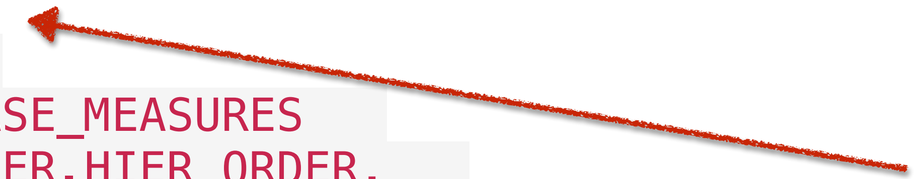
Standard SQL

```
SELECT TIME_HIER.MEMBER_NAME AS TIME_SLICE,
  PRODUCT_HIER.MEMBER_NAME   AS PRODUCT_SLICE,
  TIME_HIER.LEVEL_NAME AS TIME_LEVEL,
  PRODUCT_HIER.LEVEL_NAME AS PRODUCT_LEVEL,
  TIME_HIER.HIER_ORDER AS TIME_HIER_ORDER,
  PRODUCT_HIER.HIER_ORDER AS PRODUCT_HIER_ORDER,
  F1_REVENUE,
  F2_BILLED_QTY
FROM F0_SALES_BASE_MEASURES
ORDER BY TIME_HIER.HIER_ORDER,
  PRODUCT_HIER.HIER_ORDER;
```



Standard SQL

```
SELECT TIME_HIER.MEMBER_NAME AS TIME_SLICE,
  PRODUCT_HIER.MEMBER_NAME   AS PRODUCT_SLICE,
  TIME_HIER.LEVEL_NAME AS TIME_LEVEL,
  PRODUCT_HIER.LEVEL_NAME AS PRODUCT_LEVEL,
  TIME_HIER.HIER_ORDER AS TIME_HIER_ORDER,
  PRODUCT_HIER.HIER_ORDER AS PRODUCT_HIER_ORDER,
  F1_REVENUE,
  F2_BILLED_QTY
FROM F0_SALES_BASE_MEASURES
ORDER BY TIME_HIER.HIER_ORDER,
  PRODUCT_HIER.HIER_ORDER;
```



Reference to auto aggregated measures

Standard SQL Code

```
SELECT TIME_HIER.MEMBER_NAME AS TIME_SLICE,
  PRODUCT_HIER.MEMBER_NAME   AS PRODUCT_SLICE,
  F1_REVENUE,
  F2_BILLED_QTY
FROM F0_SALES_BASE_MEASURES
WHERE TIME_HIER.LEVEL_NAME  IN ('CAL_MONTH')
AND PRODUCT_HIER.LEVEL_NAME IN ('BRAND')
ORDER BY TIME_HIER.HIER_ORDER,
  PRODUCT_HIER.HIER_ORDER;
```

Level Definition

```
WHERE TIME_HIER.LEVEL_NAME  IN ('CAL_MONTH')
AND PRODUCT_HIER.LEVEL_NAME IN ('BRAND')
```

```
WHERE TIME_HIER.LEVEL_NAME  IN ('CAL_YEAR')
AND PRODUCT_HIER.LEVEL_NAME IN ('LOB')
```

Data Layer

Organisation Layer

Visualization Layer

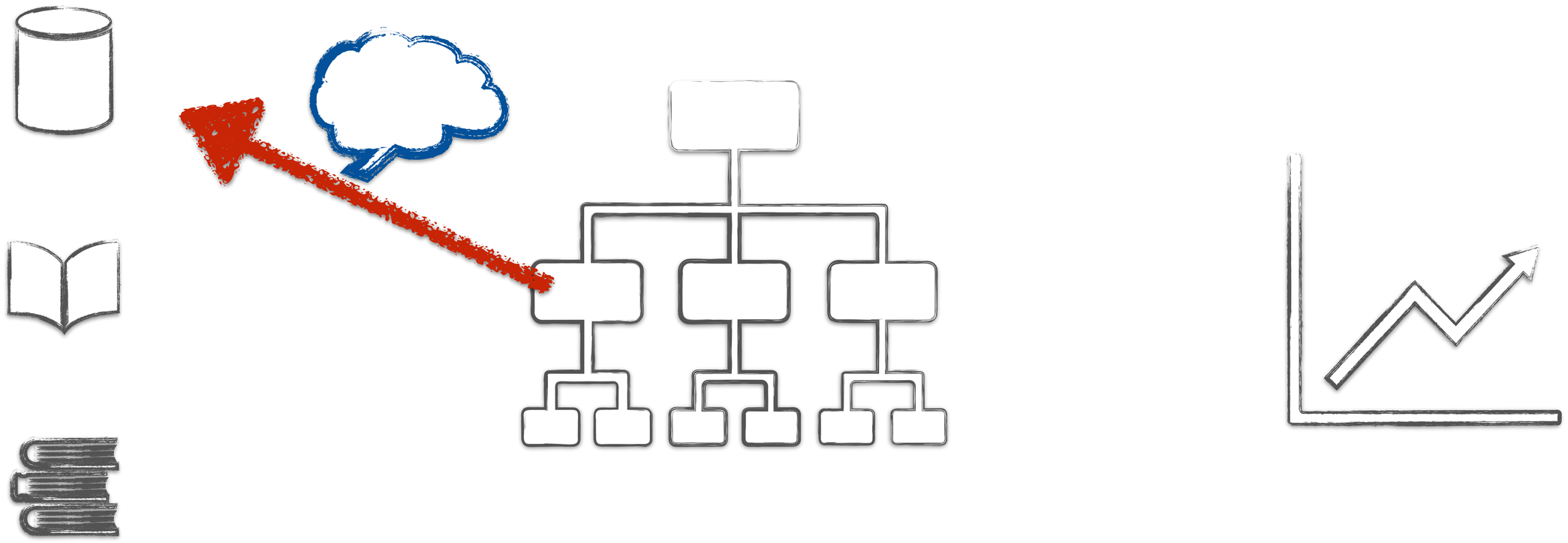


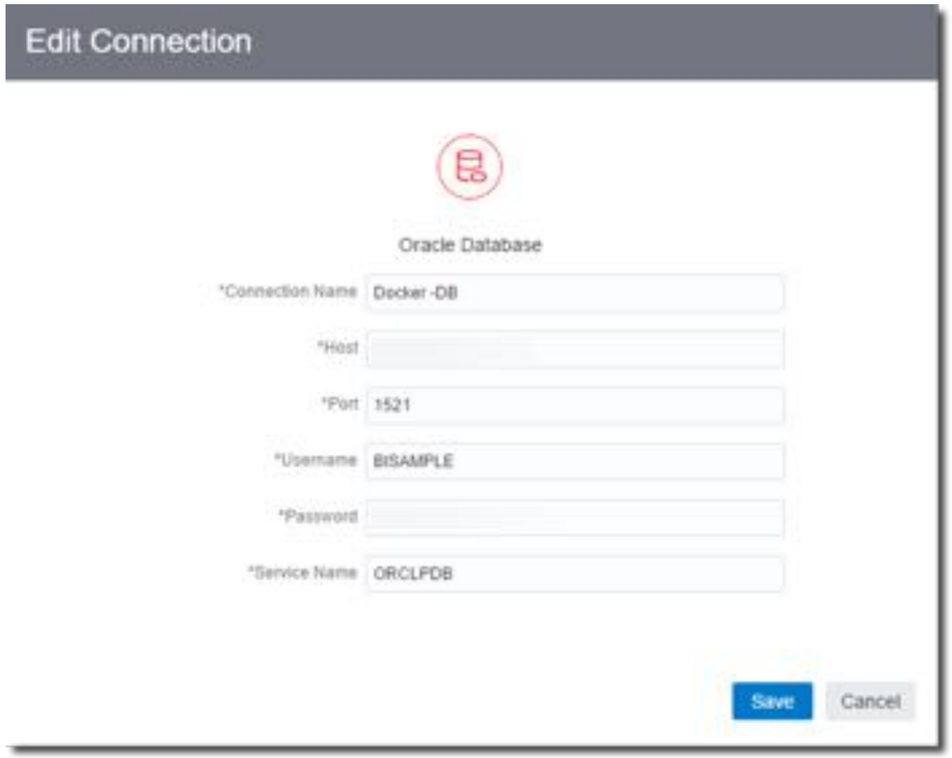
Photo by Jerry Kiesewetter on Unsplash


Analytic
Views
OBIEE

Using Analytics Views with Data Visualization

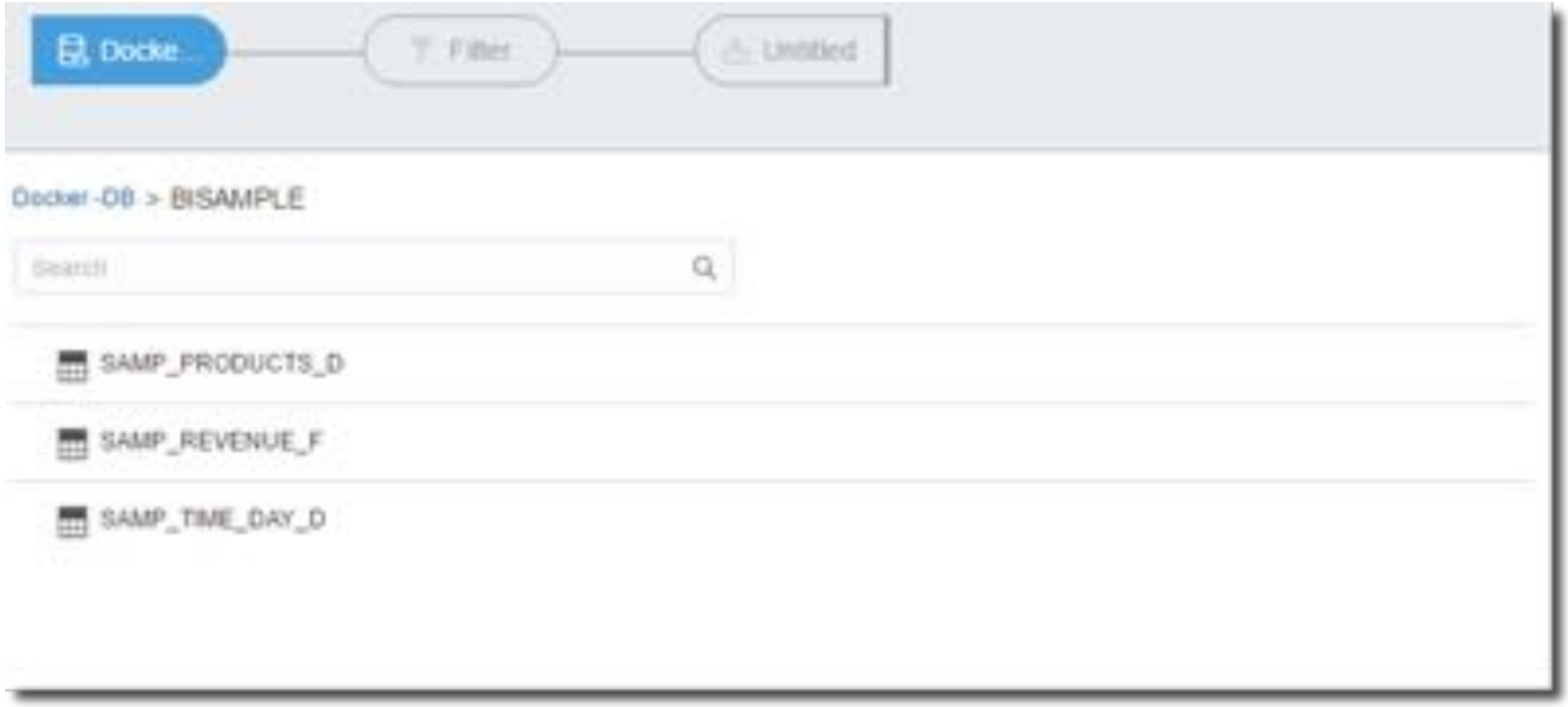
Analytic Views in DVD/DVCS/OAC

Create Database Connection



Analytic Views in DVD/DVCS/OAC

Analytic Views not Listed as Sources!!!!

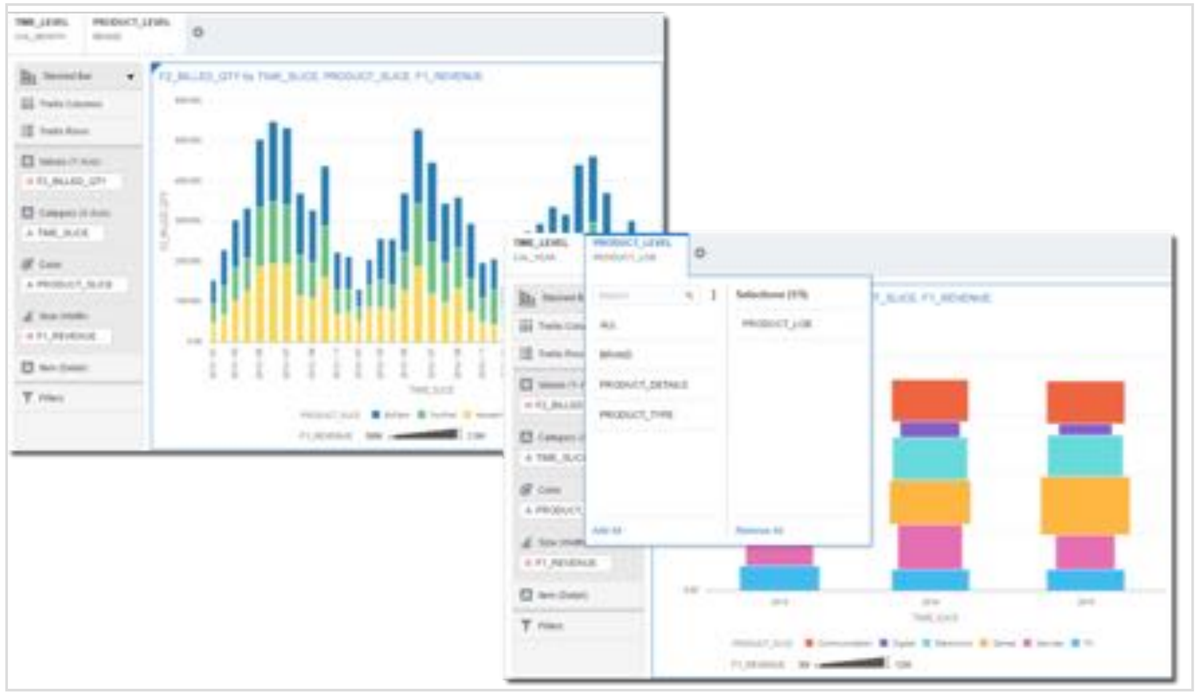


Analytic Views in DVD/DVCS/OAC

Statement

```
SELECT TIME_HIER.MEMBER_NAME AS TIME_SLICE,
  PRODUCT_HIER.MEMBER_NAME   AS PRODUCT_SLICE,
  TIME_HIER.LEVEL_NAME AS TIME_LEVEL,
  PRODUCT_HIER.LEVEL_NAME AS PRODUCT_LEVEL,
  TIME_HIER.HIER_ORDER AS TIME_HIER_ORDER,
  PRODUCT_HIER.HIER_ORDER AS PRODUCT_HIER_ORDER,
  F1_REVENUE,
  F2_BILLED_QTY
FROM F0_SALES_BASE_MEASURES
ORDER BY TIME_HIER.HIER_ORDER,
  PRODUCT_HIER.HIER_ORDER;
```

BISAMPLE
- TIME_SLICE
- PRODUCT_SLICE
- TIME_LEVEL
- PRODUCT_LEVEL
- F1_REVENUE
- F2_BILLED_QTY
- TIME_HIER_ORDER
- PRODUCT_HIER_ORDER



Reusability vs Performance

All Levels

```
SELECT TIME_HIER.MEMBER_NAME AS TIME_SLICE,
  PRODUCT_HIER.MEMBER_NAME   AS PRODUCT_SLICE,
  TIME_HIER.LEVEL_NAME AS TIME_LEVEL,
  PRODUCT_HIER.LEVEL_NAME AS PRODUCT_LEVEL,
  TIME_HIER.HIER_ORDER AS TIME_HIER_ORDER,
  PRODUCT_HIER.HIER_ORDER AS PRODUCT_HIER_ORDER,
  F1_REVENUE,
  F2_BILLED_QTY
FROM F0_SALES_BASE_MEASURES
ORDER BY TIME_HIER.HIER_ORDER,
  PRODUCT_HIER.HIER_ORDER;
```

vs

One Level

```
SELECT TIME_HIER.MEMBER_NAME AS TIME_SLICE,
  PRODUCT_HIER.MEMBER_NAME   AS PRODUCT_SLICE,
  F1_REVENUE,
  F2_BILLED_QTY
FROM F0_SALES_BASE_MEASURES
WHERE TIME_HIER.LEVEL_NAME  IN ('CAL_MONTH')
AND PRODUCT_HIER.LEVEL_NAME IN ('BRAND')
ORDER BY TIME_HIER.HIER_ORDER,
  PRODUCT_HIER.HIER_ORDER;
```

Reusability vs Performance

- No Level Filter Pushdown
- Mixing Multiple Levels
- DataSource Reusability

All Levels

```
SELECT TIME_HIER.MEMBER_NAME AS TIME_SLICE,
  PRODUCT_HIER.MEMBER_NAME   AS PRODUCT_SLICE,
  TIME_HIER.LEVEL_NAME AS TIME_LEVEL,
  PRODUCT_HIER.LEVEL_NAME AS PRODUCT_LEVEL,
  TIME_HIER.HIER_ORDER AS TIME_HIER_ORDER,
  PRODUCT_HIER.HIER_ORDER AS PRODUCT_HIER_ORDER,
  F1_REVENUE,
  F2_BILLED_QTY
FROM F0_SALES_BASE_MEASURES
ORDER BY TIME_HIER.HIER_ORDER,
  PRODUCT_HIER.HIER_ORDER;
```

Reusability vs Performance

- Level Filter Pushdown
- One Datasource per Level

One Level

```
SELECT TIME_HIER.MEMBER_NAME AS TIME_SLICE,
  PRODUCT_HIER.MEMBER_NAME   AS PRODUCT_SLICE,
  F1_REVENUE,
  F2_BILLED_QTY
FROM F0_SALES_BASE_MEASURES
WHERE TIME_HIER.LEVEL_NAME  IN ('CAL_MONTH')
AND PRODUCT_HIER.LEVEL_NAME IN ('BRAND')
ORDER BY TIME_HIER.HIER_ORDER,
  PRODUCT_HIER.HIER_ORDER;
```

Analytics View Benefits

- Upfront Fact/Hierachy Definition
- Lowers the Knowledge Required To Query
- Standard SQL

Analytics View Limitations

- DB Definition
- Multiple Levels of Same Hierarchy
- Performance Issues - No Where Clause

Aren’t we going back to Mode 1?

- IT Driven

- No External Application
- Unique Source of Truth
- Flexible
- Easily Extendible

Light Mode 1

Photo by Ales Krivec on Unsplash

Should I use Analytic Views?

- Complex DataSource
- Predictable set of Facts and Aggregation Methods
- No Data Organisation Layer (OBIEE)

Photo by Nghia Le on Unsplash

Enabling Self-Service Analytics with Analytic Views & Data Visualization from Cloud to Desktop