

VERTICA



Stratebi & Vertica (Gold Partner Vertica)



José Carlos García – Analytics Project Manager

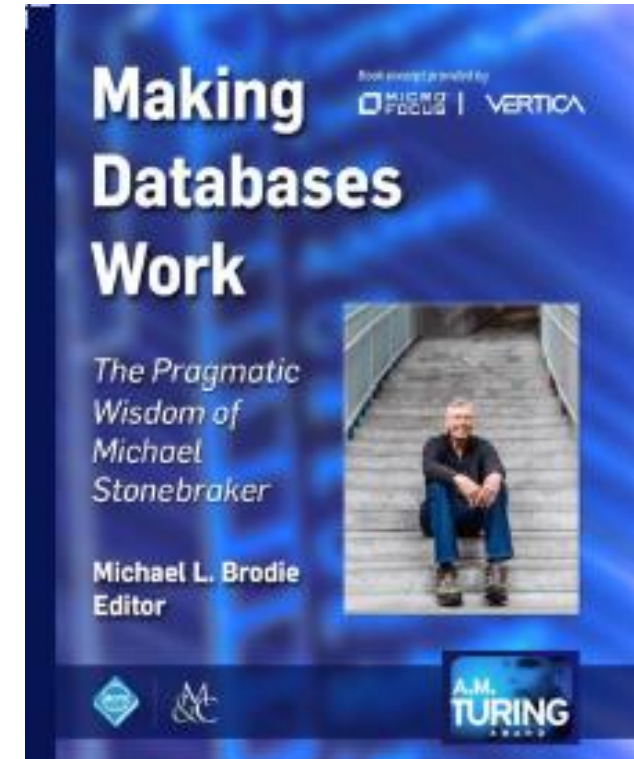
- ¿Qué es Vertica?
- Arquitectura y Características Principales
- Ejemplos de uso y casos reales
- Buenas Prácticas
- Demostración Vertica + Power BI(Direct Query)

¿Qué es Vertica?



Historia de la innovación

VERTICA



Situación DW Legacy

VERTICA

Testimonios:

Cada departamento tiene su sistema de reporting

Refresco de la información

Problemas de rendimiento en consultas

Acceso a la información desde dispositivos móviles

Generación manual de informes, cada semana, mes, etc...

Gobierno del dato limitado o inexistente, trazabilidad.

Problemas de calidad del dato

No disponemos de una plataforma colaborativa

El mismo indicador tiene distinto valor según quien lo genere

Acceso a la información

Los informes los tiene que generar IT, cuellos de botella....

Sistema Proactivo? Alertas?

Nos piden el dato agregado pero hay que mostrar el detalle

Se generan silos de información

Análisis descriptivo, pero y el predictivo/prescriptivo?

Inconsistencia en datos maestros, necesitamos un MDM?

Procesos de generación de informes duplicados

Mantenimiento de procedimientos almacenados y vistas mat

No existe una plataforma de datos centralizada: interno, proveedores, cliente

Aplicación manual de reglas de negocio.

¿Qué es Vertica?

VERTICA

Vertica es una plataforma analítica avanzada construida para ser escalable y dar respuesta al complejo mundo actual basado en datos. Combinando alta disponibilidad, motor de procesamiento MPP con funciones analíticas y de machine learning integradas.

SQL Database



Carga y almacena
datos en un data
warehouse
concebido para **una
analítica rapidísima**



Analytics & ML



Crea, entrena y despliega
análisis avanzados y
**modelos de machine
learning** a gran escala

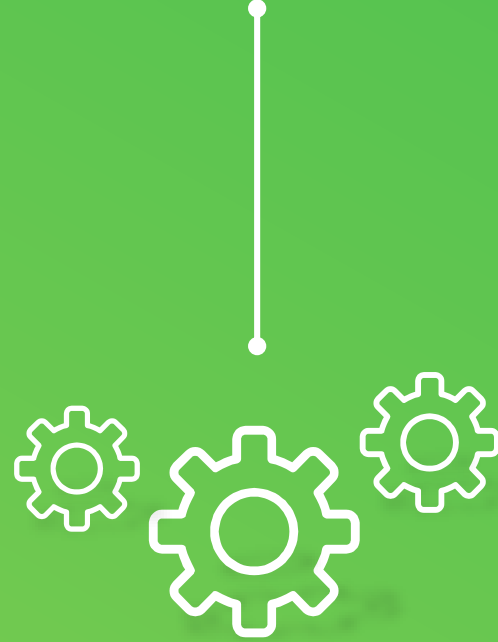


Query Engine



Realiza preguntas
complejas de forma
**independiente a donde
residan los datos**

Arquitectura





Almacenamiento Columnar

Acelera el tiempo de consulta al leer sólo los datos necesarios



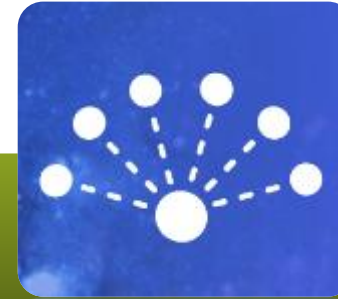
Compresión

Reduce las costosas operaciones E/S para mejorar el rendimiento



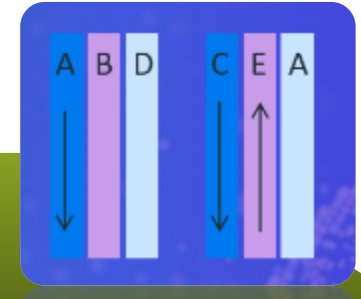
MPP Scale-out

Proporciona alta disponibilidad



Consulta Distribuida

Cualquier nodo puede iniciar las consultas y utilizar el resto de nodos para su ejecución



Proyecciones

Combinación de alta disponibilidad con optimización en consultas

Arquitectura

VERTICA

Big Data ETL



Carga Real-Time



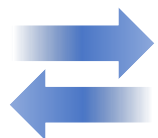
ETL



Funciones Definidas por el Usuario (UDF)

R	Java	C++	Python	SQL
Geoespacial	Real-Time		Análisis de Textos	
Series de Eventos	VERTICA		Reconocimiento Patrones	
Series Temporales	Machine Learning		Regresión	
Almacenamiento y Procesamiento Distribuido				
Seguridad				

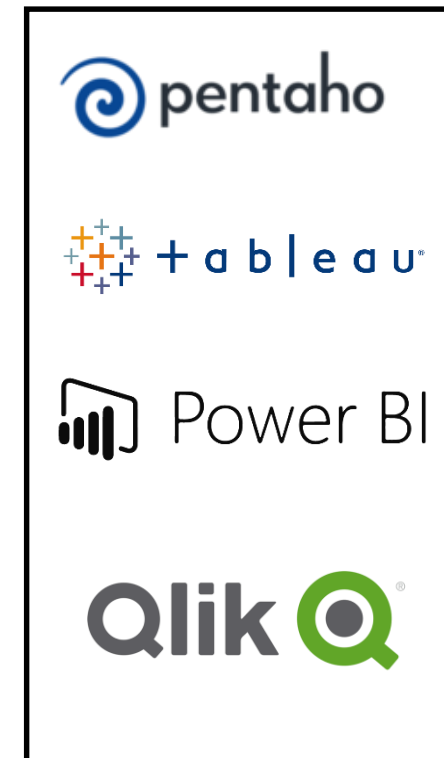
CARGA
DEFINIDA
USUARIO



ODBC
JDBC
OLEDB



BI & Visualización



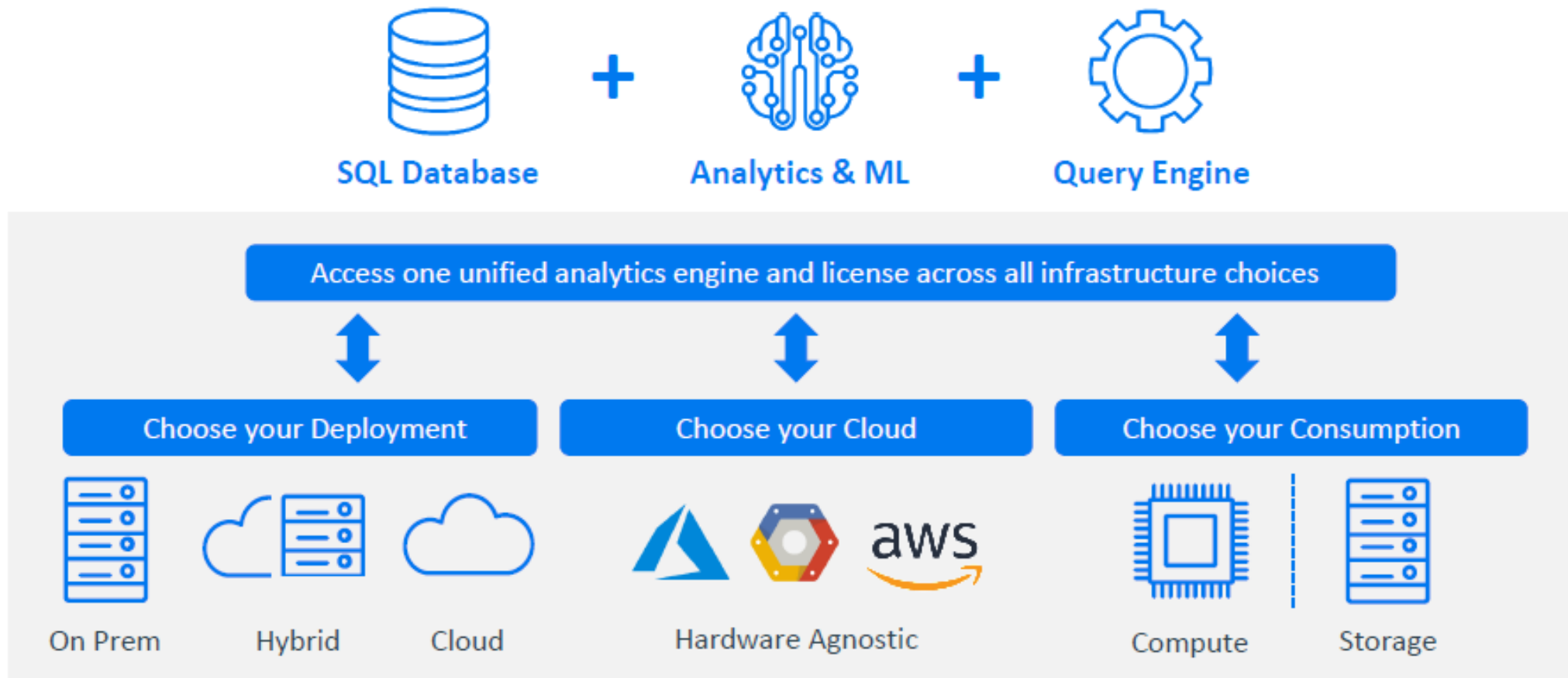
Integración con almacenamiento y tablas de sistemas



Libertad de Despliegue

VERTICA

El tipo de despliegue que decida hoy no le afectara en el futuro



End-to-end machine learning



Vertica soporta todo el proceso de análisis predictivo



Data Analysis	Data Preparation	Modeling	Evaluation	Deployment
• Statistical Summary • Time Series • Sessionize • Pattern Matching • Date/Time Algebra • Window Partition • Sequences • And more...	• Outlier Detection • Normalization • Imbalanced Data Processing • Sampling • Missing Value Imputation • And More...	• SVM • Random Forests • Logistic Regression • Linear Regression • Ridge Regression • Naïve Bayes • Cross Validation • And More...	• Model-level Stats • ROC Tables • Error Rate • Lift Table • Confusion Matrix • R-Squared • MSE • And More...	• Deploy Anywhere • In Database Scoring • Massively Parallel Processing • Speed • Scale • Security • And More...

MACHINE LEARNING FUNCTION

- KMEANS
- LINEAR_REG
- LOGISTIC_REG
- NAIVE_BAYES
- RF_CLASSIFIER
- RF_REGRESSOR
- SVM_CLASSIFIER
- SVM_REGRESSOR



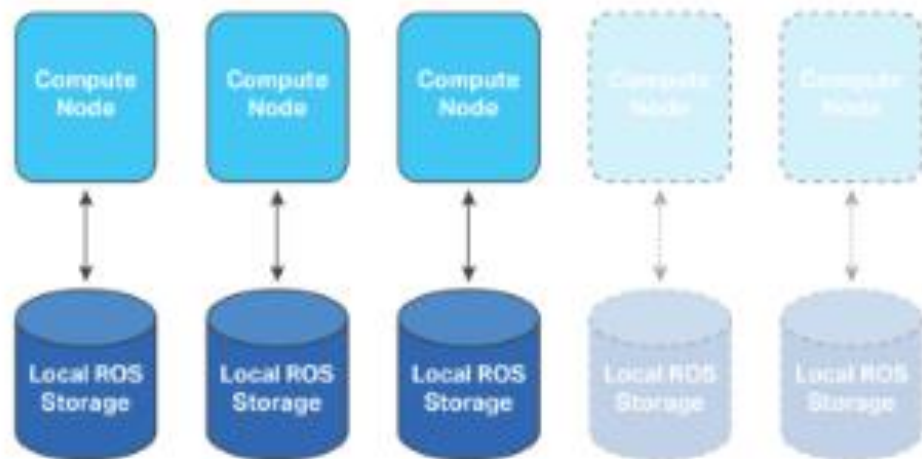
	KMEANS	RF CLAS	RF REG	REG LOG	SVM CLAS	SVM REG	NAIVE BAYES
SEGMENTACIÓN DE CLIENTES	✓						
DETECCIÓN DE FRAUDE		✓		✓	✓		✓
CONCESIÓN DE SEGUROS		✓		✓	✓		✓
RIESGO DE MOROSIDAD		✓		✓	✓		✓
FUGA DE CLIENTES		✓		✓	✓		✓
SISTEMAS DE RECOMENDACIÓN		✓		✓	✓		✓
INTENCIÓN DE VOTOS EN CLIENTES		✓		✓	✓		✓
PREDICCIÓN DE VENTAS			✓			✓	
PREDICCIÓN DEL NÚMERO DE CANCELACIONES			✓			✓	
ESTIMACIÓN EN ESTANCIAS HOSPITALARIAS			✓			✓	

Modos de almacenamiento

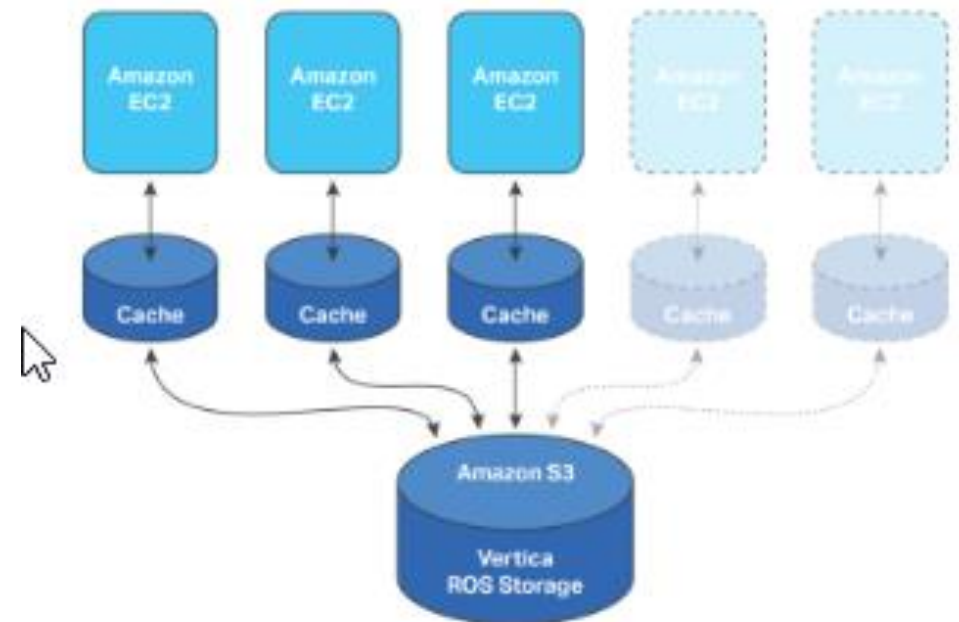
VERTICA

Diferentes posibilidades:

Enterprise Mode



Eon Mode

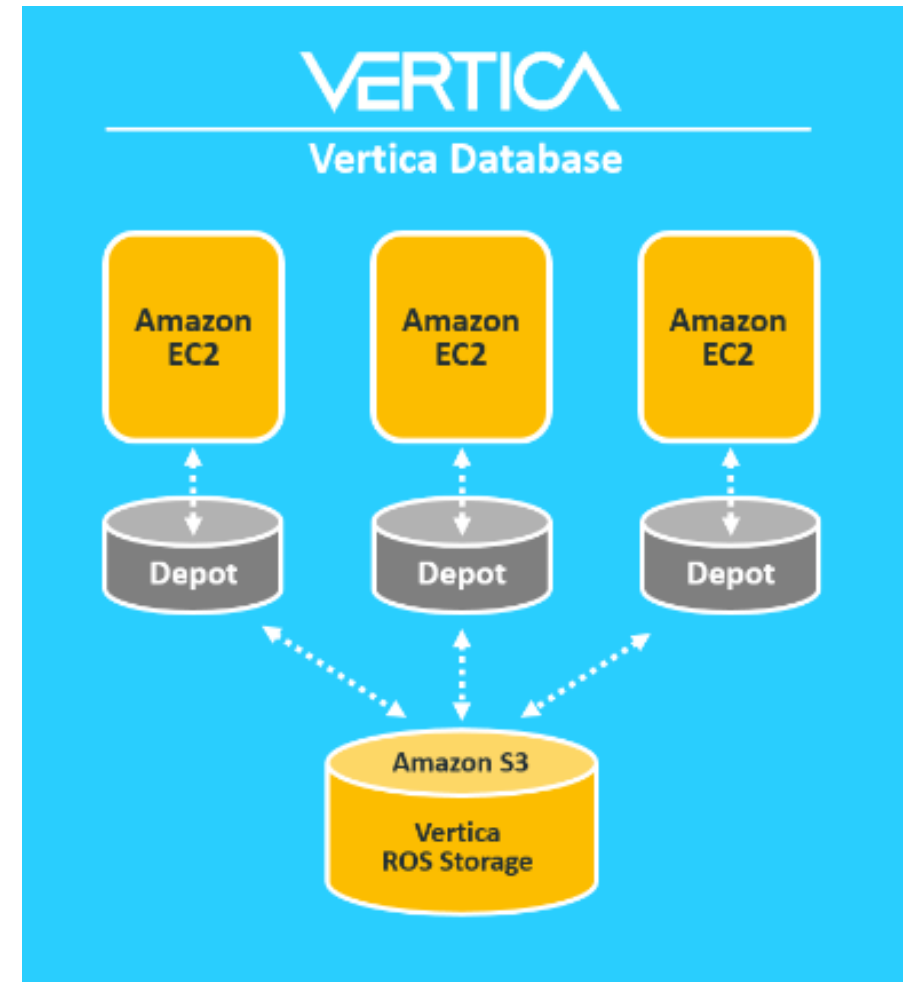


Modos de almacenamiento

VERTICA EON ABRE UN MUNDO DE POSIBILIDADES

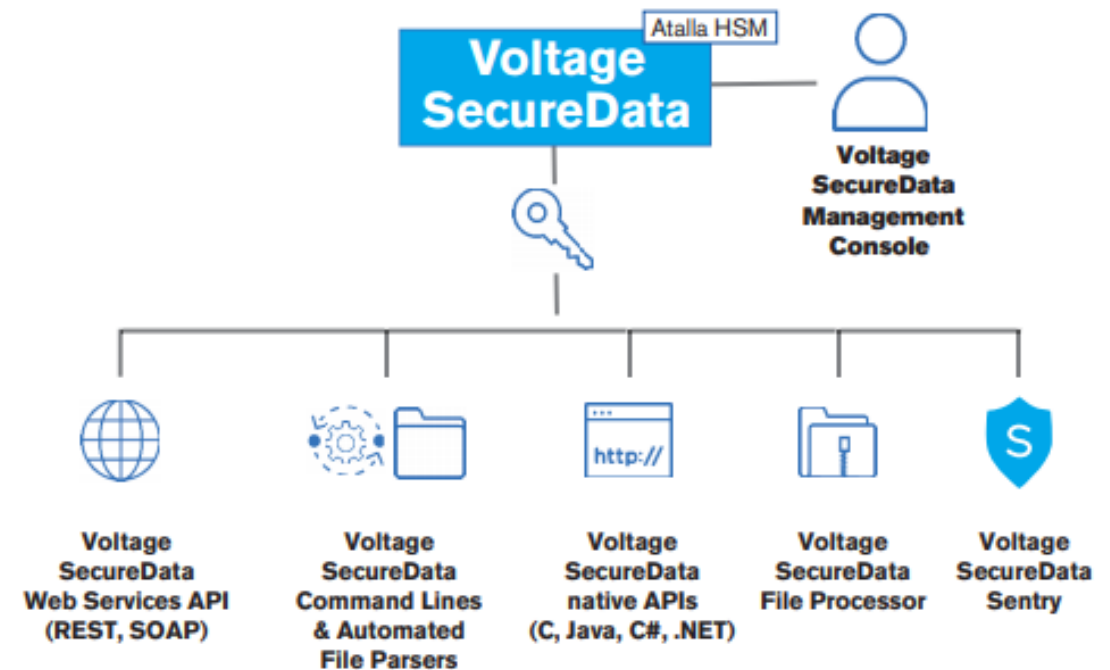
- Arquitectura de análisis de última generación
- Separa procesamiento y almacenamiento
- Escala elástica
- Maximización de economía *cloud*
- Suporta *Workloads* dinámicos
- Simplifica operaciones en base de datos
- Posibilita automatización para *analytics*

VERTICA

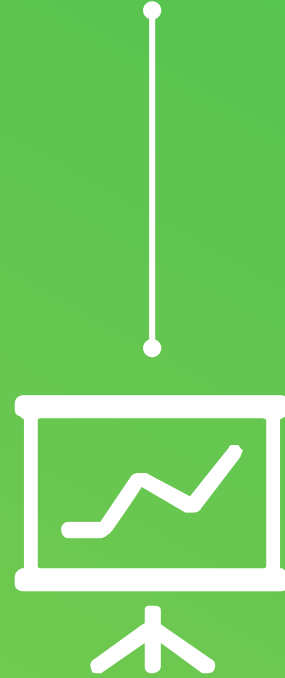


VOLTAGE SECUREDATA PROTEGE CONTRA HACKERS

Para ayudar a mitigar los riesgos asociados con el procesamiento de datos confidenciales, GDPR recomienda encriptar sus datos. El complemento Voltage SecureData para Vertica ofrece un enfoque único centrado en los datos para el cifrado de datos en Vertica, con información confidencial preservada de principio a fin, desde el momento de la captura, pasando por las aplicaciones de análisis de negocio, hasta el almacenamiento de datos en el back-end.



Referencias



Más de **1.000 empresas** ya usan **Vertica** a nivel mundial

- En StrateBI somos partners <https://www.vertica.com/partner/stratebi/>

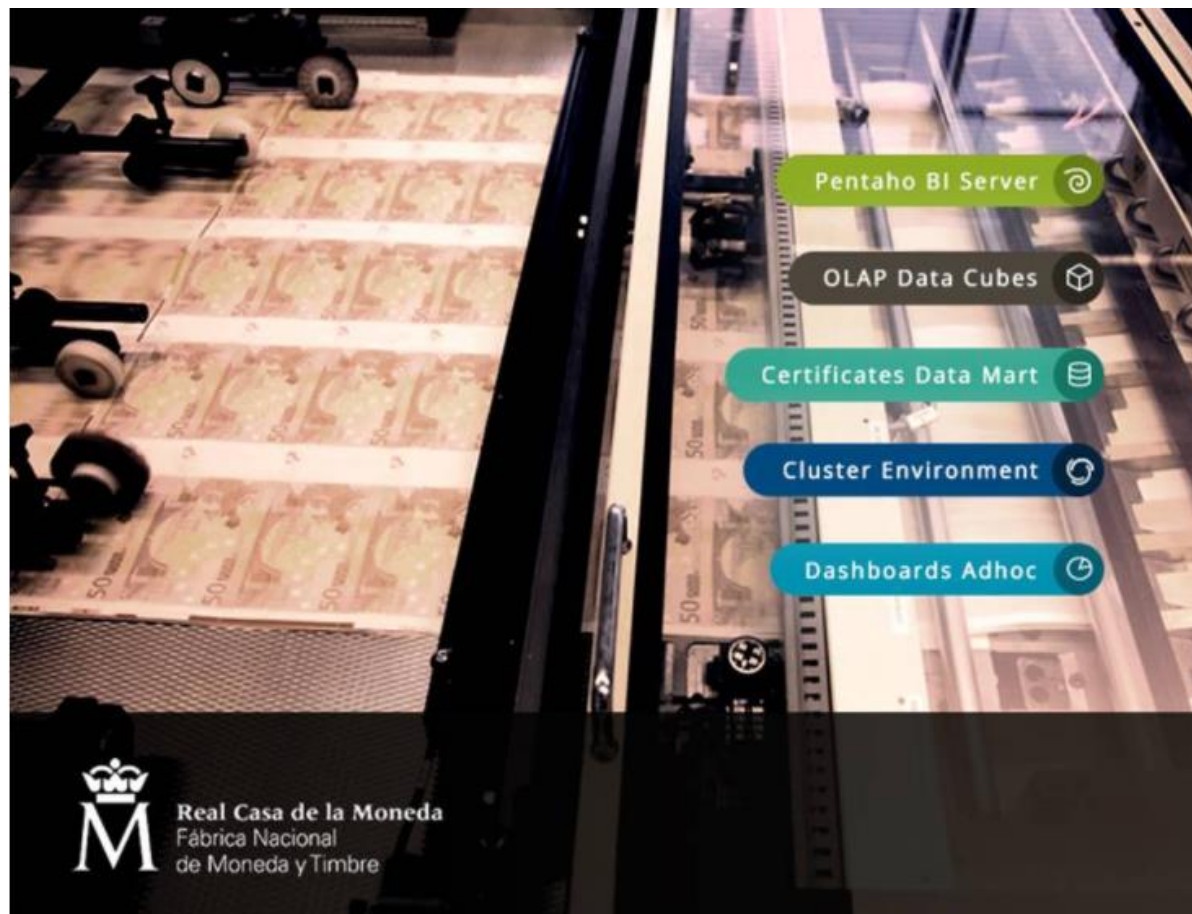




Toyota tenía la intención de desarrollar un sistema de BI para un solo módulo de la empresa. Unos años más tarde, Toyota y Stratebi siguen juntos y la empresa cuenta ahora con un sistema 360 totalmente automatizado e integrado con fines analíticos.

El departamento financiero fue el primero. Luego seguido por otros departamentos, como los sectores de operaciones y gestión de la empresa.

El sistema de BI utiliza Vertica y ya está en funcionamiento. Cuenta con un tiempo de respuesta optimizado para consultas de datos complejas, según se requería inicialmente.



La Fabrica Nacional de Moneda y Timbre no disponía de un sistema *business intelligence* y tenía la necesidad de poder analizar la información de los diferentes aplicativos.

En el proceso de modelado se hizo hincapié en implementar una estructura de análisis que permitiese analizar los certificados adaptándose a la complejidad de su estructura organizativa, formada por una estructura piramidal.

Stratebi optó por Vertica como una solución fiable y escalable. En la actualidad, más de 1.000 usuarios del sistema de la FNMT pueden acceder a información estadística a partir de certificados electrónicos y datos de gestión.



La empresa líder en Alarmas y Seguridad selecciono a Stratebi para implantar su sistema de ciberseguridad, integrando las diversas redes sociales ofrecidas por Datashift.

El sistema extrae información de diferentes fuentes para cumplir con el objetivo de proporcionar información útil en el análisis de sus clientes.

Las fuentes de datos son: blogs, boards, facebook, news, reddit, tumblr, twitter, wordpress y youtube.

Nos apoyamos en una base de datos analítica como Vertica, columnar y fácilmente escalable.

Referencias

VERTICA

Casos de Éxito

Telefónica

 **Verifica**
Part of B2Holding

 **TOYOTA**
FINANCIAL SERVICES



**TETÃ VIRU
MOHENDAPY**
Motenondeha
Ministerio de
HACIENDA



Real Casa de la Moneda
Fábrica Nacional
de Moneda y Timbre



 **Hankook**
masters

Carlider
LASSA
TYRES

ID
LOGISTICS


PROSEGUR

Demo



CLUSTER VERTICA DEMO

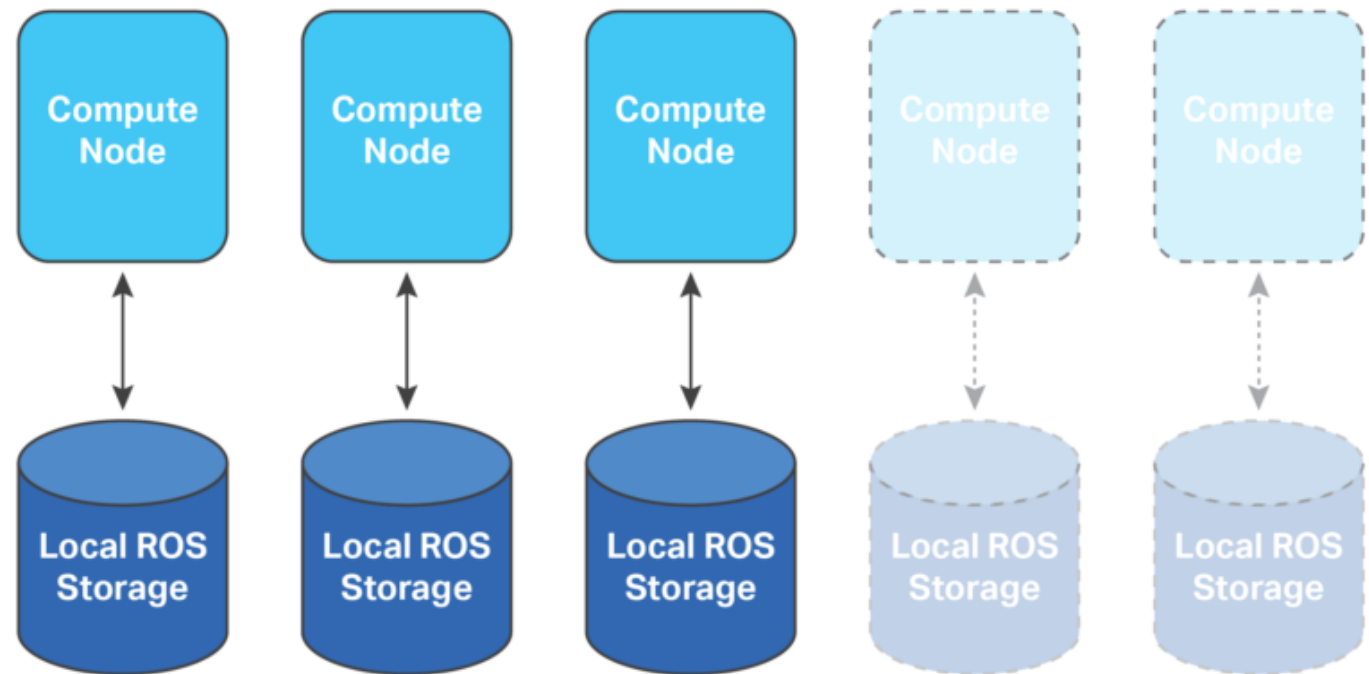
VERTICA

3 Nodos Cloud Public

Especificaciones de cada nodo:

- 8 vCPU(no dedicados)
- 32 GB RAM
- 240 Disk space

Enterprise Mode



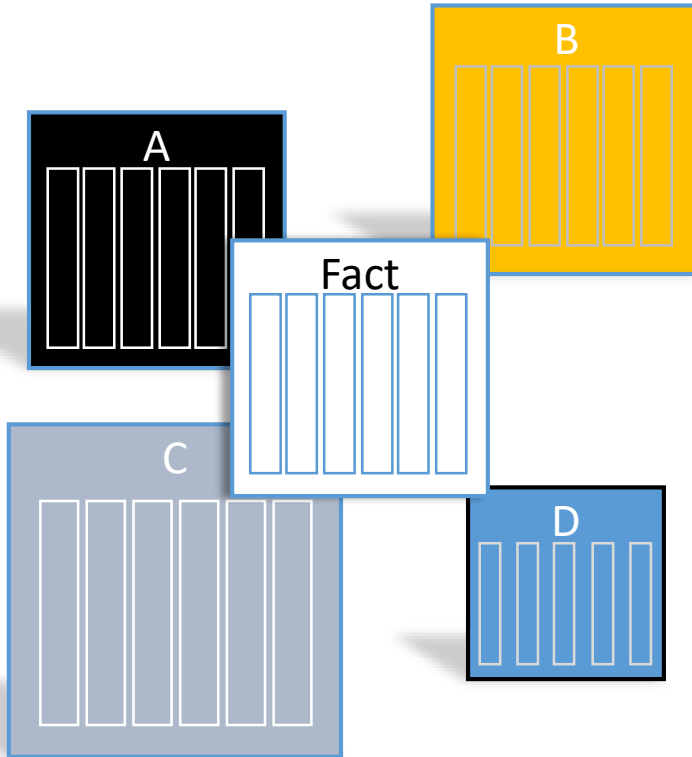
FLATTENED TABLES

Reto

El diseño de bases de datos normalizado a menudo utiliza un modelo de esquema en forma de estrella, que comprende múltiples **tablas de datos de hechos** de gran tamaño y muchas **tablas de dimensiones** más pequeñas.

Las consultas típicamente involucran uniones entre una tabla de hechos y tablas de múltiples dimensiones.

Penalización por JOINS



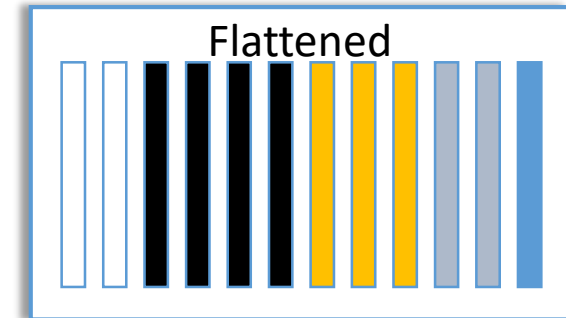
Una solución

Crear **tablas aplanadas** que combinen todas las columnas de tablas de hechos y dimensiones que las consultas necesiten.

Pros: Acelerar la ejecución de consultas.

Contras: El mantenimiento de conjuntos redundantes de datos normalizados y desnormalizados tiene **costes ETL**.

VERTICA



Mejor solución

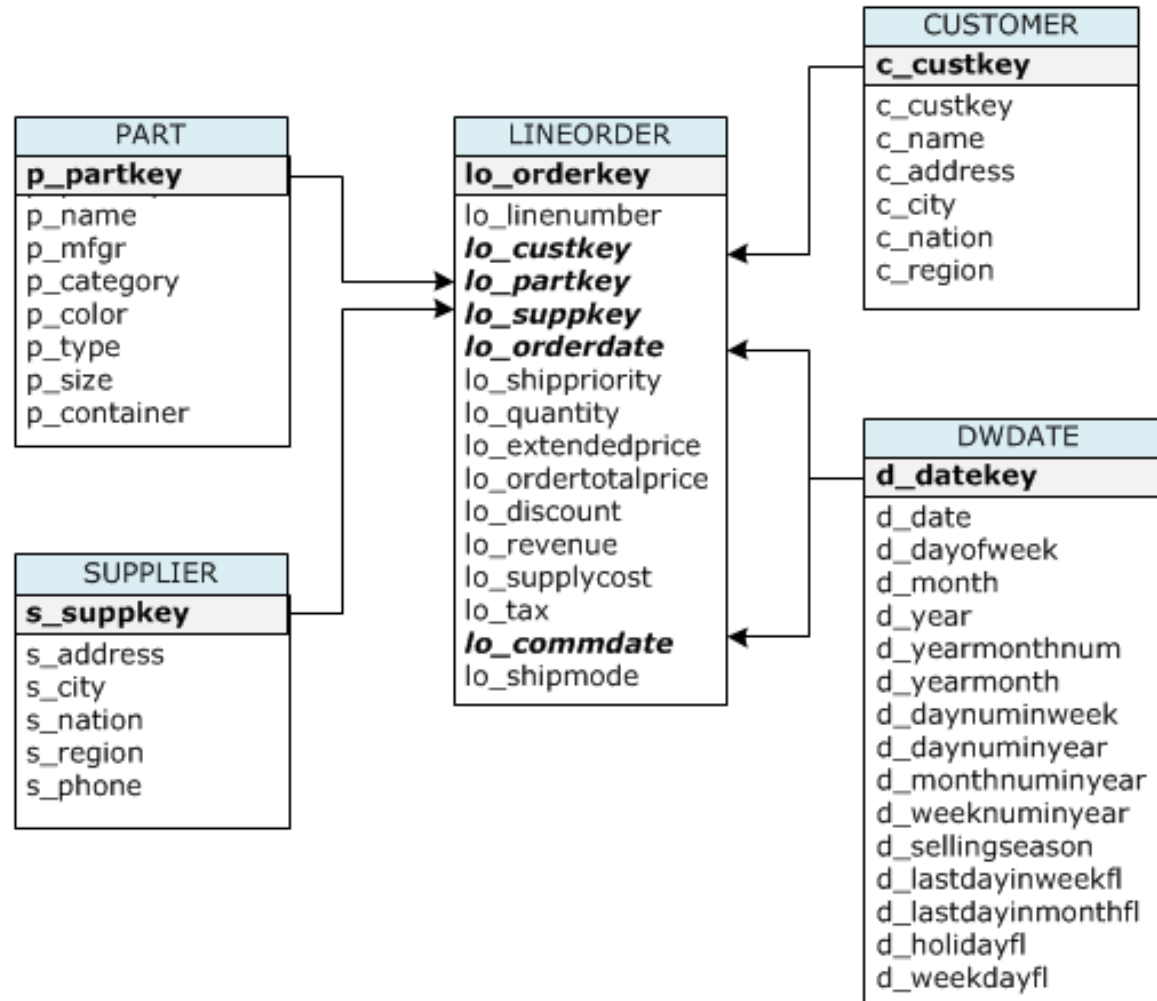
Las flattened de Vertica pueden incluir columnas que cargan sus valores consultando otras tablas.

Las operaciones en las tablas de origen y en la tabla aplanada están desacopladas; los cambios en una de ellas no se propagan automáticamente a la otra.

Esto minimiza la sobrecarga que es típica de las tablas desnormalizadas.

FLATTENED TABLES

VERTICA



```
Select count(*)  
from Evento_BI.p_lineorder_100;
```

123 count
119,972,104

1 row(s) fetched - 202ms

FLATTENED TABLES



Consultas con JOIN y sin aplicar optimizador Database Designer

```
-- Q1.1 --
select sum(v_revenue) as revenue
from Evento_BI.p_lineorder_100 f
left join Evento_BI.dates d on f.lo_orderdate = d.d_datekey
where d.d_year = 1993
and f.lo_discount between 1 and 3
and f.lo_quantity < 25;
```

123 revenue	1 row(s) fetched - 638ms
8,945,615,531,166	

```
-- Q1.2 --
select sum(f.v_revenue) as revenue
from Evento_BI.p_lineorder_100 f
left join Evento_BI.dates d on f.lo_orderdate = d.d_datekey
where d.d_yearmonthnum = 199401
and f.lo_discount between 4 and 6
and f.lo_quantity between 26 and 35;
```

123 revenue	1 row(s) fetched - 687ms
1.930.098.131.694	

```
-- Q1.3 --
select sum(f.v_revenue) as revenue
from Evento_BI.p_lineorder_100 f
left join Evento_BI.dates d on f.lo_orderdate = d.d_datekey
where d.d_weeknuminyear = 6 and d.d_year = 1994
and f.lo_discount between 5 and 7
and f.lo_quantity between 26 and 35;
```

123 revenue	1 row(s) fetched - 681ms
522,712,647,938	

```
-- Q2.1 --
select d.d_year, p.p_brand, sum(f.lo_revenue) as lo_revenue
from Evento_BI.p_lineorder_100 f
left join Evento_BI.dates d on f.lo_orderdate = d.d_datekey
left join Evento_BI.part p on f.lo_partkey = p.p_partkey
left join Evento_BI.supplier s on f.lo_suppkey = s.s_suppkey
where p.p_category = 'MFGR#12' and s.s_region = 'AMERICA'
group by d.d_year, p.p_brand
order by d.d_year, p.p_brand;
```

123 d_year	ABC p_brand	123 lo_revenue	200 row(s) fetched - 8.502s (+192ms)
1,992	MFGR#1201	12,963,864,694	
1,992	MFGR#1202	13,840,263,702	
1,992	MFGR#1203	13,502,210,684	

FLATTENED TABLES

VERTICA

Consultas con JOIN y sin aplicar optimizador Database Designer

```
-- Q2.2 --
select d.d_year, p.p_brand, sum(f.lo_revenue) as lo_revenue
from Evento_BI.p_lineorder_100 f
left join Evento_BI.dates d on f.lo_orderdate = d.d_datekey
left join Evento_BI.part p on f.lo_partkey = p.p_partkey
left join Evento_BI.supplier s on f.lo_suppkey = s.s_suppkey
where p.p_brand between 'MFGR#2221' and 'MFGR#2228' and s.s_region = 'ASIA'
group by d.d_year, p.p_brand
order by d.d_year, p.p_brand;
```

123 d_year	ABC p_brand	123 lo_revenue	56 row(s) fetched - 8.612s
1,992	MFGR#2221	13,136,902,340	
1,992	MFGR#2222	12,764,123,956	
1,992	MFGR#2223	13,268,193,022	

```
-- Q2.3 --
select d.d_year, p.p_brand, sum(f.lo_revenue) as lo_revenue
from Evento_BI.p_lineorder_100 f
left join Evento_BI.dates d on f.lo_orderdate = d.d_datekey
left join Evento_BI.part p on f.lo_partkey = p.p_partkey
left join Evento_BI.supplier s on f.lo_suppkey = s.s_suppkey
where p.p_brand = 'MFGR#2239' and s.s_region = 'EUROPE'
group by d.d_year, p.p_brand
order by d.d_year, p.p_brand;
```

123 d_year	ABC p_brand	123 lo_revenue	7 row(s) fetched - 8.400s
1,992	MFGR#2239	12,922,064,618	
1,993	MFGR#2239	12,983,586,706	
1,994	MFGR#2239	12,277,199,144	

```
-- Q3.1 --
select c.c_nation, s.s_nation, d.d_year, sum(f.lo_revenue) as lo_revenue
from Evento_BI.p_lineorder_100 f
left join Evento_BI.dates d on f.lo_orderdate = d.d_datekey
left join Evento_BI.customer c on f.lo_custkey = c.c_custkey
left join Evento_BI.supplier s on f.lo_suppkey = s.s_suppkey
where c.c_region = 'ASIA' and s.s_region = 'ASIA' and d.d_year >= 1992 and d.d_year <= 1997
group by c.c_nation, s.s_nation, d.d_year
order by d.d_year asc, lo_revenue desc;
```

ABC c_nation	ABC s_nation	123 d_year	123 lo_revenue	150 row(s) fetched - 7.521s (+2ms)
INDONESIA	INDONESIA	1,992	114,941,142,132	
INDIA	INDONESIA	1,992	112,716,551,524	
VIETNAM	INDONESIA	1,992	110,465,372,098	

```
-- Q3.2 --
select c.c_city, s.s_city, d.d_year, sum(f.lo_revenue) as lo_revenue
from Evento_BI.p_lineorder_100 f
left join Evento_BI.dates d on f.lo_orderdate = d.d_datekey
left join Evento_BI.customer c on f.lo_custkey = c.c_custkey
left join Evento_BI.supplier s on f.lo_suppkey = s.s_suppkey
where c.c_nation = 'UNITED STATES' and s.s_nation = 'UNITED STATES'
and d.d_year >= 1992 and d.d_year <= 1997
group by c.c_city, s.s_city, d.d_year
order by d.d_year asc, lo_revenue desc;
```

ABC c_city	ABC s_city	123 d_year	123 lo_revenue	200 row(s) fetched - 6.728s (+96ms)
UNITED ST5	UNITED ST7	1,992	1,360,190,870	
UNITED ST6	UNITED ST7	1,992	1,354,225,064	
UNITED ST9	UNITED ST7	1,992	1,352,263,706	

FLATTENED TABLES



Añadimos columnas dimensión a la tabla de hechos

```
ALTER TABLE evento_BI.P_LINEORDER_100 ADD COLUMN C_NAME varchar(25)
SET USING (SELECT C_NAME FROM evento_BI.CUSTOMER
WHERE CUSTOMER.C_CUSTKEY = P_LINEORDER_100.LO_CUSTKEY);
```

Actualizamos columnas dimensión de la tabla de hechos

```
SELECT REFRESH_COLUMNS ('Evento_BI.p_lineorder_100', '*');

SELECT REFRESH_COLUMNS ('Evento_BI.p_lineorder_100', 'C_NAME, C_ADDRESS');
```

```
CREATE TABLE evento_BI.P_LINEORDER_100
(
    LO_ORDERKEY int,
    LO_LINENUMBER int,
    LO_CUSTKEY int,
    LO_PARTKEY int,
    LO_SUPPKEY int,
    LO_ORDERDATE int NOT NULL,
    LO_ORDERPRIOTITY varchar(15),
    LO_SHIPPRIOTITY int,
    LO_QUANTITY int,
    LO_EXTENDEDPRICE int,
    LO_ORDTOTALPRICE int,
    LO_DISCOUNT int,
    LO_REVENUE int,
    LO_SUPPLYCOST int,
    LO_TAX int,
    LO_COMMITDATE int,
    LO_SHIPMODE varchar(10),
    V_REVENUE int,
    C_NAME varchar(25) SET USING ( SELECT CUSTOMER.C_NAME
FROM evento_BI.CUSTOMER
WHERE (CUSTOMER.C_CUSTKEY = P_LINEORDER_100.LO_CUSTKEY)),
    C_ADDRESS varchar(40) SET USING ( SELECT CUSTOMER.C_ADDRESS
FROM evento_BI.CUSTOMER
WHERE (CUSTOMER.C_CUSTKEY = P_LINEORDER_100.LO_CUSTKEY)),
    C_CITY varchar(10) SET USING ( SELECT CUSTOMER.C_CITY
FROM evento_BI.CUSTOMER
WHERE (CUSTOMER.C_CUSTKEY = P_LINEORDER_100.LO_CUSTKEY)),
    C_NATION varchar(15) SET USING ( SELECT CUSTOMER.C_NATION
FROM evento_BI.CUSTOMER
WHERE (CUSTOMER.C_CUSTKEY = P_LINEORDER_100.LO_CUSTKEY)),
```

FLATTENED TABLES

VERTICA

Consultas sin JOIN y sin aplicar optimizador Database Designer

```
-- Q1.1 --
select sum(v_revenue) as revenue
from Evento_BI.p_lineorder_100 f
where f.d_year = 1993
and f.lo_discount between 1 and 3
and f.lo_quantity < 25;
```

123 revenue	row(s) fetched - 880ms
8,945,615,531,166	

```
-- Q1.2 --
select sum(f.v_revenue) as revenue
from Evento_BI.p_lineorder_100 f
where f.d_yearmonthnum = 199401
and f.lo_discount between 4 and 6
and f.lo_quantity between 26 and 35;
```

123 revenue	1 row(s) fetched - 287ms
1,930,098,131,694	

```
-- Q1.3 --
select sum(f.v_revenue) as revenue
from Evento_BI.p_lineorder_100 f
where f.d_weeknuminyear = 6 and f.d_year = 1994
and f.lo_discount between 5 and 7
and f.lo_quantity between 26 and 35;
```

123 revenue	1 row(s) fetched - 685ms
522,712,647,938	

```
-- Q2.1 --
select f.d_year, f.p_brand, sum(f.lo_revenue) as lo_revenue
from Evento_BI.p_lineorder_100 f
where f.p_category = 'MFGR#12' and f.s_region = 'AMERICA'
group by f.d_year, f.p_brand
order by f.d_year, f.p_brand;
```

123 d_year	ABC p_brand	123 lo_revenue	200 row(s) fetched - 1.128s (+58ms)
1,992	MFGR#1201	12,963,864,694	
1,992	MFGR#1202	13,840,263,702	
1,992	MFGR#1203	13,502,210,684	

FLATTENED TABLES

VERTICA

Consultas sin JOIN y sin aplicar optimizador Database Designer

```
-- Q2.2 --
select f.d_year, f.p_brand, sum(f.lo_revenue) as lo_revenue
from Evento_BI.p_lineorder_100 f
where f.p_brand between 'MFGR#2221' and 'MFGR#2228' and f.s_region = 'ASIA'
group by f.d_year, f.p_brand
order by f.d_year, f.p_brand;
```

123 d_year	ABC p_brand	123 lo_revenue
1,992	MFGR#2221	13,136,902,340
1,992	MFGR#2222	12,764,123,956
1,992	MFGR#2223	13,268,193,022

56 row(s) fetched - 808ms (+1ms)

```
-- Q2.3 --
select f.d_year, f.p_brand, sum(f.lo_revenue) as lo_revenue
from Evento_BI.p_lineorder_100 f
where f.p_brand = 'MFGR#2239' and f.s_region = 'EUROPE'
group by f.d_year, f.p_brand
order by f.d_year, f.p_brand;
```

123 d_year	ABC p_brand	123 lo_revenue
1,992	MFGR#2239	12,922,064,618
1,993	MFGR#2239	12,983,586,706
1,994	MFGR#2239	12,277,199,144

7 row(s) fetched - 488ms

```
-- Q3.1 --
select f.c_nation, f.s_nation, f.d_year, sum(f.lo_revenue) as lo_revenue
from Evento_BI.p_lineorder_100 f
where f.c_region = 'ASIA' and f.s_region = 'ASIA'
and f.d_year >= 1992 and f.d_year <= 1997
group by f.c_nation, f.s_nation, f.d_year
order by f.d_year asc, lo_revenue desc;
```

150 row(s) fetched - 1.188s (+1ms)

ABC c_nation	ABC s_nation	123 d_year	123 lo_revenue
INDONESIA	INDONESIA	1,992	114,941,142,132
INDIA	INDONESIA	1,992	112,716,551,524
VIETNAM	INDONESIA	1,992	110,465,372,098

```
-- Q3.2 --
select f.c_city, f.s_city, f.d_year, sum(f.lo_revenue) as lo_revenue
from Evento_BI.p_lineorder_100 f
where f.c_nation = 'UNITED STATES' and f.s_nation = 'UNITED STATES'
and f.d_year >= 1992 and f.d_year <= 1997
group by f.c_city, f.s_city, f.d_year
order by f.d_year asc, lo_revenue desc;
```

200 row(s) fetched - 1.100s (+1ms)

ABC c_city	ABC s_city	123 d_year	123 lo_revenue
UNITED ST5	UNITED ST7	1,992	1,360,190,870
UNITED ST6	UNITED ST7	1,992	1,354,225,064
UNITED ST9	UNITED ST7	1,992	1,352,263,706

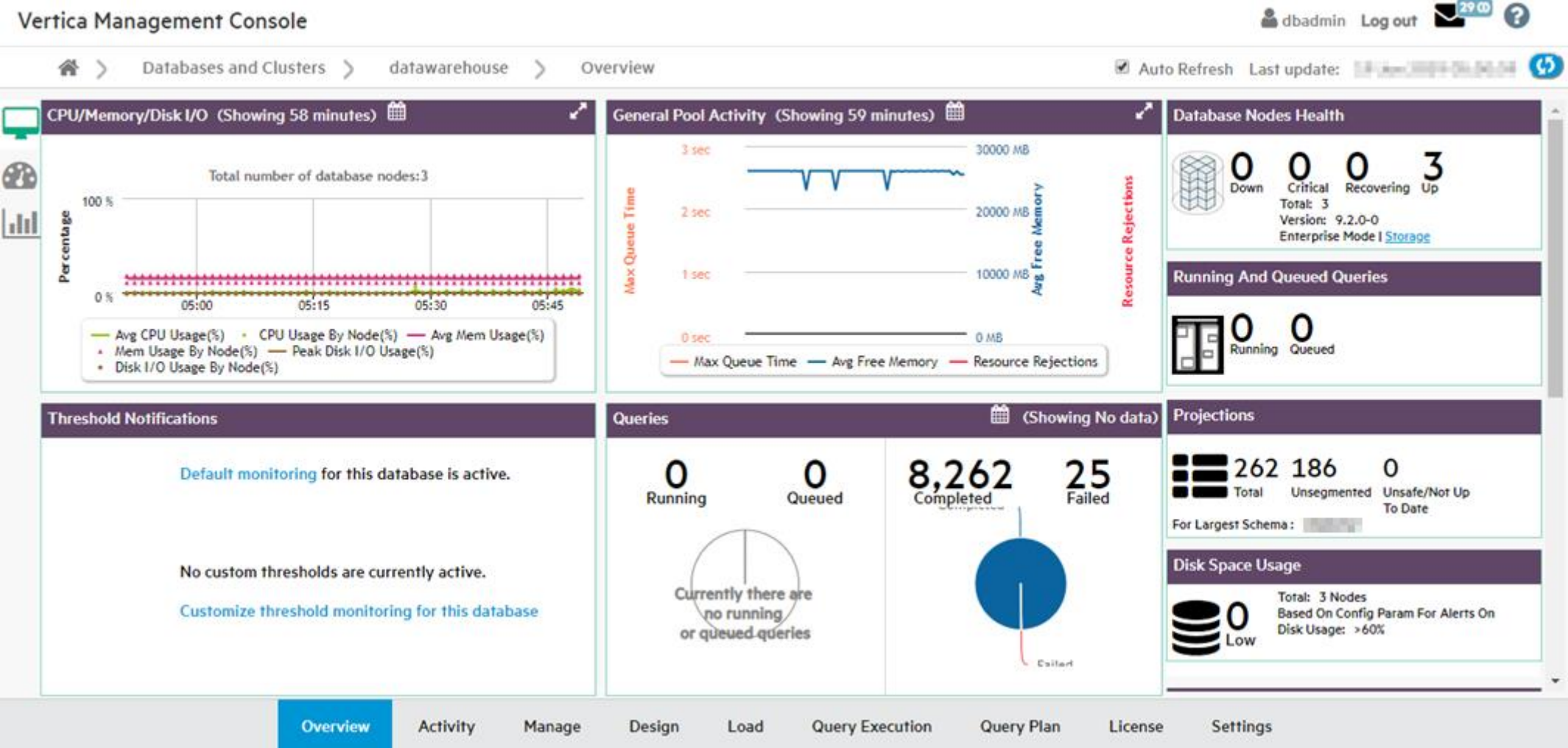
FLATTENED TABLES

VERTICA

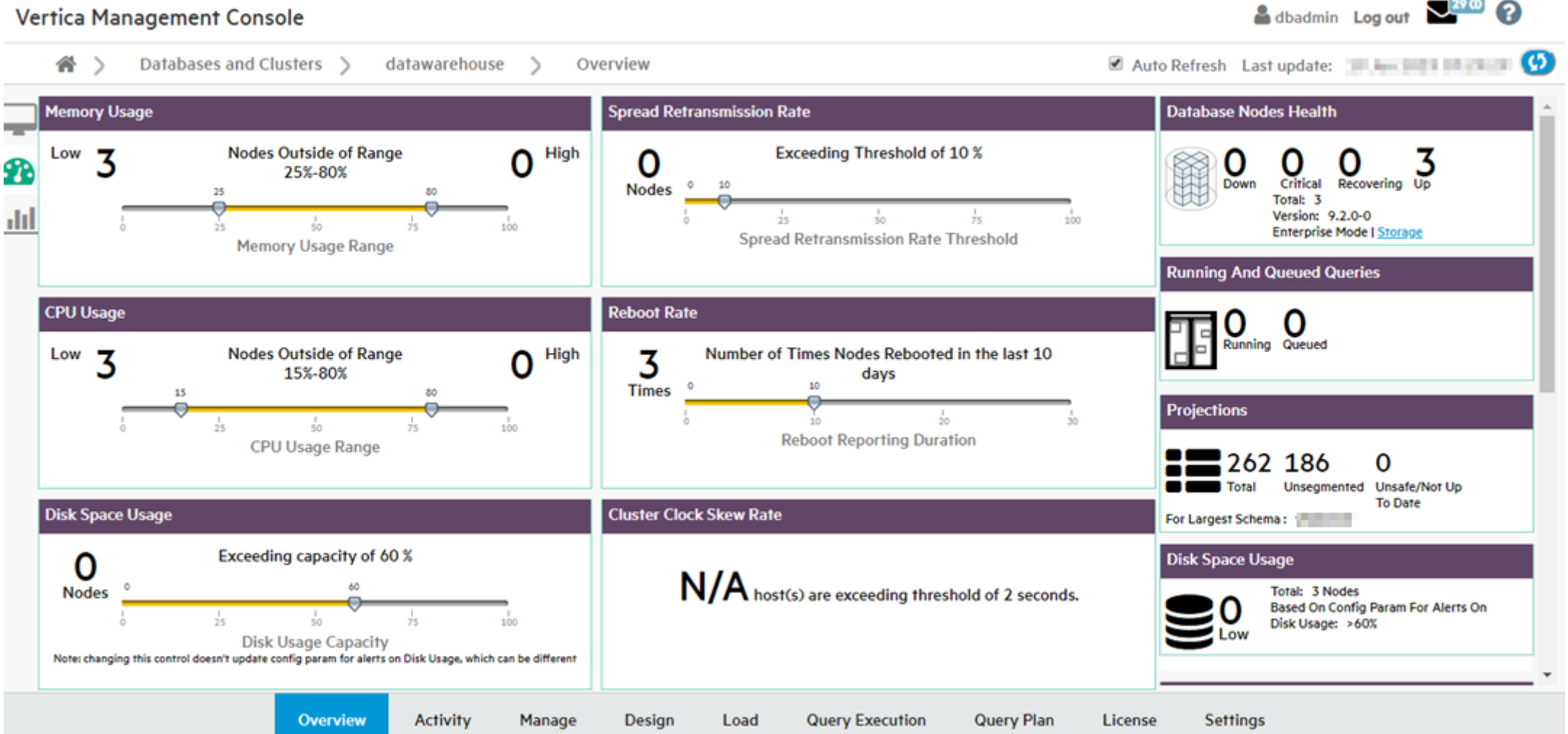
Resultados:

Consulta	Estrella(Joins)	Flattened
Q1.1	< 1 seg	< 1 seg
Q1.2	< 1 seg	< 1 seg
Q1.3	< 1 seg	< 1 seg
Q2.1	8,5 seg	1,1 seg
Q2.2	8,6 seg	< 1 seg
Q2.3	8,4 seg	< 1 seg
Q3.1	7,5 seg	1,1 seg
Q3.2	8,7 seg	1,1 seg

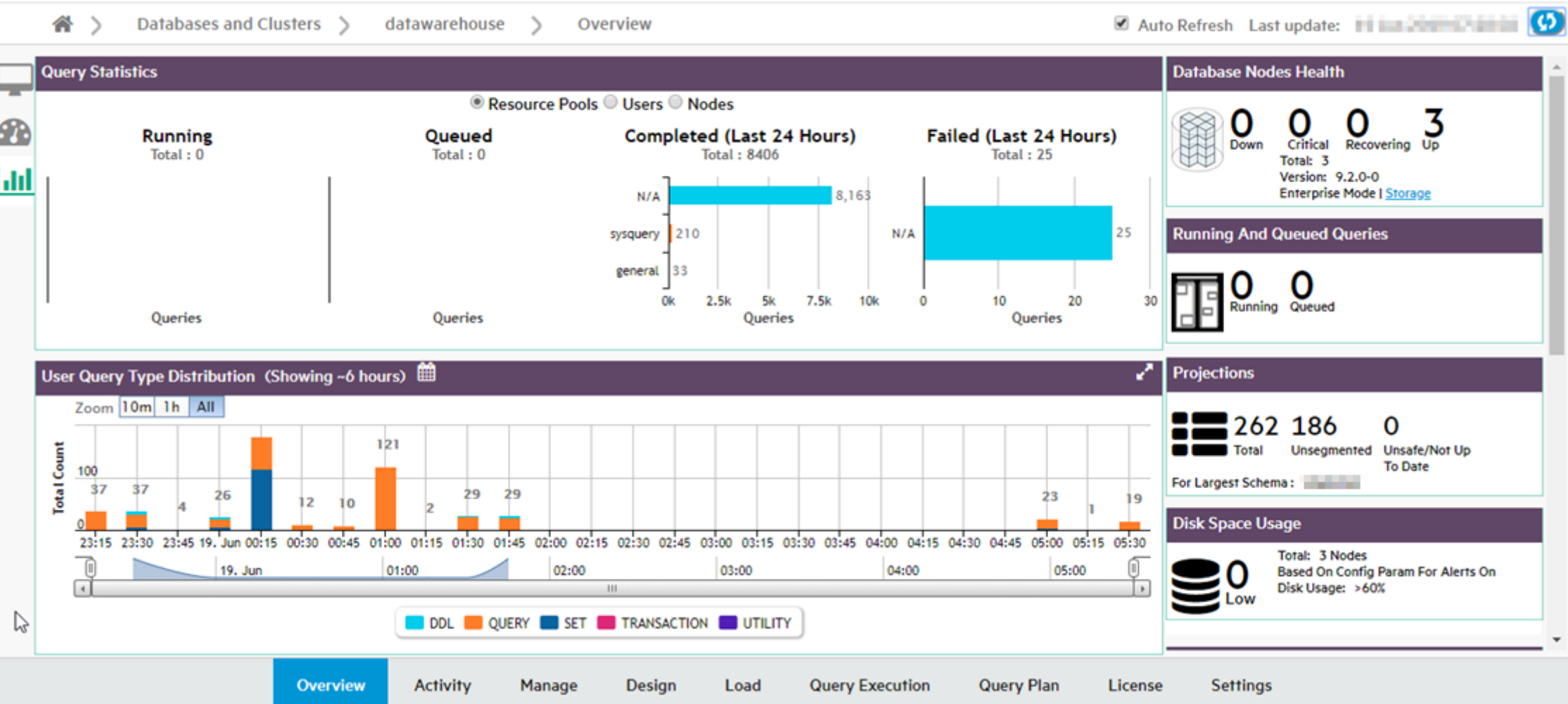
Management Console



Management Console



Management Console



Management Console

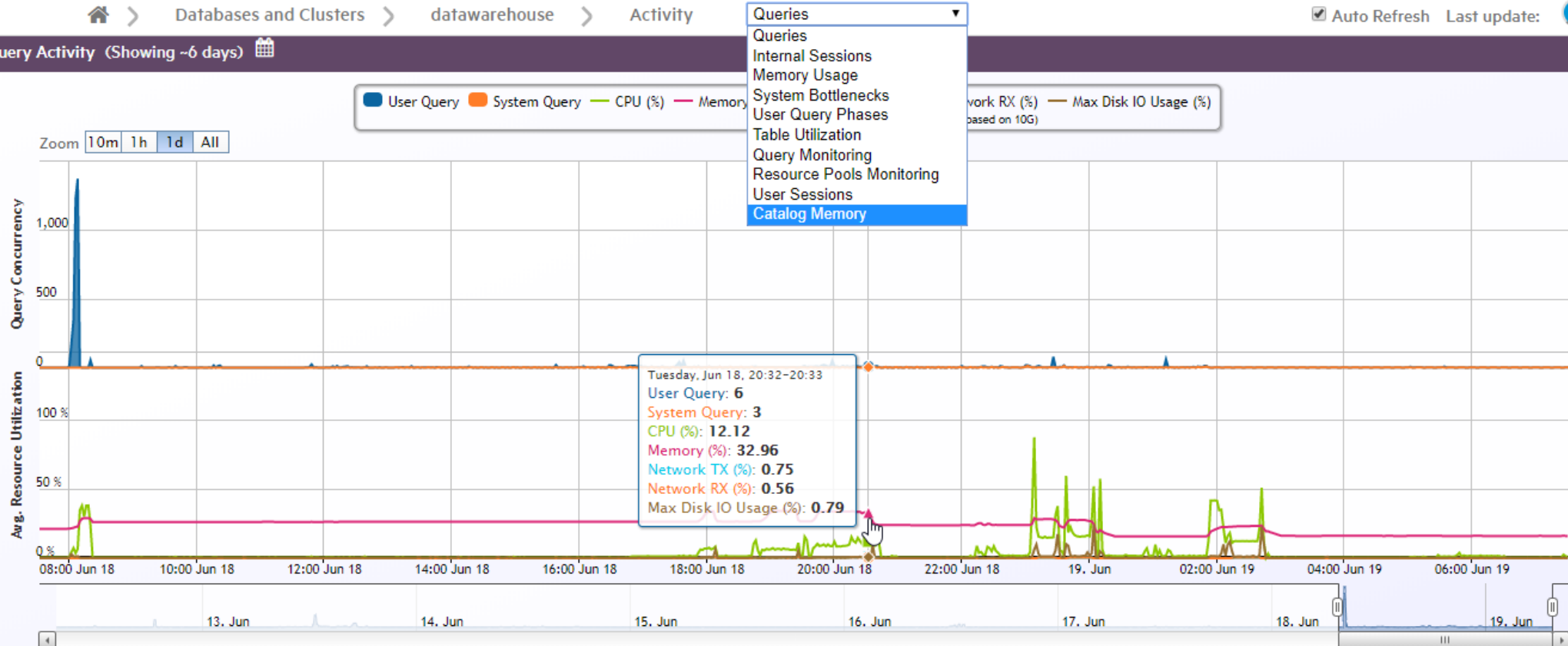
Vertica Management Console

dbadmin Log out 29 00 ?

Databases and Clusters > datawarehouse > Activity

Auto Refresh Last update:

Query Activity (Showing -6 days)



Overview

Activity

Manage

Design

Load

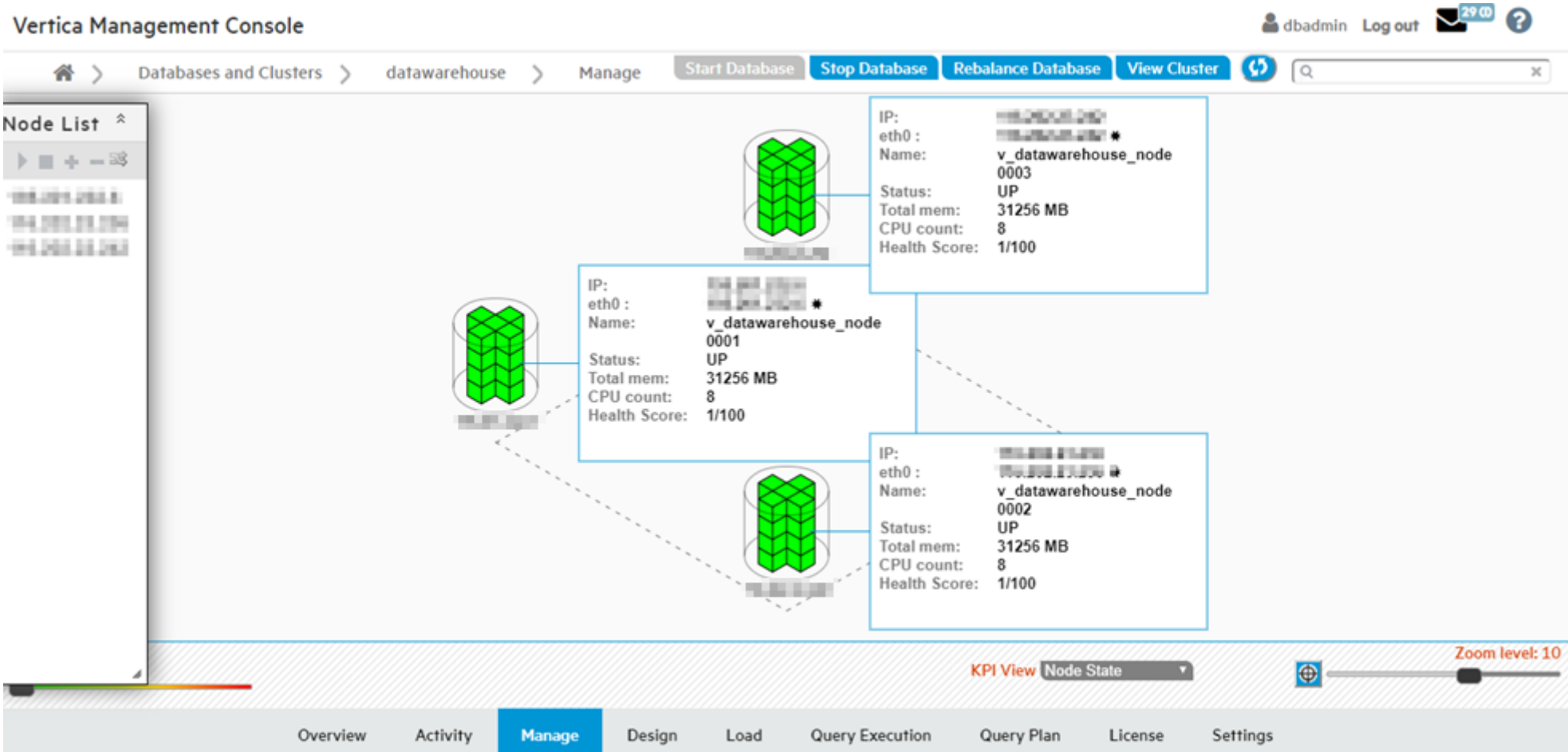
Query Execution

Query Plan

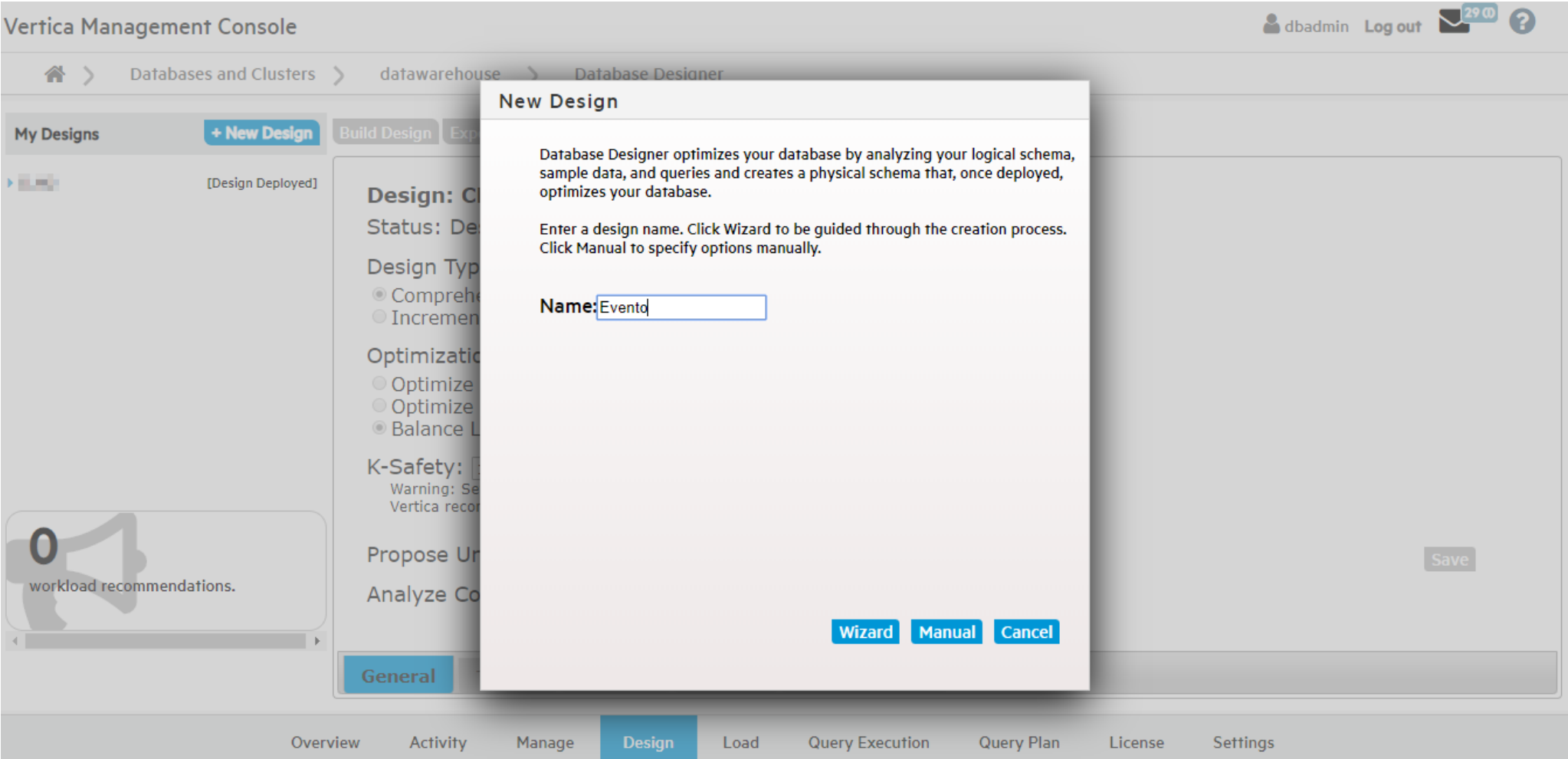
License

Settings

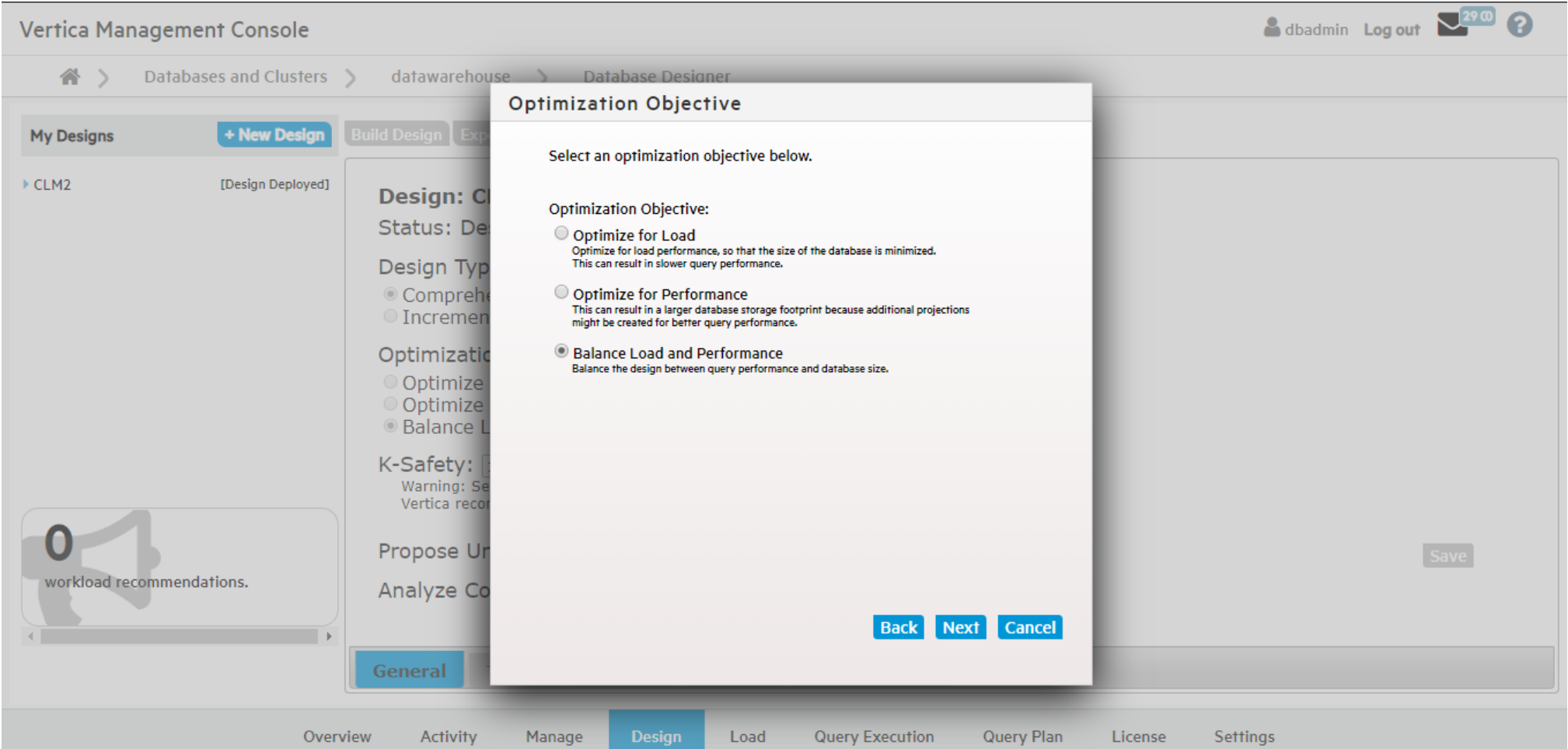
Management Console



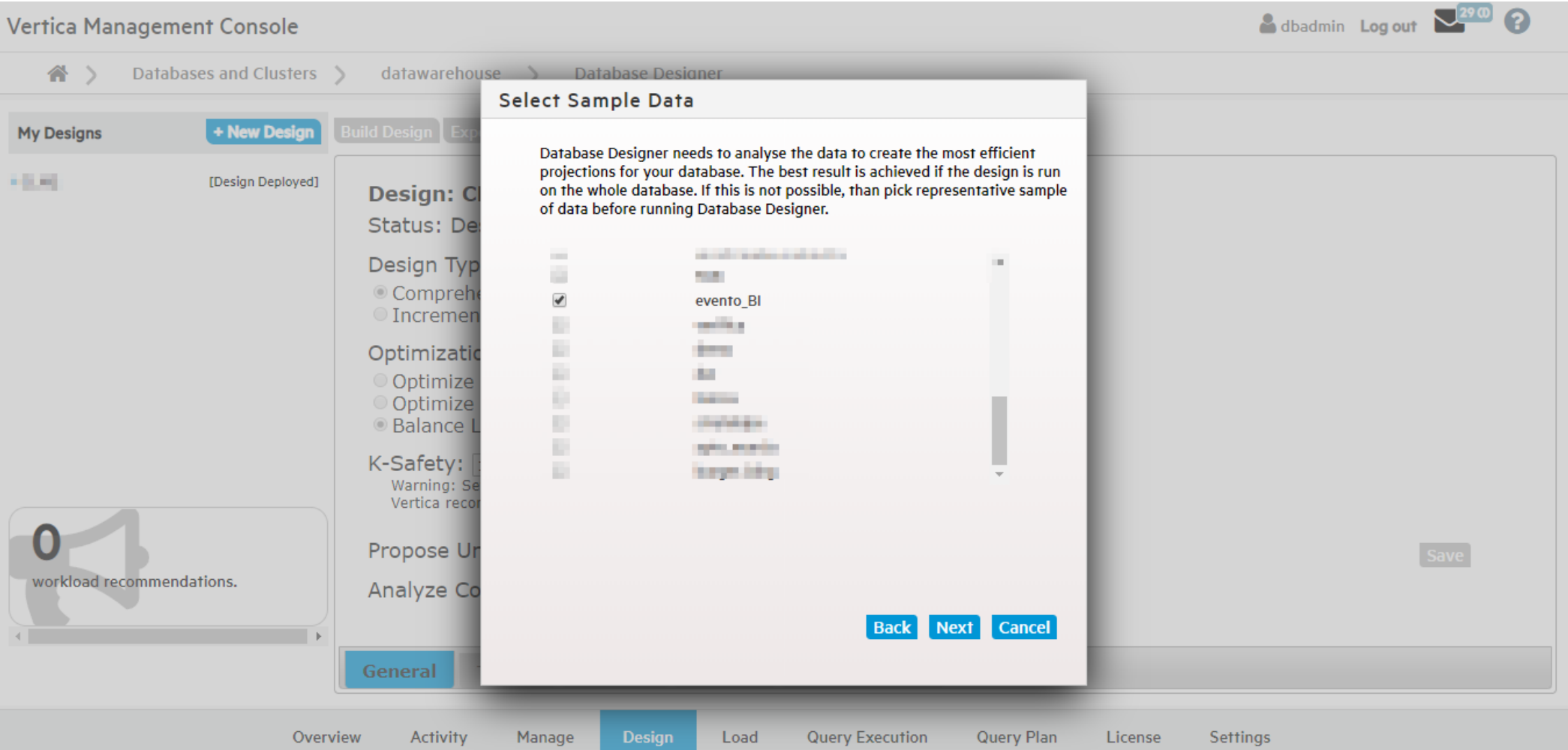
Management Console



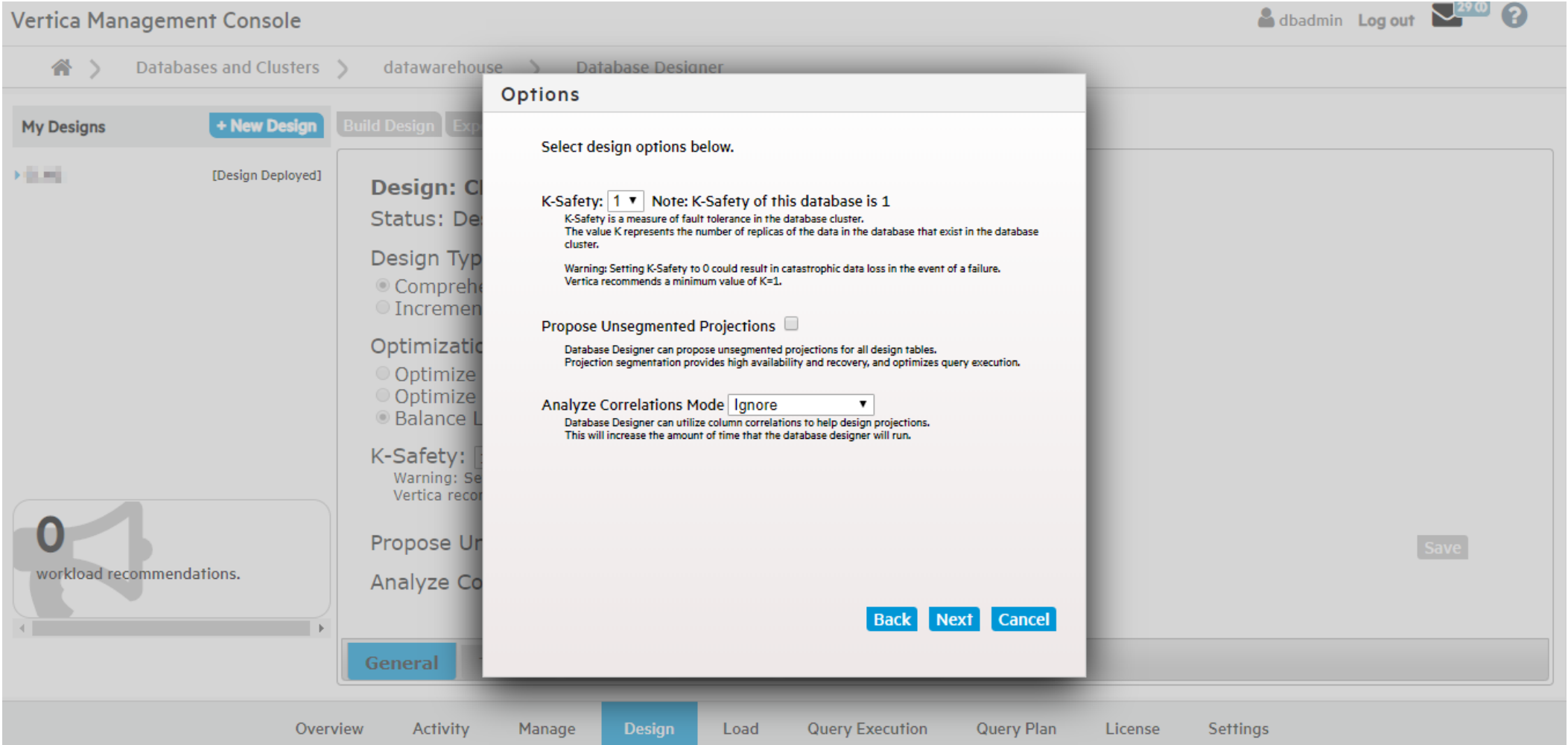
Management Console



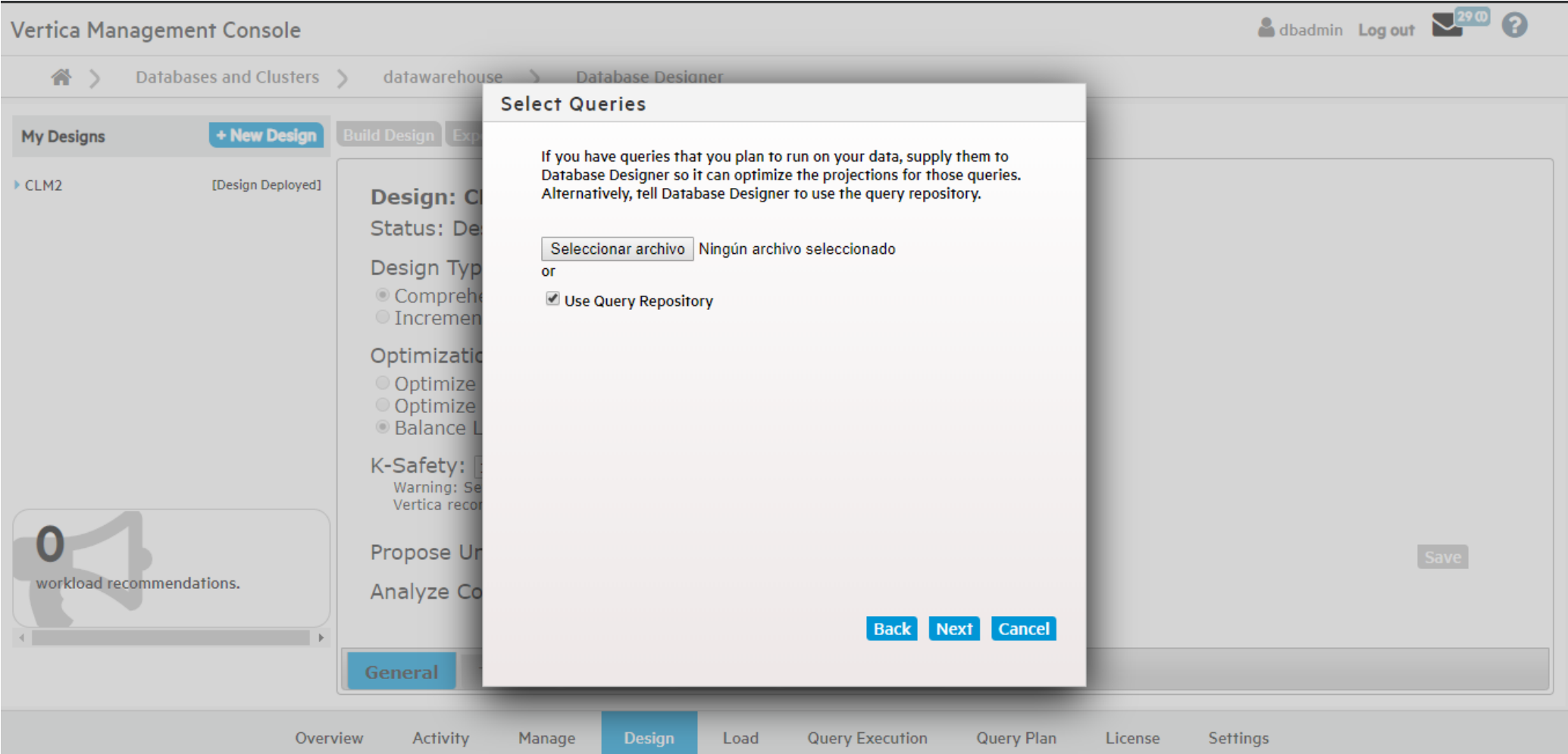
Management Console



Management Console



Management Console



Management Console

The screenshot displays the Vertica Management Console interface. At the top, the breadcrumb navigation shows 'Databases and Clusters' > 'datawarehouse' > 'Database Designer'. The 'My Designs' section on the left indicates '[Design Deployed]'. The main panel shows design configuration options: 'Design: C...', 'Status: De...', 'Design Type' with radio buttons for 'Compreh...' and 'Incremen...', 'Optimization' with radio buttons for 'Optimize...', 'Optimize...', and 'Balance L...', and 'K-Safety: [...' with a warning message. A 'Propose Un...' and 'Analyze Co...' section is partially visible. A 'General' tab is selected at the bottom. An 'Execution Options' dialog box is open in the center, titled 'Select from the preferences below.' It contains three checked options: 'Analyze Statistics' (with a description about optimal decisions), 'Auto-build' (with a description about running the designer), and 'Auto-deploy' (with a description about automatic deployment). The dialog has 'Back', 'Next', and 'Cancel' buttons at the bottom. In the background, a 'Save' button is visible on the right side of the main panel.

Vertica Management Console

dbadmin Log out 29 0 ?

Databases and Clusters > datawarehouse > Database Designer

My Designs + New Design Build Design Exp

[Design Deployed]

Design: C
Status: De
Design Type
• Compreh
• Incremen
Optimization
• Optimize
• Optimize
• Balance L
K-Safety: [
Warning: Se
Vertica recor
Propose Un
Analyze Co

General

0
workload recommendations.

Execution Options

Select from the preferences below.

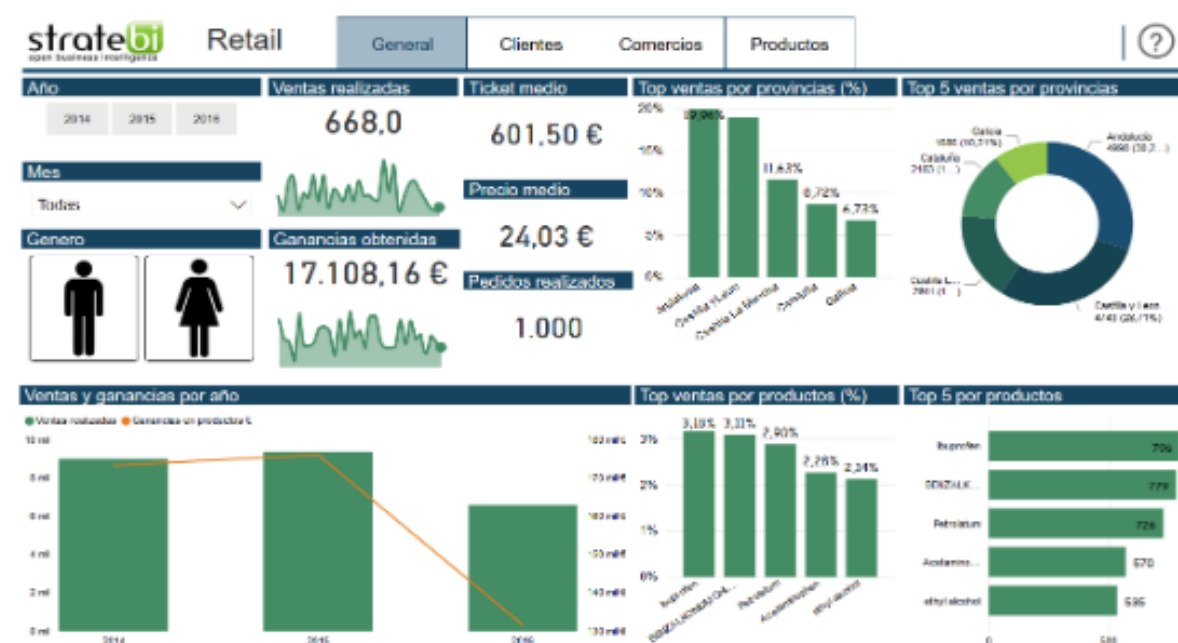
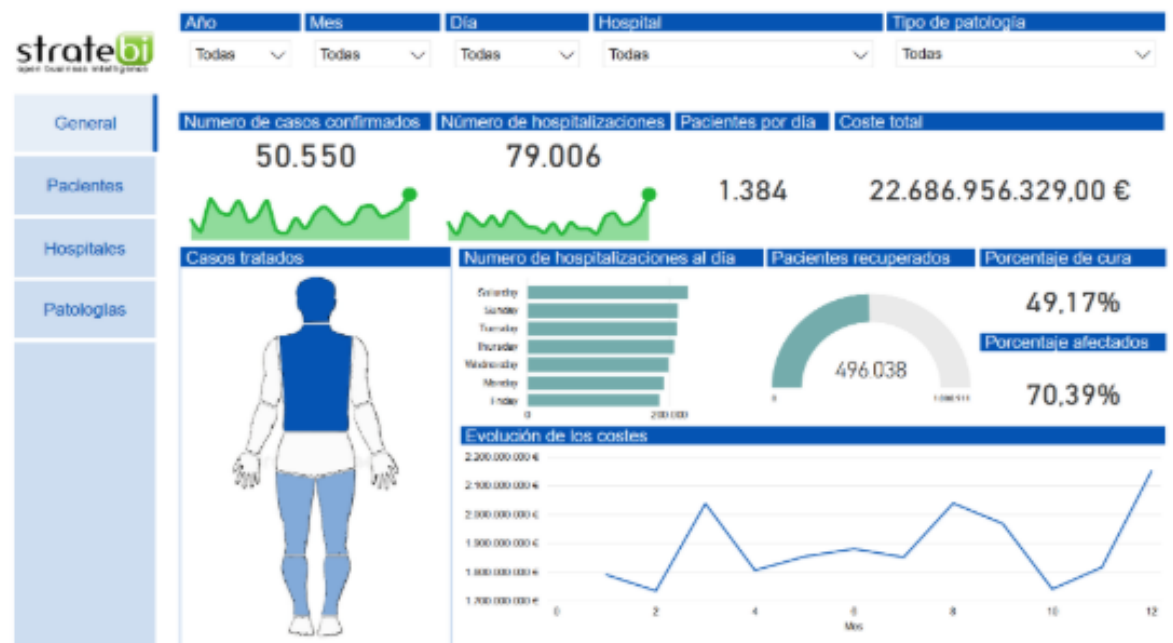
- ☒ Analyze Statistics
Analyzing statistics will help the database designer make more optimal decisions for its proposed design. You do not need to check this box if your table statistics are up to date.
- ☒ Auto-build
Select this preference to have the designer run as soon as you complete the wizard. This will only build a proposed design.
- ☒ Auto-deploy
If you've automatically built the design, you can have it automatically deployed by selecting this option.

Back Next Cancel

Save

Overview Activity Manage Design Load Query Execution Query Plan License Settings

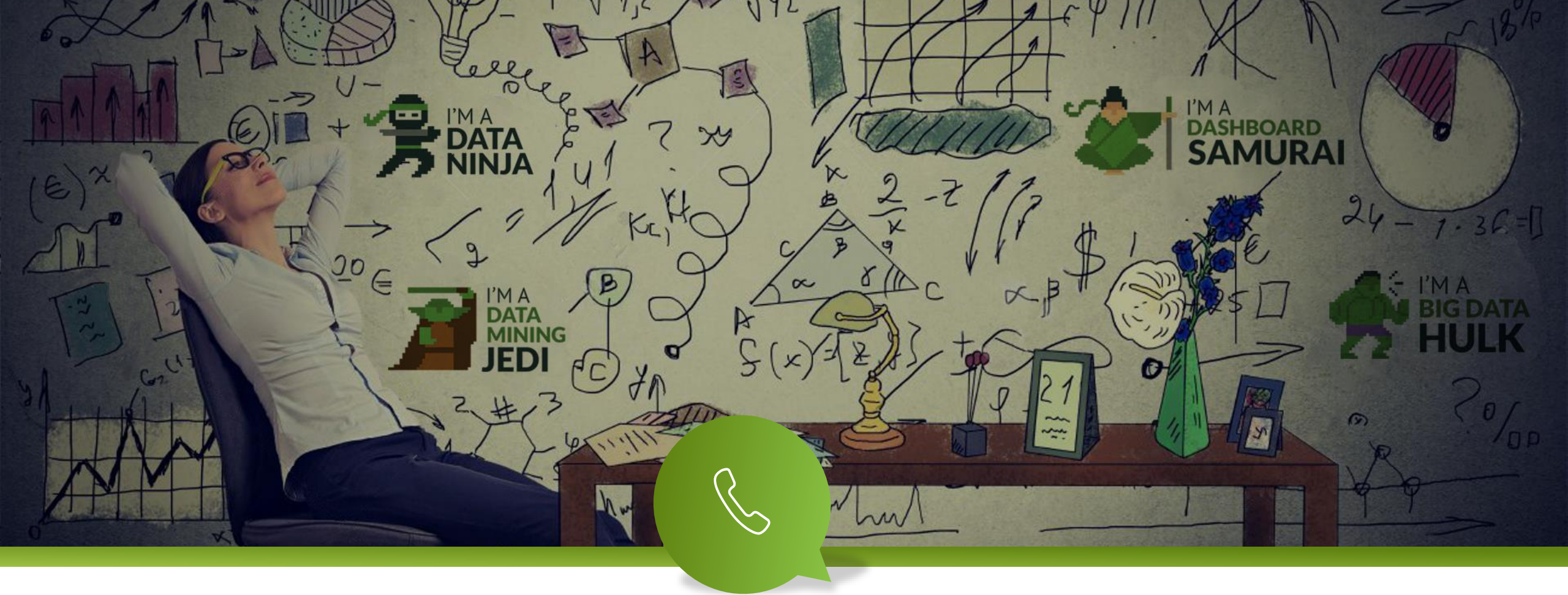
POWER BI - DIRECT QUERY



Clique aquí

para empezar con

VERTICA + stratebi



DATOS DE CONTACTO:

Madrid: Avda. del Brasil, 17

Barcelona: C/ Valencia, 63

Alicante: C/Italia, 23

stratebi
open business intelligence



91 788 34 10



www.stratebi.com



info@stratebi.com



Facebook.com/stratebiopenbi



[@stratebi](https://twitter.com/stratebi)



plus.google.com/stratebi

 **TodoBI**
< Business Intelligence >