**BODEGAS DE DATOS**

**PROYECTO DE CURSO**

Ingrith Montoya Bocanegra

Yenifer Coronado Cardona

Walter Lopez

Hector Bohorquez

Especialización Desarrollo de Bases de Datos

Facultad de Ciencias Naturales e Ingeniería

Fundación Universitaria de Bogotá Jorge Tadeo Lozano

751114 Bodegas de Datos

Jaime Alberto Chavarriaga Lozano

7 de octubre de 2023



**Proyecto de Curso 1: Revisión de Alcance**

El proyecto del curso busca desarrollar una bodega de datos para AdventureWorks, una multinacional dedicada a la venta de bicicletas.

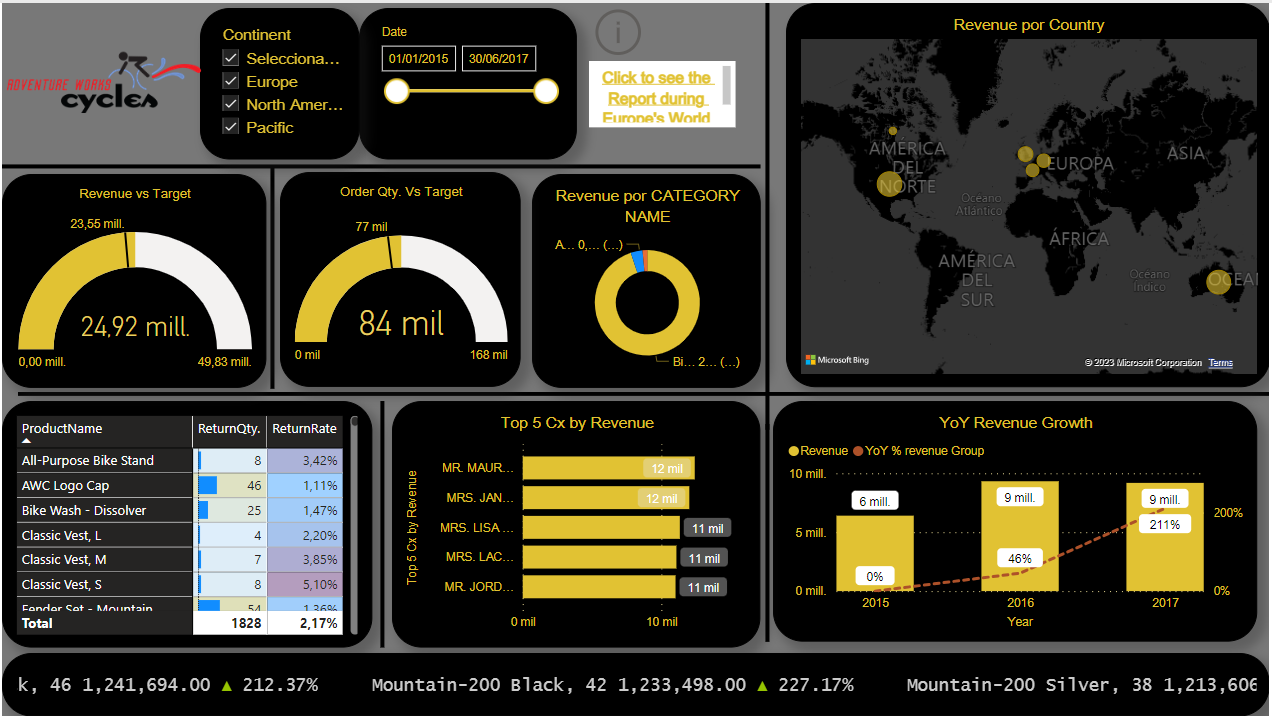
**Descripción:** Esta primera actividad nos permite de una manera exploratoria ver y analizar los distintos tablero que se pueden encontrar en la red, dándonos herramientas visuales para poder interpretar tableros e identificar cuales son los hechos y cuáles con las dimensionse.

Revisión de tableros y reportes relacionado con AdventureWorks

● Revise (al menos) tres tableros o reportes similares, disponibles en Internet, sobre consultas y análisis han hecho otras personas sobre AdventureWorks. Revise estos proyectos similares y determine las principales variables y dimensiones que se revisan en cada uno.

Tableros revisados y descripciones generales

Tablero 1: Aventure Work Cyclea



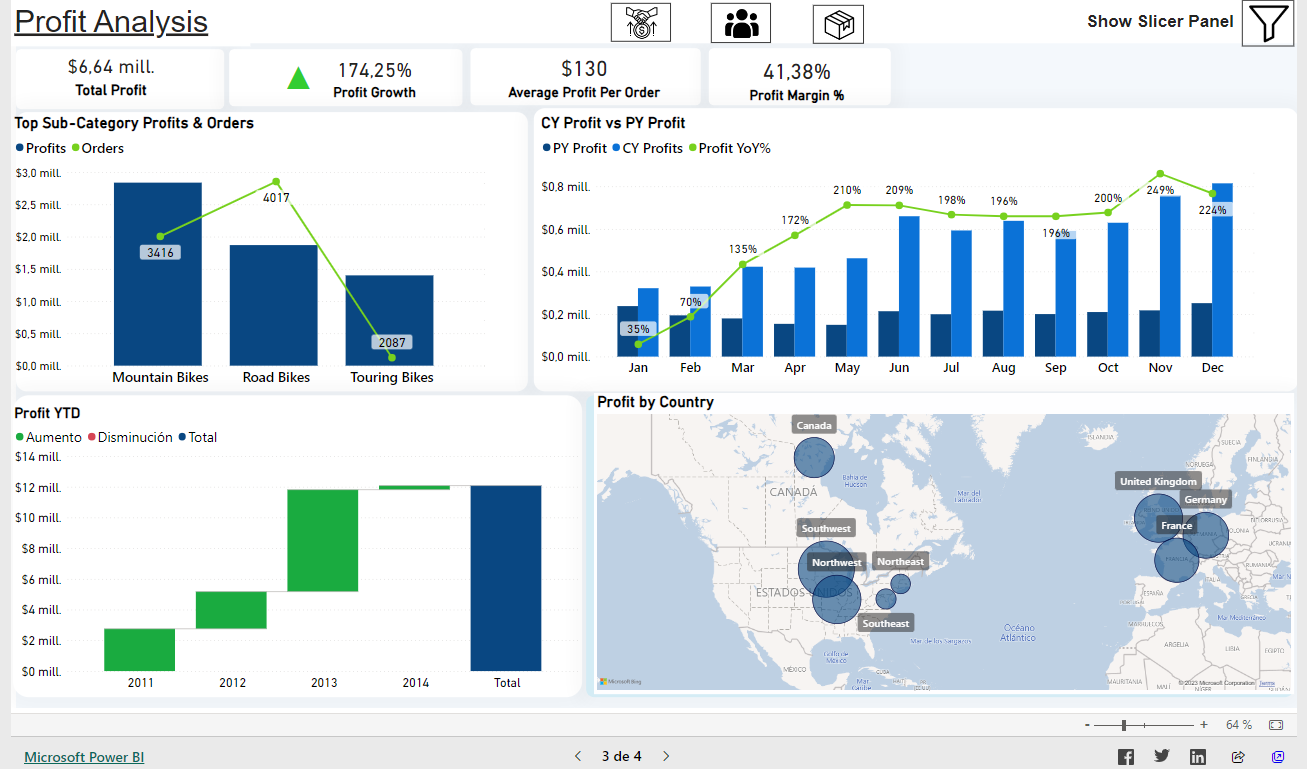
Principales Variables y Dimensiones:

| **Sitios web, documentos, reportes revisados** | |
| --- | --- |
| **Nombre, URL** | **Descripción de principales características** |
| <https://community.fabric.microsoft.com/t5/Data-Stories-Gallery/Adventure-Work-Cycles/m-p/2520057> | Visor de 9 paneles que muestra las ventas en distintas dimensiones, en el panel principal se tiene dos filtros uno por continente y otro por fecha desde 2015 a 2017. Que permiten visualizar las ventas reales vs objetivos, la cantidad de órdenes de compra reales vs las esperadas, los ingresos por categoría, el top 5 de clientes que más compran, también muestra tabla con los productos por cantidad y rentabilidad. |

**Listado inicial de hechos y dimensiones:**

|  | **Descripción** |
| --- | --- |
| **Variables / Hechos: Las ventas, como hecho y transacción que muestra la relación con las distintas dimensiones como se describe a continuación, permitiendo ver su comportamiento a través del tiempo, los continentes, los productos, los clientes y las categorías.** | |
| Criterio de Análisis / Dimensión | **Ventas por orden y sub-categoria / Producto** |
| Criterio de Análisis / Dimensión | **Ventas por país / Territorio** |
| Criterio de Análisis / Dimensión | **Top 5 de sus principales clientes / Cliente** |
| Criterio de Análisis / Dimensión | **Ventas por categoría de producto Bicicletas, accesorios, otros / Producto** |

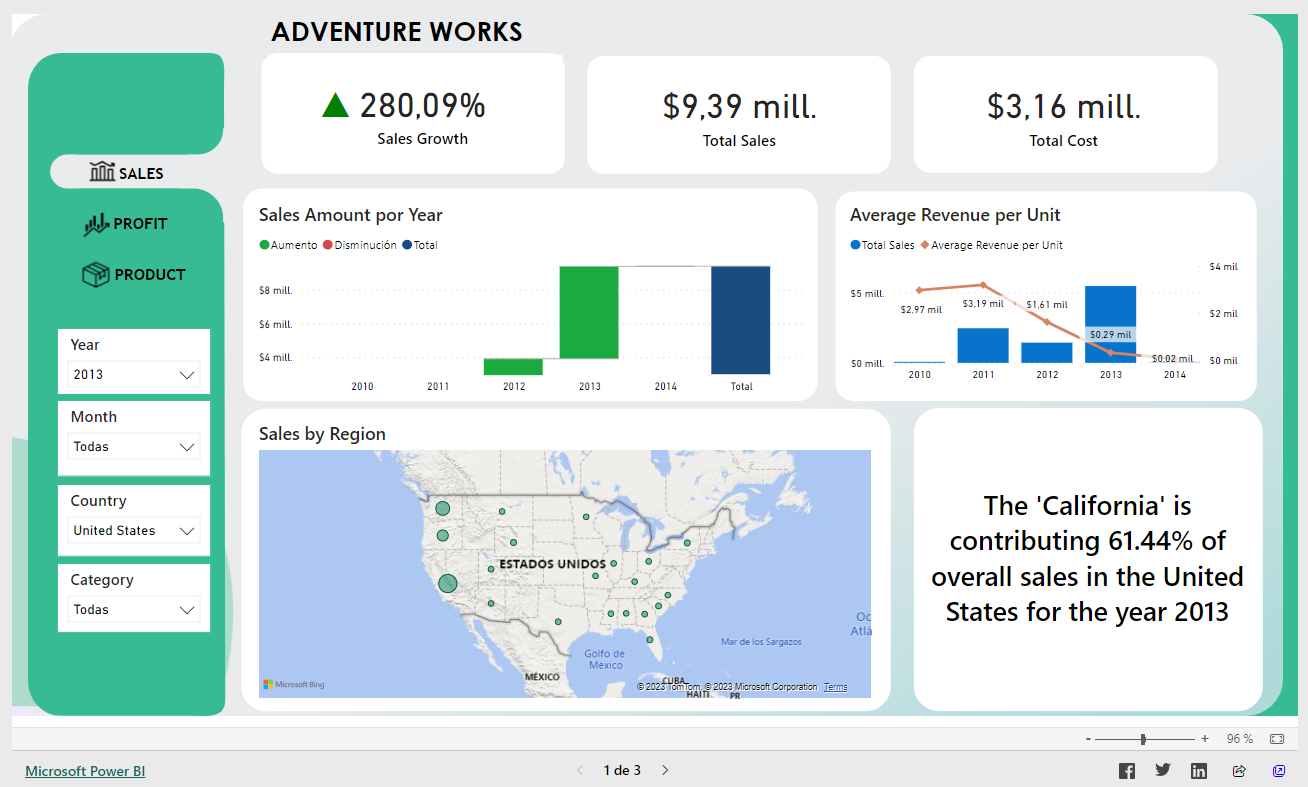
Tablero 2: Adventure Works Revenue & Profit Analysis



| **Sitios web, documentos, reportes revisados** | |
| --- | --- |
| **Nombre, URL** | **Descripción de principales características** |
| <https://community.fabric.microsoft.com/t5/Data-Stories-Gallery/Adventure-Works-Revenue-amp-Profit-Analysis/m-p/2541797> | Este tablero muestra las ventas de bicicletas desde perspectiva de los ingresos y los beneficios obtenidos, está compuesto por 4 tableros, en el tablero principal se pueden observar las ganancias totales, el % de crecimiento de las utilidades, las ganancias promedio por orden, ganancias por cantidad de ordenes de sub-categoria, el crecimiento porcentual de las utilidades del mes actual vs el mismo mes del año anterior. |

|  | **Descripción** |
| --- | --- |
| **Variables / Hechos: Las utilidades y las ventas del negocio de bicicletas vistas desde distintos angulos como, el total de ganancias, el incremento porcentual por mes actual vs mismo mes del año anterior, por total de ordenes por sup-categoria, utilidades por pais, discriminación del comportamiento de utilidades por año, identificando aumentos en color verde y disminución en color naranja, junto con el total de utilidades en color azul. Las ventas por sub-categoria** | |
| Criterio de Análisis / Dimensión | **Utilidad por Sub-categoría y órdenes / Producto** |
| Criterio de Análisis / Dimensión | **Utilidad por país / Territorio** |
| Criterio de Análisis / Dimensión | **Ventas por sub-categoría y país / Producto y Territorio** |
| Criterio de Análisis / Dimensión | **Ventas por rango de edad de clientes / Clientes** |
| Criterio de Análisis / Dimensión | **Ventas anuales según la ocupación del cliente / Clientes y Tiempo** |

Tablero 3: Adventure Works



| **Sitios web, documentos, reportes revisados** | |
| --- | --- |
| **Nombre, URL** | **Descripción de principales características** |
| <https://community.fabric.microsoft.com/t5/Data-Stories-Gallery/Sales-Analysis-for-AdventureWorks/td-p/2549649> | Este tablero muestra las ventas de bicicletas desde diferentes dimensiones, cuenta con 4 paneles en donde el panel principal permite analizar los datos desde tres filtros, las ventas, las ganancias y los productos, a través del tiempo discriminados en año y mes, los países y las categorías. Mostrando el % total de crecimiento de las ventas, las ventas totales y el costo total. |

|  | **Descripción** |
| --- | --- |
| **Variables / Hechos: las ventas del negocio de bicicletas vistas desde distintos ángulos como las ventas totales, su porcentaje de incremento, las regiones donde se concentran las mayores ventas, también permite analizar las ganancias y los productos a través del tiempo, los paise y las sub-categorías.** | |
| Criterio de Análisis / Dimensión | **Ventas por región / Territorio** |
| Criterio de Análisis / Dimensión | **Beneficio bruto y proyección de beneficios por año / Tiempo** |
| Criterio de Análisis / Dimensión | **Total de productos vendidos por año, mes, país y categoría / Tiempo, Territorio y Producto** |
| Criterio de Análisis / Dimensión | **Productos comprados por ocupación / Clientes** |
| Criterio de Análisis / Dimensión | **Top 5 de productos vendidos / Productos** |

# Proyecto de Curso 2: Revisión de la Base de Datos

**Revisión de tablas relacionadas con los tableros y reportes**

● A partir de los tableros y reportes revisados, y de los hechos y dimensiones revisados en la hoja de trabajo anterior, determine cuáles son las tablas que están relacionadas con cada uno.

**Listado de Tablas relacionadas**

|  | **Descripción** |
| --- | --- |
| **Variables / Hechos: Primer tablero revisado: Las ventas, como hecho y transacción que muestra la relación con las distintas dimensiones como se describe a continuación, permitiendo ver su comportamiento de las ventas a través del tiempo, los continentes, los productos, los clientes y las categorías.** | |
| Criterio de Análisis / Dimensión / Tablas | **Revenue Vs Target / Product-Transaction History- Sales Order Detail- Sales Order Header** |
| Criterio de Análisis / Dimensión / Tablas | **Revenue por Country / Sales Territory-Sales Order Detail- Sales Order Header** |
| Criterio de Análisis / Dimensión / Tablas | **Top 5 Cx by Revenue/ Customer- Sales Order Detail- Sales Order Header** |
| Criterio de Análisis / Dimensión / Tablas | **Total Sales / Sales Order Detail- Sales Order Header** |
| Criterio de Análisis / Dimensión / Tablas | **Total Sales por Category / Product- Product Category-Sales Order Detail- Sales Order Header** |

**Creación de consultas**

● Considerando cada una de las variables y criterios de análisis, realice una serie de consultas que le permitan extraer los datos relacionados. Estas consultas se usarán más adelante para construir la bodega de datos.

● Determine cuáles variables y dimensiones que se querían revisar inicialmente no se pueden determinar a través de las consultas

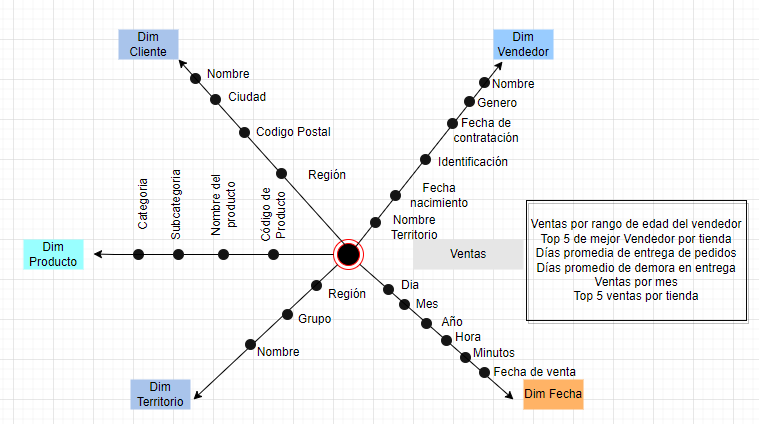
**Listado de Consultas revisadas**

| **Consulta, URL, nombre** | **Descripción** |
| --- | --- |
| Total de Ventas | **SELECT FORMAT(SUM(LineTotal),'#,0.00') AS TotalSales FROM Sales\_SalesOrderDetail ;** |
| Total Órdenes de Venta | **SELECT FORMAT((SUM(OrderQty)),'#,0.00') AS TotalOrderQuantity FROM Sales\_SalesOrderDetail;** |
| Total de Ventas por producto Categoría | **SELECT t2.ProductCategoryID, t2.Name,**  **FORMAT((SUM(LineTotal)),'#,0.00') AS TotalAmount**  **FROM**  **(SELECT c.ProductCategoryID, c.Name, t1.ProductSubcategoryID,**  **t1.Name AS SubName, t1.ProductID**  **FROM Production.ProductCategory AS c**  **JOIN**  **(SELECT s.ProductCategoryID, s.ProductSubcategoryID,**  **p.ProductID, p.Name**  **FROM Production.ProductSubcategory AS s**  **JOIN Production.Product AS p**  **ON s.ProductSubcategoryID = p.ProductSubcategoryID) AS t1**  **ON c.ProductCategoryID = t1.ProductCategoryID) AS t2**  **JOIN Sales.SalesOrderDetail AS s**  **ON t2.ProductID = s.ProductID**  **GROUP BY t2.ProductCategoryID, t2.Name**  **ORDER BY t2.ProductCategoryID;** |
| Total de Ventas por Subcategoría | **SELECT t2.Name, FORMAT((SUM(s.LineTotal)),'#,0.00') AS**  **SubCategoryTotalSales**  **FROM (SELECT t1.ProductSubcategoryID,**  **t1.ProductCategoryID, t1.Name, ProductID**  **FROM (SELECT ProductSubcategoryID, ProductCategoryID, Name**  **FROM Production\_ProductSubcategory**  **WHERE ProductCategoryID = (SELECT ProductCategoryID**  **FROM Production\_ProductCategory**  **WHERE Name = 'Bikes')) AS t1**  **LEFT JOIN Production\_Product AS p**  **ON t1.ProductSubcategoryID = p.ProductSubcategoryID) AS t2**  **LEFT JOIN Sales\_SalesOrderDetail AS s**  **ON t2.ProductID = s.ProductID**  **GROUP BY t2.Name;** |

**Consideraciones (datos difíciles de obtener, datos que no se pueden obtener, información adicional)**

|  | **Descripción** |
| --- | --- |
| Dato, Consulta, … |  |
| : |  |

# Proyecto de Curso 3: Modelo Dimensional



**Elabore un modelo dimensional para el proyecto**

● Considerando las variables que se desean incluir en los tableros y reportes, y las consultas realizadas en la base de datos, elabore un modelo dimensional para el proyecto.

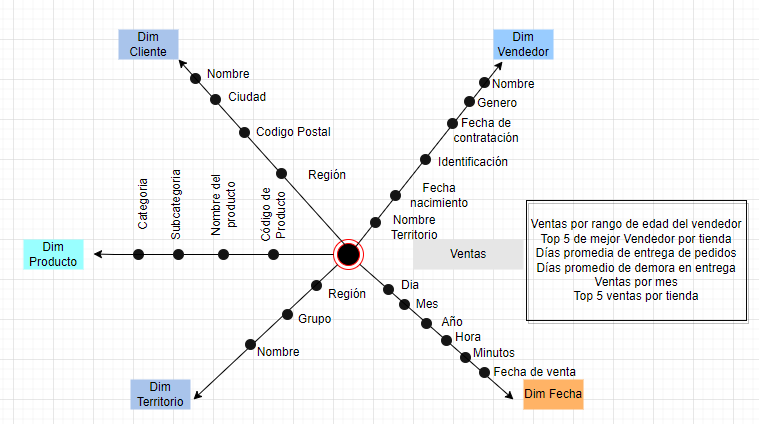
● Elabore una estrella para cada una de las variables (hechos) a analizar. Cree un modelo para cada uno de estas variables, colocando los hechos en el centro y creando líneas y puntos para cada una de las dimensiones y niveles de análisis.

● Considere los datos que sí pudo obtener a través de las consultas. No incluya atributos o dimensiones que no pudo obtener en las consultas

# Proyecto de Curso 4: Diseño de la Bodega de Datos

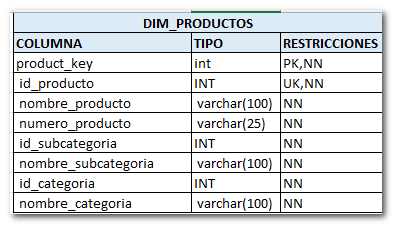
**Elabore un diseño para la Bodega de Datos**

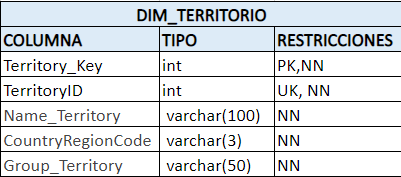
● Considerando los modelos dimensionales planteados en las hojas de trabajo anteriores, *seleccione un solo modelo dimensional*.

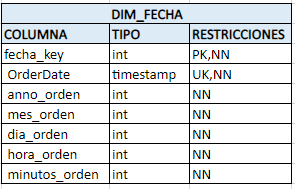


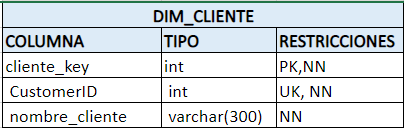
● Elabore un *diseño de la base de datos usando un esquema en estrella*. Plantee una tabla para los hechos y una tabla para cada una de las dimensiones.

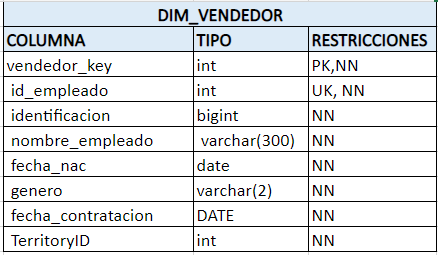
● Defina los nombres de las tablas, y los nombres y tipos de datos de cada una de las columnas de esas tablas.

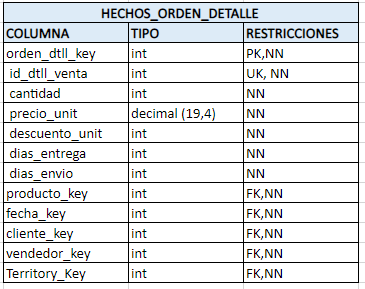




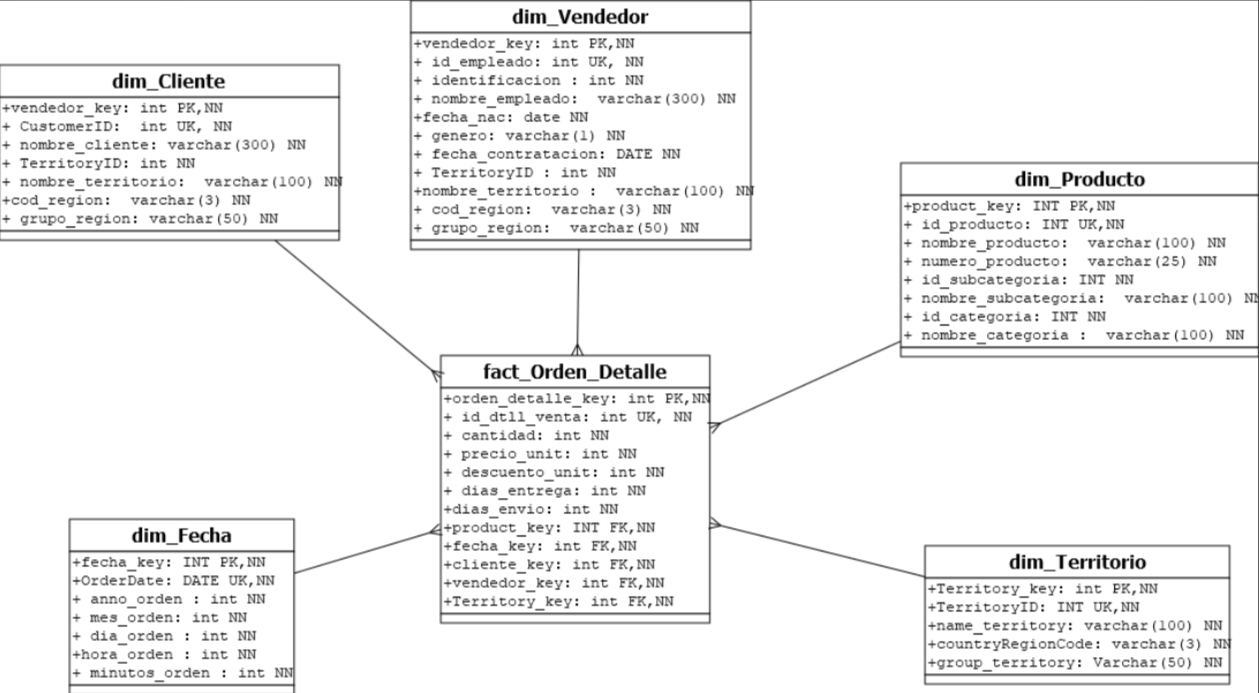








● Construya un modelo entidad relación y/o una especificación de las tablas.



# Proyecto de Curso 5: Codificación de la Bodega de Datos

Codifique la Bodega de Datos

● Usando el diseño de la base de datos usando un esquema en estrella, elaborado en las hojas de trabajo anteriores, escriba un script SQL para la creación de la base de datos.

● Cree un archivo .sql para la creación de la bodega de datos.

Create database DWHADVENTURE;

● Incluya en el archivo el código de la creación de tablas, inicie primero con las dimensiones y coloque el código de creación de las tablas de hecho al final.

-- DIMENSION PRODUCTOS -- TABLA

CREATE TABLE IF NOT EXISTS dim\_producto(

producto\_key INT NOT NULL AUTO\_INCREMENT,

id\_producto INT NOT NULL,

nombre\_producto VARCHAR(100) NOT NULL,

numero\_producto VARCHAR(25) NOT NULL,

id\_subcategoria INT NOT NULL,

nombre\_subcategoria VARCHAR(100) NOT NULL,

id\_categoria INT NOT NULL,

nombre\_categoria VARCHAR(100) NOT NULL,

PRIMARY KEY (producto\_key),

UNIQUE KEY (id\_producto)

);

-- DIMENSION TIEMPO -- TABLA

CREATE TABLE IF NOT EXISTS dim\_fecha(

fecha\_key INT NOT NULL AUTO\_INCREMENT,

orderdate TIMESTAMP NOT NULL,

anno\_orden INT NOT NULL,

mes\_orden INT NOT NULL,

dia\_orden INT NOT NULL,

hora\_orden INT NOT NULL,

minutos\_orden INT NOT NULL,

PRIMARY KEY (fecha\_key),

UNIQUE KEY (orderdate)

);

-- DIMENSION CLIENTE -- TABLA

CREATE TABLE IF NOT EXISTS dim\_cliente(

cliente\_key INT NOT NULL AUTO\_INCREMENT,

customer\_id INT NOT NULL,

nombre\_cliente VARCHAR(300) NOT NULL,

territory\_id INT NOT NULL,

PRIMARY KEY (cliente\_key),

UNIQUE KEY (customer\_id)

);

-- DIMENSION VENDEDOR -- TABLA

CREATE TABLE IF NOT EXISTS dim\_vendedor(

vendedor\_key INT NOT NULL AUTO\_INCREMENT,

id\_empleado INT NOT NULL,

identificacion BIGINT NOT NULL,

nombre\_empleado VARCHAR(300) NOT NULL,

fecha\_nac DATE NOT NULL,

genero VARCHAR(2) NOT NULL,

fecha\_contratacion DATE NOT NULL,

territory\_id INT NOT NULL,

PRIMARY KEY (vendedor\_key),

UNIQUE KEY (id\_empleado)

);

-- DIMENSION TERRITORIO -- TABLA

CREATE TABLE IF NOT EXISTS dim\_territorio(

territory\_key INT NOT NULL AUTO\_INCREMENT,

id\_territory INT NOT NULL,

name\_territory VARCHAR(100) NOT NULL,

CountryRegionCode VARCHAR(3),

group\_territory VARCHAR(50) NOT NULL,

PRIMARY KEY (territory\_key),

UNIQUE KEY (id\_territory)

);

-- DIMENSION HECHOS\_DETALLES -- TABLA

CREATE TABLE IF NOT EXISTS orden\_detalle(

orden\_dtll\_key INT NOT NULL AUTO\_INCREMENT,

id\_dtll\_venta INT NOT NULL,

cantidad INT NOT NULL,

precio\_unit INT NOT NULL,

descuento\_unit INT NOT NULL,

dias\_entrega INT NOT NULL,

dias\_envio INT NOT NULL,

producto\_key INT NOT NULL,

fecha\_key INT NOT NULL,

cliente\_key INT NOT NULL,

vendedor\_key INT NOT NULL,

territory\_key INT NOT NULL,

PRIMARY KEY (orden\_dtll\_key),

UNIQUE KEY (id\_dtll\_venta),

FOREIGN KEY (producto\_key) REFERENCES dim\_producto(producto\_key),

FOREIGN KEY (fecha\_key) REFERENCES dim\_fecha(fecha\_key),

FOREIGN KEY (cliente\_key) REFERENCES dim\_cliente(cliente\_key),

FOREIGN KEY (vendedor\_key) REFERENCES dim\_vendedor(vendedor\_key),

FOREIGN KEY (territory\_key) REFERENCES dim\_territorio(territory\_key)

);

● Incluya el código en un repositorio Github.

# Proyecto de Curso 6: ETL para las dimensiones

**Codifique el ETL para las dimensión tiempo**

● Cree primero una consulta sobre la base de datos transaccional que le permita obtener todas las fechas relacionadas con las transacciones o eventos que son de interés en la bodega de datos. Incluya el nombre de la base de datos con el nombre de la tabla.

**Ejecución:**

**SELECT distinct**

**sh.OrderDate orderdate,**

**year(sh.OrderDate) as anno\_orden,**

**month(sh.OrderDate) as mes\_orden,**

**dayofweek(sh.OrderDate) as dia\_orden,**

**hour(sh.OrderDate)as hora\_orden,**

**minute(sh.OrderDate) as minutos\_orden**

**FROM Sales\_SalesOrderHeader as sh;**

● Modifique la consulta para obtener la información (por ejemplo, mes, año) que se desea almacenar en las dimensiones. Revise el diseño de la dimensión y determine cuales son las funciones de SQL que les permite obtener esta información. Use la consulta anterior como un subquery o un CTE.

**Ejecución:**

WITH fechas as (

SELECT DISTINCT sh.OrderDate AS FECHA

FROM Sales\_SalesOrderHeader as sh

), FECHA\_VENTA AS (

SELECT

F.FECHA orderdate,

year(F.FECHA) as anno\_orden,

month(F.FECHA) as mes\_orden,

dayofweek(F.FECHA) as dia\_orden,

hour(F.FECHA)as hora\_orden,

minute(F.FECHA) as minutos\_orden

FROM fechas as F)

SELECT \* FROM FECHA\_VENTA

limit 5;

● Modifique la consulta para insertar los datos en la dimensión tiempo. Incluya el nombre de la base de datos en el nombre de la tabla

Ejecución:

INSERT INTO DWHADVENTURE.dim\_fecha(

orderdate, anno\_orden, mes\_orden, dia\_orden, hora\_orden, minutos\_orden

)

WITH fechas as (

SELECT DISTINCT sh.OrderDate AS FECHA

FROM Sales\_SalesOrderHeader as sh

), FECHA\_VENTA AS (

SELECT

F.FECHA orderdate,

year(F.FECHA) as anno\_orden,

month(F.FECHA) as mes\_orden,

dayofweek(F.FECHA) as dia\_orden,

hour(F.FECHA)as hora\_orden,

minute(F.FECHA) as minutos\_orden

FROM fechas as F)

SELECT \* FROM FECHA\_VENTA;

● Incluya el código en el repositorio Github.

**Codifique el ETL para las otras dimensiones**

● Cree una consulta sobre la base de datos transaccional que le permita obtener los datos de las entidades que corresponden en las otras dimensiones de la bodega de datos. Identifique cuáles son las tablas que corresponden con cada entidad y determine qué columnas corresponden con los datos de las dimensiones.

**Dimensión Cliente**

WITH CLIENTE AS (

SELECT comp.CustomerID,

per.FirstName,

per.MiddleName,

per.LastName,

comp.TerritoryID

FROM Sales\_Customer as comp

join Person\_Person as per on per.BusinessEntityID = comp.PersonID

), cliente\_adw as(

select c.CustomerID ,

concat (c.FirstName, ' ', c.LastName) as nombre\_cliente,

c.TerritoryID

from CLIENTE as c

)

SELECT \* FROM cliente\_adw ;

INSERT INTO DWHADVENTURE.dim\_cliente ( customer\_id ,nombre\_cliente, territory\_id

)

WITH CLIENTE AS (

SELECT comp.CustomerID,

per.FirstName,

per.MiddleName,

per.LastName,

comp.TerritoryID

FROM Sales\_Customer as comp

join Person\_Person as per on per.BusinessEntityID = comp.PersonID

), cliente\_adw as(

select c.CustomerID ,

concat (c.FirstName, ' ', c.LastName) as nombre\_cliente,

c.TerritoryID

from CLIENTE as c

)

SELECT \* FROM cliente\_adw ;

**Dimensión Producto**

SELECT pp.ProductID id\_producto,

pp.Name nombre\_producto,

pp.ProductNumber numero\_producto,

sbc.ProductSubcategoryID id\_subcategoria,

sbc.Name nombre\_subcategoria,

pc.ProductCategoryID id\_categoria,

pc.Name nombre\_categoria

FROM Production\_Product as pp

join Production\_ProductSubcategory as sbc on sbc.ProductSubcategoryID = pp.productsubcategoryid

join Production\_ProductCategory as pc on pc.ProductCategoryID = sbc.ProductCategoryID ;

INSERT INTO DWHADVENTURE.dim\_producto (id\_producto, nombre\_producto, numero\_producto , id\_subcategoria , nombre\_subcategoria, id\_categoria , nombre\_categoria )

SELECT pp.ProductID id\_producto,

pp.Name nombre\_producto,

pp.ProductNumber numero\_producto,

sbc.ProductSubcategoryID id\_subcategoria,

sbc.Name nombre\_subcategoria,

pc.ProductCategoryID id\_categoria,

pc.Name nombre\_categoria

FROM Production\_Product as pp

join Production\_ProductSubcategory as sbc on sbc.ProductSubcategoryID = pp.productsubcategoryid

join Production\_ProductCategory as pc on pc.ProductCategoryID = sbc.ProductCategoryID ;

**Dimensión Territorio**

SELECT terr.TerritoryID,

terr.Name,

terr.CountryRegionCode,

terr.Group

FROM Sales\_SalesTerritory as terr;

INSERT INTO DWHADVENTURE.dim\_territorio(id\_territory, name\_territory, CountryRegionCode, group\_territory )

SELECT terr.TerritoryID,

terr.Name,

terr.CountryRegionCode,

terr.Group

FROM Sales\_SalesTerritory as terr;

**Dimensión Vendedor**

WITH vendedor AS(

select

emp.BusinessEntityID id\_empleado,

emp.NationalIDNumber identificacion,

per.FirstName,

per.LastName,

emp.BirthDate fecha\_nac,

emp.Gender genero,

emp.HireDate fecha\_contratacion,

IFNULL(sp.TerritoryID, 0) TerritoryID

FROM HumanResources\_Employee as emp

join Person\_Person as per on per.BusinessEntityID = emp.BusinessEntityID

JOIN Sales\_SalesPerson sp on sp.BusinessEntityID = per.BusinessEntityID

), vendedor\_adw AS(

SELECT v.id\_empleado,

v.identificacion,

CONCAT(v.FirstName, ' ',v.LastName) as nombre\_empleado,

v.fecha\_nac,

v.genero,

v.fecha\_contratacion,

v.TerritoryID

FROM vendedor as v

)

SELECT \* FROM vendedor\_adw;

INSERT INTO DWHADVENTURE.dim\_vendedor (id\_empleado,identificacion , nombre\_empleado, fecha\_nac , genero ,

fecha\_contratacion, territory\_id )

WITH vendedor AS(

select

emp.BusinessEntityID id\_empleado,

emp.NationalIDNumber identificacion,

per.FirstName,

per.LastName,

emp.BirthDate fecha\_nac,

emp.Gender genero,

emp.HireDate fecha\_contratacion,

IFNULL(sp.TerritoryID, 0) TerritoryID

FROM HumanResources\_Employee as emp

join Person\_Person as per on per.BusinessEntityID = emp.BusinessEntityID

JOIN Sales\_SalesPerson sp on sp.BusinessEntityID = per.BusinessEntityID

), vendedor\_adw AS(

SELECT v.id\_empleado,

v.identificacion,

CONCAT(v.FirstName, ' ',v.LastName) as nombre\_empleado,

v.fecha\_nac,

v.genero,

v.fecha\_contratacion,

v.TerritoryID

FROM vendedor as v

)

SELECT \* FROM vendedor\_adw;

**Dimension hechos:**

SELECT det.SalesOrderID id\_dtll\_venta,

det.OrderQty cantidad,

det.UnitPrice precio\_unit,

det.UnitPriceDiscount descuento\_unit,

oh.DueDate ,

oh.OrderDate ,

oh.ShipDate

/\*dp.producto\_key,

df.fecha\_key,

dc.cliente\_key,

dv.vendedor\_key,

dt.territory\_key \*/

FROM Sales\_SalesOrderDetail as det

JOIN Sales\_SalesOrderHeader as oh on oh.SalesOrderID = det.SalesOrderID

JOIN DWHADVENTURE.dim\_cliente as dc on dc.customer\_id = oh.CustomerID

JOIN DWHADVENTURE.dim\_vendedor as dm on dm.id\_empleado = oh.SalesPersonID

JOIN DWHADVENTURE.dim\_territorio as dt on dt.id\_territory = oh.TerritoryID

JOIN DWHADVENTURE.dim\_fecha AS df on df.orderdate = oh.OrderDate

JOIN DWHADVENTURE.dim\_producto AS dp on dp.id\_producto = det.ProductID

limit 5;

INSERT INTO DWHADVENTURE.orden\_detalle (id\_dtll\_venta, cantidad, precio\_unit, descuento\_unit, dias\_entrega, dias\_envio, producto\_key, fecha\_key, cliente\_key, vendedor\_key, territory\_key

)

with fact as(

SELECT det.SalesOrderID id\_dtll\_venta,

det.OrderQty cantidad,

det.UnitPrice precio\_unit,

det.UnitPriceDiscount descuento\_unit,

oh.DueDate ,

oh.OrderDate ,

oh.ShipDate,

dp.producto\_key as producto\_key,

df.fecha\_key as fecha\_key,

dc.cliente\_key as cliente\_key,

dm.vendedor\_key as vendedor\_key,

dt.territory\_key as territory\_key

FROM Sales\_SalesOrderDetail as det

JOIN Sales\_SalesOrderHeader as oh on oh.SalesOrderID = det.SalesOrderID

JOIN DWHADVENTURE.dim\_cliente as dc on dc.customer\_id = oh.CustomerID

JOIN DWHADVENTURE.dim\_vendedor as dm on dm.id\_empleado = oh.SalesPersonID

JOIN DWHADVENTURE.dim\_territorio as dt on dt.id\_territory = oh.TerritoryID

JOIN DWHADVENTURE.dim\_fecha AS df on df.orderdate = oh.OrderDate

JOIN DWHADVENTURE.dim\_producto AS dp on dp.id\_producto = det.ProductID

), fact\_consol as(

select f.id\_dtll\_venta,

f.cantidad,

f.precio\_unit,

f.descuento\_unit,

DATEDIFF(f.DueDate ,f.OrderDate ) as dias\_entrega,

DATEDIFF(f.ShipDate ,f.OrderDate ) as dias\_envio ,

f.producto\_key,

f.fecha\_key,

f.cliente\_key,

f.vendedor\_key,

f.territory\_key

from fact as f

)

select \* from fact\_consol dd ;

● Si se requiere, modifique la consulta para hacer cálculos y obtener la información (por ejemplo, días de demora, utilidad de la venta) que se desea almacenar en las dimensiones. Revise el diseño de la dimensión y determine cuales son las funciones de SQL que les permite obtener esta información. Use la consulta anterior como un subquery o un CTE.

o

● Modifique la consulta para insertar los datos en la dimensión.

o Por ejemplo,

INSERT INTO datawh.dim\_xxx (

...

)

SELECT ...

● Incluya el código en el repositorio Github.

<https://github.com/imontoyab/mysql_adw_ingrith>

● Repita este procedimiento para cada dimensión

# Proyecto de Curso 7: ETL para los hechos

**Codifique el ETL para las tablas de hechos**

● Cree primero una consulta sobre la base de datos transaccional que le permita obtener todos los datos de interés para almacenar en las tablas de hechos. Esta consulta normalmente es una sábana de datos (*data sheet*) o un listado general con todos los datos de las transacciones. Una todas las tablas que se requieran para obtener los id de negocio de las entidades que hacen parte de las dimensiones

**Ejecución:**

**/\*---------------producto com mayor valor vendido--------------------------\*/**

WITH VENTA AS(

select dp.id\_producto, dp.nombre\_producto, sum(od.cantidad \* od.precio\_unit) AS SUM

from orden\_detalle od

join dim\_producto as dp on dp.producto\_key = od.producto\_key

group by dp.id\_producto, dp.nombre\_producto

),MAYOR\_VALOR AS(

SELECT V.nombre\_producto, V.SUM VALOR\_VENTA

FROM VENTA V

ORDER BY SUM DESC

LIMIT 1

)

SELECT \* FROM MAYOR\_VALOR;

**/\*------------------5 PRODUCTOS MÁS VENDIDO -------------------------\*/**

WITH VENTA AS(

select dp.id\_producto, dp.nombre\_producto, sum(od.cantidad ) AS TOTAL

from orden\_detalle od

join dim\_producto as dp on dp.producto\_key = od.producto\_key

group by dp.id\_producto, dp.nombre\_producto

),MAYOR\_VALOR AS(

SELECT V.nombre\_producto, V.TOTAL

FROM VENTA V

ORDER BY V.TOTAL DESC

LIMIT 5

)

SELECT \* FROM MAYOR\_VALOR;

**/\*-------------PRODUCTO MÁS VENDIDO POR TERRITORIO------------------\*/**

WITH VENTA AS(

select dt.name\_territory , dp.nombre\_producto, sum(od.cantidad) AS CANTIDAD

from orden\_detalle od

join dim\_producto as dp on dp.producto\_key = od.producto\_key

JOIN dim\_territorio AS dt ON dt.territory\_key = od.territory\_key

group by dt.name\_territory, dp.nombre\_producto

),

prod\_terr as(

SELECT v.name\_territory , MAX(v.CANTIDAD) AS MAX\_CANT

FROM VENTA v

GROUP BY v.name\_territory

), PP AS(

SELECT pt.name\_territory, v.nombre\_producto, pt.MAX\_CANT

FROM prod\_terr as pt

join VENTA as v on v.CANTIDAD = pt.MAX\_CANT and v.name\_territory = pt.name\_territory

order by MAX\_CANT desc

)

SELECT \* FROM PP

;

● Incluya el código en el repositorio Github.

● Repita el procedimiento por cada tabla de hechos

# Proyecto de Curso 8: Consultas sobre la bodega de datos

**Elabore consultas sobre la bodega de datos**

● Revise las consultas, reportes y tableros que se seleccionaron para crear la bodega de datos.

● Seleccione dos o tres consultas y determine la consulta SQL que permite obtener estos datos

● Inicie la consulta obteniendo una sábana de datos (data sheet) uniendo la información de la tabla de hechos con las tablas de las dimensiones

o Por ejemplo,  
 SELECT …  
 FROM datawh.dim\_transaccion  
 JOIN datawh.dim\_tiempo USING (tiempo\_key)

JOIN datawh.dim\_cliente USING (cliente\_key)  
 JOIN datawh.dim\_producto USING (producto\_key)  
 JOIN datawh.dim\_almacen USING (almacen\_key)

● Modifique la consulta para obtener los datos que se necesitan en la consulta. Use funciones de grupo, subqueries y secuencias de CTEs si es necesario.

● Incluya el código en el repositorio Github.

## 

**Repositorio:** (Dirección URL del repositorio Github con el proyecto)

<https://github.com/imontoyab/mysql_adw_ingrith>

**1. Principales problemas:**

* En la tabla de vendedor no se estaba generando el cruce correcto de los datos.
* Al momento de crear las tablas se utilizó un tipo de dato distinto al que se tenía en las tablas transaccionales.
* Se identificó que se repetía el territorio en tres tablas vendedor, cliente y detalle de ventas, por lo cual se generó una dimensión adicional llamada territorio.
* Al validar los datos no se encontró para la dimensión de cliente datos como género y fecha de nacimiento.
* La tabla detalle de ventas, no guarda de forma obligatoria el código del vendedor, por lo cual se tenían campos nulos, que no se sumaban generando inconsistencias al momento de realizar el join con la dimensión de vendedores, trayendo una menor cantidad de datos al comparar con la base transaccional.

**2. Principales Aprendizajes y Conclusiones:**

Podemos generar un contenedor a partir de una plantilla, la cual podemos compartir y trabajar simultáneamente en equipo, construyendo la bodega de datos en equipo y no por separado.

Las bodegas de datos nos permiten usar la información transaccional de la base y generar datos precalculados, para el análisis de la información.

Generación de ETL y CTE, identificado que el ETL sirve para pasar la información por medio de CTE de las tablas transaccionales a la bodega de datos.

Con la información de la bodega de datos se requieren unir menos tablas ya que la información está desnormalizada.

La importancia de tener bases sólidas de lenguaje SQL para el trabajo de la bodega de datos.