

## **Bases de datos II**

08/03/2021 - 11/03/2021

Apuntes de la clase 08/03/2021 - 11/03/2021

IS-601

**Emilson Acosta**

### **Clase #4.1**

#### **Herramienta de soporte a la toma de decisiones Pentaho DSS**

Una vez creado el datamart y la base de datos OLAP se procede a cargar la información a la base de datos OLAP por medio de ETL, una vez el ETL se haya ejecutado lo que queda es realizar los reportes. Pentaho se instala en el puerto 8081. El objetivo es realizar una conexión con el gestor de base de datos SQL Server:

1. Manage Data Source
2. Nuevo Data Source
3. Asignar nombre al origen de los datos
4. Seleccionar tipo de datos del origen usando **Tablas de base de datos**
5. Agregar una nueva conexión
6. En el tipo de la base de datos seleccionar **MS SQL Server**
7. Escoger el nombre de la conexión con el prefijo cx
8. En el nombre del host **localhost**
9. En el nombre de la base de datos se indicara el nombre de la base de datos que se utilizará.
10. El puerto se deja por defecto
11. En el nombre de usuario se usará un nombre de usuario SQL Server **sa** pass: **123**
12. Probar la conexión
13. Conexión creada
14. Seleccionar reportes y análisis
15. Agregar todas las tablas al lado derecho, el diagrama "sysdiagrams" no se utilizará.
16. Indicar la tabla de hechos en Fact table.
17. Configurar como se relaciona una tabla de dimensión con una de hechos seleccionando el campo que comparten las tablas de dimensión con la de hechos.
18. Crear unión o Create Join

## Crear reportes

1. Crear nuevo
2. Reporte de análisis
3. Seleccionar el origen de datos
4. Arrastrar la métrica
5. Arrastrar campos a las filas y columnas

### Reporte de análisis #1

- a. Métrica: Total venta productos
- b. Filas: Nombre del producto
- c. Columnas: Año

The screenshot shows the Pentaho User Console interface. The main window displays an Analysis Report for the 'Total venta producto' metric, grouped by 'Año' (Year) and 'Producto' (Product Name). The report is presented in a table format with columns for the year (1996, 1997, 1998) and rows for each product. The 'Total venta producto' metric is calculated for each product across the years.

Producto	1996	1997	1998
Alice Mutton	6962	17605	813
Aniseed Syrup	240	1724	1081
Boston Crab Meat	2778	9815	5311
Camembert Pierrot	9025	20505	17291
Camrnon Tigerz	4725	15950	849
Chai	1606	4887	6291
Chang	3018	7039	6291
Chartreuse verte	3558	4476	426
Chef Anton's Cajun Seasoning	1852	5215	1501
Chef Anton's Gumbo Mix	1931	374	3041
Chocolate	-	1282	81
Côte de Blaye	24874	49198	67321
Escargots de Bourgogne	1378	2076	2421
Filo Mix	246	2124	861
Flotemysost	4248	8439	6861
Geitost	386	786	471
Genen Shouyu	310	1475	
Gnocchi di nonna Alice	2763	32604	7221
Gorgonzola Telino	4155	7301	3461

## Filtros

1. Arrastrar el campo del cual se quiere filtrar

### Reporte de análisis #2 con filtro

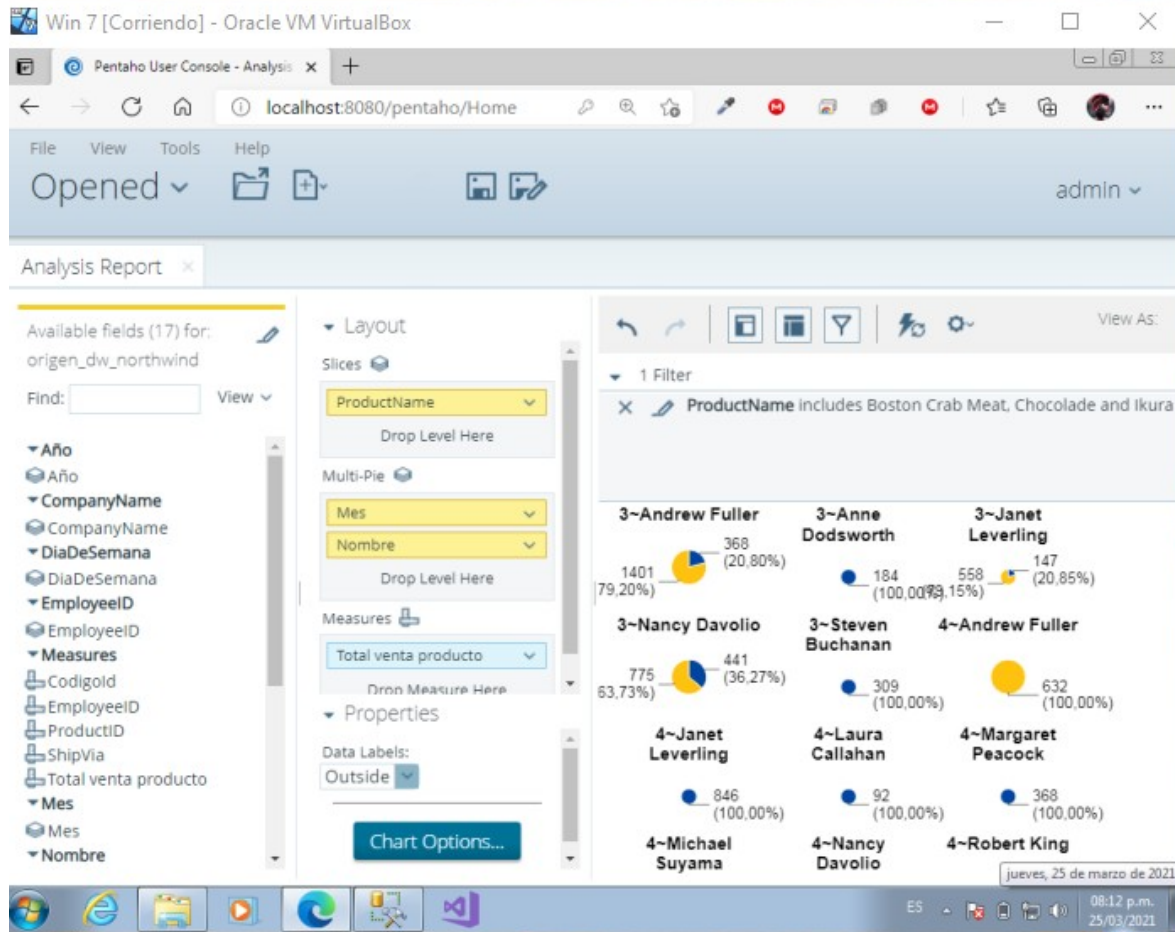
- a. Métrica: Total venta productos
- b. Filas: Nombre del producto
- c. Columnas: Año
- d. Filtro: Nombre del producto

The screenshot shows the Pentaho User Console interface. The 'Analysis Report' tab is active, displaying a table with sales data. The table is filtered by 'Product Name' (Boston Crab Meat, Chocolate, and Ikura). The columns are 'Año' (Year) and 'Total venta producto' (Total product sales). The rows are grouped by 'Product Name'.

Product Name	1996	1997	1998
Boston Crab Meat	2778	9815	5318
Chocolate	-	1282	87
Ikura	1929	9936	9002

### Reporte de análisis #3 con otra vista

- a. Métrica: Total venta productos
- b. Filas: Nombre del producto
- c. Columnas: Mes y nombre de empleado
- d. Filtro: Nombre del producto



## Guardar reporte

1. Tomar en cuenta la dirección donde se guarda
2. En la carpeta home están las carpetas de cada usuario creado por defecto
3. El reporte guardado donde Admin solo se podrá visualizar por admin

## Clase #4.2

Los reportes se pueden exportar y también guardar de forma pública, haciendo que cualquier usuario pueda visualizarlo.

**Dashboard:** Tablero de control, donde en una sola pantalla se puede visualizar distintos tipos de información

1. Click en la flecha a la par de OPENED
2. Home
3. Crear nuevo
4. Dashboard

5. Seleccionar la estructura del tablero de control
6. Arrastrar los reportes a visualizar
7. Agregar titulo

## Clase #5.1

### Procedimientos almacenados

Parte de los subprogramas de Oracle, a través del cual se pueden hacer inserciones, modificaciones, etc.

1. Usa la sintaxis básica de un bloque anónimo
2. Como los triggers se debe incluir al inicio el CREATE OR REPLACE PROCEDURE  
NombreProcedimiento
3. Los procedimientos no siempre deben recibir parámetros

```
CREATE OR REPLACE PROCEDURE <NOMBRE PROCEDIMIENTO> (<PARAMETROS>)  
  
IS  
    <DECLARACION DE VARIABLES>  
BEGIN  
    <INSTRUCCIONES SQL>  
END;
```

Se pueden utilizar las instrucciones SQL de la primer unidad como asignacion de variables, excepciones, ciclos, condiciones, etc.

### Declaración de parametros

1. Indicar el nombre del parámetro
2. Tipo de dato

```
--- Declaración de variable  
<NOMBRE PARAMETRO> IN | OUT | IN OUT <TIPO DE DATO>  
  
--- Ejemplo  
NOMBRE VARCHAR2;
```

Es importante tener en cuenta que no se debe indicar el rango de la variable del parámetro dentro de un procedimiento almacenado. **IN:** Significa que el parámetro solo puede ser usado para recibir valores y no puede ser modificado en el cuerpo del procedimiento, cuando no se

especifica si el parámetro es IN o OUT se toma como IN. **OUT:** Significa que el parámetro solo puede ser usado para ser modificado en el cuerpo del procedimiento y retornar un valor. **IN OUT:** Significa que el parámetro solo puede ser usado para recibir valores y se puede modificar en el cuerpo del procedimiento y retornarse al bloque que llamo el procedimiento. Todos los parámetros de salida pueden recibir valores null.

### Ejemplo de SP (Store Procedure) Imprimiendo valores

--- Obtener el nombre de la categoria = 2 e imprimir el valor obtenido

```
CREATE OR REPLACE PROCEDURE SP_NOMBRE_CAT(ID_CAT IN NUMBER)
IS
    NOMBRE_CAT CATEGORIES.CATEGORY_NAME%TYPE;
BEGIN
    SELECT CATEGORIES.CATEGORY_NAME INTO NOMBRE_CAT FROM CATEGORIES
    WHERE CATEGORY_ID = ID_CAT;

    DBMS_OUTPUT.PUT_LINE('El nombre de la categoria es: '||NOMBRE_CAT);
END;
```

### Ejecutar procedimiento almacenado

#### 1. EXECUTE

```
--- 1. EXECUTE
EXECUTE SP_NOMBRE_CAT(2);
```

#### 2. Bloque anónimo

```
--- 2. Bloque anonimo
BEGIN
    SP_NOMBRE_CAT(2);
END;
```

### Ejemplo de SP (Store Procedure) Retornando valores

```
CREATE OR REPLACE PROCEDURE SP_NOMBRE_CAT(ID_CAT IN NUMBER,  
NOMBRE_CAT OUT CATEGORIES.CATEGORY_NAME%TYPE)  
  
IS  
  
BEGIN  
  
    SELECT CATEGORIES.CATEGORY_NAME INTO NOMBRE_CAT FROM CATEGORIES  
    WHERE CATEGORY_ID = ID_CAT;  
  
END;
```

--- Ejecución del procedimiento

--- Bloque anónimo

```
DECLARE  
  
    NOMBRE_CATEGORIA CATEGORIES.CATEGORY_NAME%TYPE;  
  
BEGIN  
  
    SP_NOMBRE_CAT(2,NOMBRE_CATEGORIA);  
  
    DBMS_OUTPUT.PUT_LINE('El nombre de la categoria es: '||NOMBRE_CATEGORIA);  
  
END;
```

### **Ejemplo de SP (Store Procedure) usando parámetro IN OUT**

--- Obtener el nombre de la categoria = 2 y retornar el valor obtenido usando variable IN OUT

```
CREATE OR REPLACE PROCEDURE SP_NOMBRE_CAT(ID_CAT IN NUMBER,  
NOMBRE_CAT IN OUT CATEGORIES.CATEGORY_NAME%TYPE)
```

IS

BEGIN

```
SELECT CATEGORIES.CATEGORY_NAME INTO NOMBRE_CAT FROM CATEGORIES  
WHERE CATEGORY_ID = ID_CAT;
```

```
END;
```

```
--- Ejecución del procedimiento
```

```
--- Bloque anonimo
```

```
DECLARE
```

```
NOMBRE_CATEGORIA CATEGORIES.CATEGORY_NAME%TYPE:='VALOR ANTES DE  
SP';
```

```
BEGIN
```

```
DBMS_OUTPUT.PUT_LINE('El nombre de la categoria es: '||NOMBRE_CATEGORIA);
```

```
SP_NOMBRE_CAT(2,NOMBRE_CATEGORIA);
```

```
DBMS_OUTPUT.PUT_LINE('El nombre de la categoria es: '||NOMBRE_CATEGORIA);
```

```
END;
```

### **Ejemplo SP Sin parámetros**

Crear un SP que obtenga la cantidad total de productos

```
CREATE OR REPLACE PROCEDURE SP_CANT_PRODUCTOS
```

```
IS
```

```
--- Variable para almacenar la cantidad de productos
```

```
CANTIDAD NUMBER;
```

```
BEGIN
```

```
SELECT COUNT(*) INTO CANTIDAD FROM PRODUCTS;
```

```
DBMS_OUTPUT.PUT_LINE('Cantidad total de los productos: '||CANTIDAD);
```



END;

EXECUTE SP\_CANT\_PRODUCTOS;

### **DUAL Almacena el valor de SYSDATE**

SELECT SYSDATE FROM DUAL;

Los procedimientos almacenados solo pueden ser utilizados a partir del comando EXECUTE o dentro de un bloque anónimo.

### **Clase #6**

#### **Ejemplo Crear procedimiento almacenado para insertar productos controlando errores con excepciones**

Es importante revisar si hay alguna secuencia que controle a la llave primaria.

### **Apuntes de la clase #6 - #12 en el .sql**

### **Clase #13**

#### **Paquetes**

Estructura o forma de encapsular funciones, procedimientos, variables en un solo lugar para ordenarlos en base a paquetes. También es un subprograma de Oracle. El paquete puede verse como una clase a la hora de instanciar sus funciones y SP, Haciendo referencia al nombre de la clase "." Y luego al nombre de la función, SP o variable que se quiere obtener.

Ejemplo: Paquete de proceso de matrícula de clases.

Encabezado de FN o SP: Se compone de Nombre del procedimiento o función, los tipos de datos que recibe y los valores de retorna.

#### **Sintaxis básica del encabezado de paquete**

CREATE OR REPLACE PACKAGE <NOMBRE DEL PAQUETE>

IS

<CREAR VARIABLES>

<INDICAR EL ENCABEZADO DE LAS FUNCIONES>

<INDICAR EL ENCABEZADO DE LOS PROCEDIMIENTOS ALMACENADOS>

END;

### **Sintaxis básica del cuerpo de paquete**

CREATE OR REPLACE PACKAGE BODY <NOMBRE DEL PAQUETE>

IS

<CUERPO DE LAS FUNCIONES>

<CUERPO DE LOS PROCEDIMIENTOS ALMACENADOS>

END;