Diseño e implementación del data warehouse de una cadena de tiendas de ropa

Memoria

TFC – Bases de datos relacionales

Fecha: 11/01/2016

Autor: Jesús Antonio Camba Fuentes

Consultor: Manel Rella Ruiz

Este documento está dedicado a todas las personas que de una u otra forma han contribuido a que esta entrega sea posible.

Una mención muy especial a mi familia, su apoyo durante estos años ha sido fundamental.

También agradecer a todo el equipo docente de la UOC su ayuda y dedicación durante este periodo de formación.

Resumen

El objetivo de este proyecto es consolidar los conocimientos adquiridos durante estos años de estudio.

A lo largo del documento, se explicarán las etapas realizadas, para entregar una solución acorde a las necesidades del cliente.

El producto resultante permitirá analizar y tomar decisiones que faciliten la gestión y el funcionamiento del negocio.

No es necesario desarrollar un entorno gráfico, todo el trabajo se centra unicamente en el diseño de la base de datos.

El software empleado será Oracle Database 11g Express Edition y Oracle SQL Developer 4.0.2.15.

Índice

1. Introducción

Una importante cadena de ropa, solicita propuestas para crear un sistema de base de datos, que actúe como data warehouse centralizado. La finalidad es poder tomar decisiones analizando el funcionamiento del negocio.

El objetivo del proyecto, será entregar un producto que se adapte a las necesidades solicitadas y pueda resultar seleccionado en el concurso publico.

La base de datos debe permitir almacenar información de cada una de las tiendas, el catalogo de productos y actualizar la tabla de hechos que contendrá información de las ventas realizadas cada día.

Las principales consultas que deben proporcionarse son las siguientes (todas recibirán como parámetro el año y el mes a consultar):

1. Información asociada a cada una de las tiendas (número total de productos vendidos, beneficio neto, etc.)

2. Información asociada a los productos del catalogo (códigos de identificación, número de unidades vendidas, tienda que vendió más unidades, etc.)

3. Información de las ventas realizadas cada día del mes (beneficio total, identificador del producto más vendido, tienda que más beneficio generó y la cantidad de dicho beneficio).

El sistema debe incluir un módulo estadístico, que proporcione la siguiente información:

1. Beneficio total de la cadena, información del producto más vendido, día en que se vendieron más productos, ciudad en la que se consiguieron más beneficios, porcentaje de los beneficios proporcionados por las tiendas virtuales, etc.

2. Los datos mencionados en el punto anterior se limitan a un año recibido por parámetro, es necesario poder consultar esta misma información desde el momento en que se fundó la cadena.

1.1. Justificación del trabajo final de carrera (TFC) y contexto en el que se desarrolla

El TFC es una asignatura que permite realizar un trabajo de síntesis en base a los conocimientos adquiridos durante de la carrera.

La finalidad es poner en práctica los conocimientos adquiridos en las asignaturas de bases de datos, empleando el lenguaje PL/SQL y conociendo nuevas herramientas de desarrollo.

1.2. Metodología utilizada

Teniendo en cuenta que el objetivo a conseguir esta claro, se utilizará como método de desarrollo el ciclo de vida en cascada. El producto pasará por las etapas de requisitos, análisis, diseño, implementación, pruebas y documentación.

El motivo de escoger este sistema, es que el producto en desarrollo no sufrirá cambios durante el tiempo que dura el proyecto, en el caso de existir modificaciones deberían detectarse en la primeras fases (análisis de requisitos y diseño). Otra de las ventajas que proporciona, es la posibilidad de centrarse en una fase concreta, disminuyendo los conceptos a tratar en cada punto.

Se establecerán ciclos de retroalimentación al final de cada etapa. De esta forma se pueden corregir los diferentes errores que vayan surgiendo durante el diseño.

1.3. Planificación

A continuación se muestra una breve descripción de las entregas que se han realizado durante la elaboración del proyecto y el material aportado en cada una de ellas.

Posteriormente se incluye un diagrama con las tareas resultantes del análisis de los requisitos solicitados por el cliente.

1.- Plan de trabajo.

Incluye la preparación del entorno de desarrollo, lectura del enunciado del proyecto y la planificación de las tareas detectadas.

2.- Diseño de la base de datos.

Desarrollo a nivel conceptual, lógico y físico de la base de datos, debe estar acompañado de un script que permita crear de forma rápida el diseño propuesto.

3.- Implementación de procedimientos almacenados y funciones.

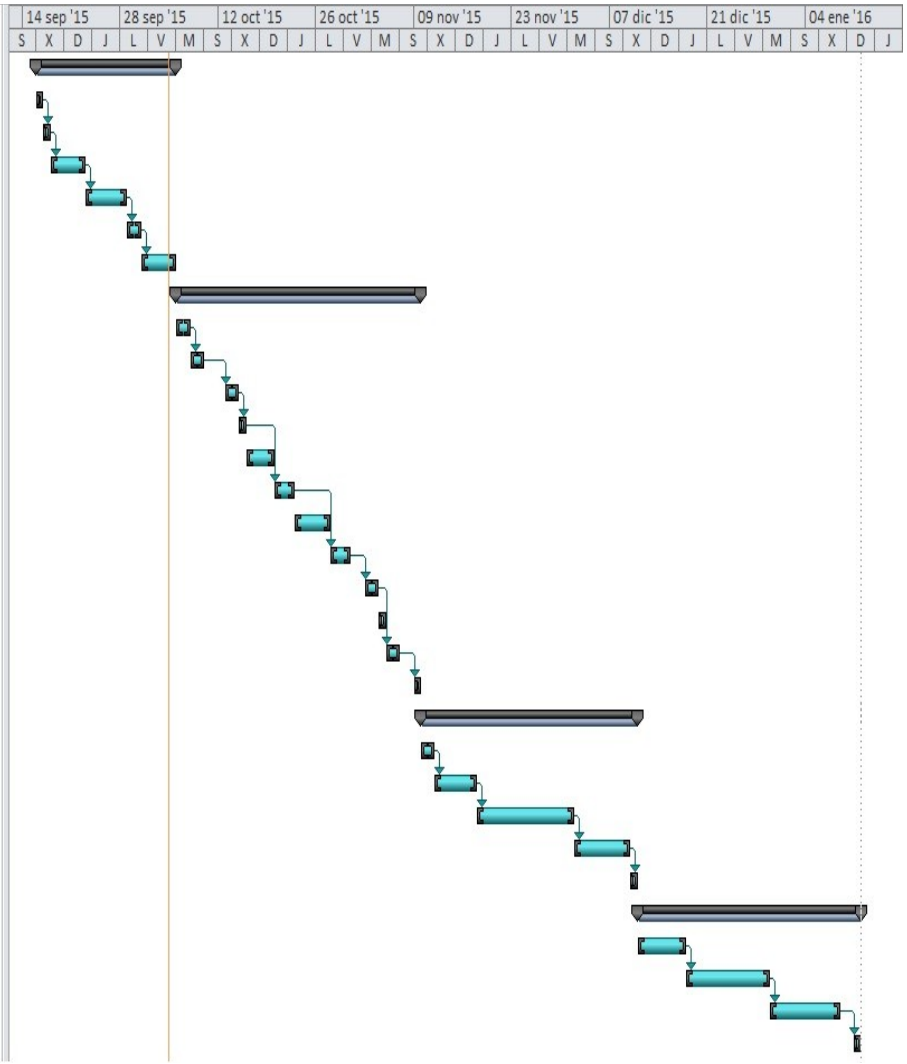
En esta etapa se realiza la codificación necesaria para cumplir con los requisitos solicitados por el cliente. Deben contemplarse los posibles errores que puedan surgir al recibir los datos de la aplicación ERP utilizada por la cadena.

4.- Fases de pruebas

Se verificará que el producto proporcionado cumple la funcionalidad solicitada.

1.4. Planificación de tareas

	Nombre de tarea	Duración	Comienzo	Fin
1	**PAC 1 - Plan de trabajo**	**13 días**	**mié 16/09/15**	**lun 05/10/15**
2	Lectura del plan docente	1 día	mié 16/09/15	mié 16/09/15
3	Instalación de software (MagicDraw, SQL Developer, Oracle Express, Microsoft Project)	1 día	jue 17/09/15	jue 17/09/15
4	Consultar fuentes de información (materiales didácticos)	3 días	vie 18/09/15	mar 22/09/15
5	Lectura del enunciado de proyecto	3 días	mié 23/09/15	lun 28/09/15
6	Diagrama de Gantt	2 días	mar 29/09/15	mié 30/09/15
7	Correcciones finales y entrega PAC 1	3 días	jue 01/10/15	lun 05/10/15
8	**PAC 2 - Material a entregar**	**24 días**	**mar 06/10/15**	**lun 09/11/15**
9	Consultar fuentes de información (materiales didácticos)	2 días	mar 06/10/15	mié 07/10/15
10	Análisis de requisitos	2 días	jue 08/10/15	vie 09/10/15
11	Diseño nivel conceptual, lógico y físico	2 días	mar 13/10/15	mié 14/10/15
12	Scripts creación de tablas, índices y disparadores	1 día	jue 15/10/15	jue 15/10/15
13	Procedimientos auxiliares (verificar número, fecha, calcular beneficio neto, etc.)	2 días	vie 16/10/15	lun 19/10/15
14	Procedimientos almacenados para gestionar altas	3 días	mar 20/10/15	jue 22/10/15
15	Procedimientos almacenados para gestionar modificaciones	3 días	vie 23/10/15	mar 27/10/15
16	Procedimientos almacenados para gestionar bajas	3 días	mié 28/10/15	vie 30/10/15
17	Procedimientos almacenados módulo estadístico	2 días	lun 02/11/15	mar 03/11/15
18	Conjunto de datos para inicialización	1 día	mié 04/11/15	mié 04/11/15
19	Juego de pruebas	2 días	jue 05/11/15	vie 06/11/15
20	Correcciones finales y entrega PAC 2	1 día	lun 09/11/15	lun 09/11/15
21	**PAC 3 - Material a entregar**	**22 días**	**mar 10/11/15**	**jue 10/12/15**
22	Consultar fuentes de información (materiales didácticos)	2 días	mar 10/11/15	mié 11/11/15
23	Correcciones detectadas en la entrega anterior	4 días	jue 12/11/15	mar 17/11/15
24	Documentar la memoria	10 días	mié 18/11/15	mar 01/12/15
25	Realizar la presentación	5 días	mié 02/12/15	mié 09/12/15
26	Correcciones finales y entrega PAC 3	1 día	jue 10/12/15	jue 10/12/15
27	**Entrega final del TFC**	**19 días**	**vie 11/12/15**	**lun 11/01/16**
28	Correcciones detectadas en la entrega anterior	5 días	vie 11/12/15	jue 17/12/15
29	Revisar producto final	7 días	vie 18/12/15	mar 29/12/15
30	Revisar documentación	6 días	mié 30/12/15	vie 08/01/16
31	Entrega del TFC	1 día	lun 11/01/16	lun 11/01/16



1.5. Producto

Al final del proyecto se entregará la siguiente documentación:

Producto: contendrá los scripts necesarios para crear la base de datos, carga inicial de información y el conjunto de pruebas que permitan verificar la validez de la solución creada.

Memoria: documento explicando el trabajo realizado para llevar a cabo la elaboración del proyecto.

Presentación: resumen de la estructura del proyecto.

1.5.1. Análisis de requisitos

En base a la documentación del proyecto, se observan los siguientes requisitos funcionales:

1.- Almacenar los datos básicos de cada tienda.

- Identificador, Ciudad, Región, E-mail del gerente, número de trabajadores, indicar si se trata de una franquicia y si es o no una tienda virtual.

2.- Almacenar el catálogo de productos que puede vender cada tienda.

- Identificador EAN13 del producto, descripción del producto y fecha de incorporación al catálogo.

- Para simplificar, no es necesario guardar el histórico de precios de cada producto, la propia cadena se encarga de almacenar esta información.

3.- Almacenar la tabla de hechos con la información de las ventas realizadas cada día.

- Las dimensiones primarias serán: identificador de la tienda, identificador del producto, fecha en formato dd/mm/aaaa, hora sin minutos en formato 24 horas.

- Asociada a cada clave primaria, se registrará la cantidad de productos vendidos ese día y hora, el precio bruto total de los productos vendidos y el beneficio neto total.

1.5.2. Funcionalidades

1.- Procedimientos de ABM (Alta + Baja + Modificación) de las tiendas, productos y tabla de hechos.

2.- No es necesario implementar los procedimientos de ABM correspondientes a las tablas maestras.

3.- Procedimiento de consulta, recibiendo como parámetro el año y el mes a consultar, permitirá obtener un listado de todas las tiendas incluyendo para cada una de ellas:

- El número total de productos vendidos.
- El número total de productos diferentes.
- El beneficio neto total obtenido.
- El porcentaje de beneficio que aporta la tienda en relación a la cadena.
- El beneficio neto dividido por el número de empleados.

El listado debe estar ordenado por el beneficio neto en orden descendente.

4.- Procedimiento de consulta, recibiendo como parámetro el año y el mes a consultar, permitirá obtener un listado de todos los productos del catálogo incluyendo para cada uno de ellos:

- El identificador del producto.
- La descripción del producto.
- El número de unidades vendidas.
- El beneficio neto generado.
- La tienda que ha vendido más unidades.
- La cantidad vendida.

El listado debe estar ordenado por el beneficio neto en orden descendente.

5.- Procedimiento de consulta, recibiendo como parámetro el año y el mes a consultar, permitirá obtener un listado de todos los días del mes, incluyendo para cada día:

- Beneficio neto obtenido ese día por toda la cadena.
- Identificador del producto más vendido, junto con las unidades vendidas.
- Identificador de la tienda que más beneficio neto ha generado y el valor en euros de dicho beneficio.

A continuación se muestran las funcionalidades que debe contemplar el módulo estadístico, estas debe alimentarse a partir de las indicadas en los puntos anteriores.

2.- Procedimiento de consulta, recibiendo como parámetro el año a consultar, permitirá obtener la siguiente información:

- Beneficio neto de toda la cadena.
- Identificador de la tienda que tenga mayor beneficio neto, así como la cantidad total de este beneficio en euros.

- Identificador del producto más vendido, así como la cantidad total de unidades vendidas.
- La hora del día en que se han vendido más productos y la cantidad total.
- La hora del día donde menos productos se han vendido y la cantidad total.
- El día del mes en el que se realizaron más ventas y la cantidad de productos vendidos.
- El día del mes en que se realizaron menos ventas y la cantidad de productos vendidos.
- La ciudad en que se obtuvieron más beneficios netos y el importe de este beneficio.
- El porcentaje de los beneficios conseguidos por las tiendas virtuales con respecto a la cadena.

3.- Esta misma consulta debe poder realizarse de forma histórica es decir teniendo en cuenta todos los años de existencia de la cadena.

1.5.3. Requerimientos de metodología

1.- Realizar los procedimientos almacenados necesarios para llevar a cabo la funcionalidades descritas. Deben de estar suficientemente documentados para que puedan ser usados sin problemas por los programadores de la capa de presentación.

2.- Los procedimientos dispondrán como mínimo de un parámetro de salida, que indicará si la ejecución finalizó correctamente o en caso de error indicará el problema detectado.

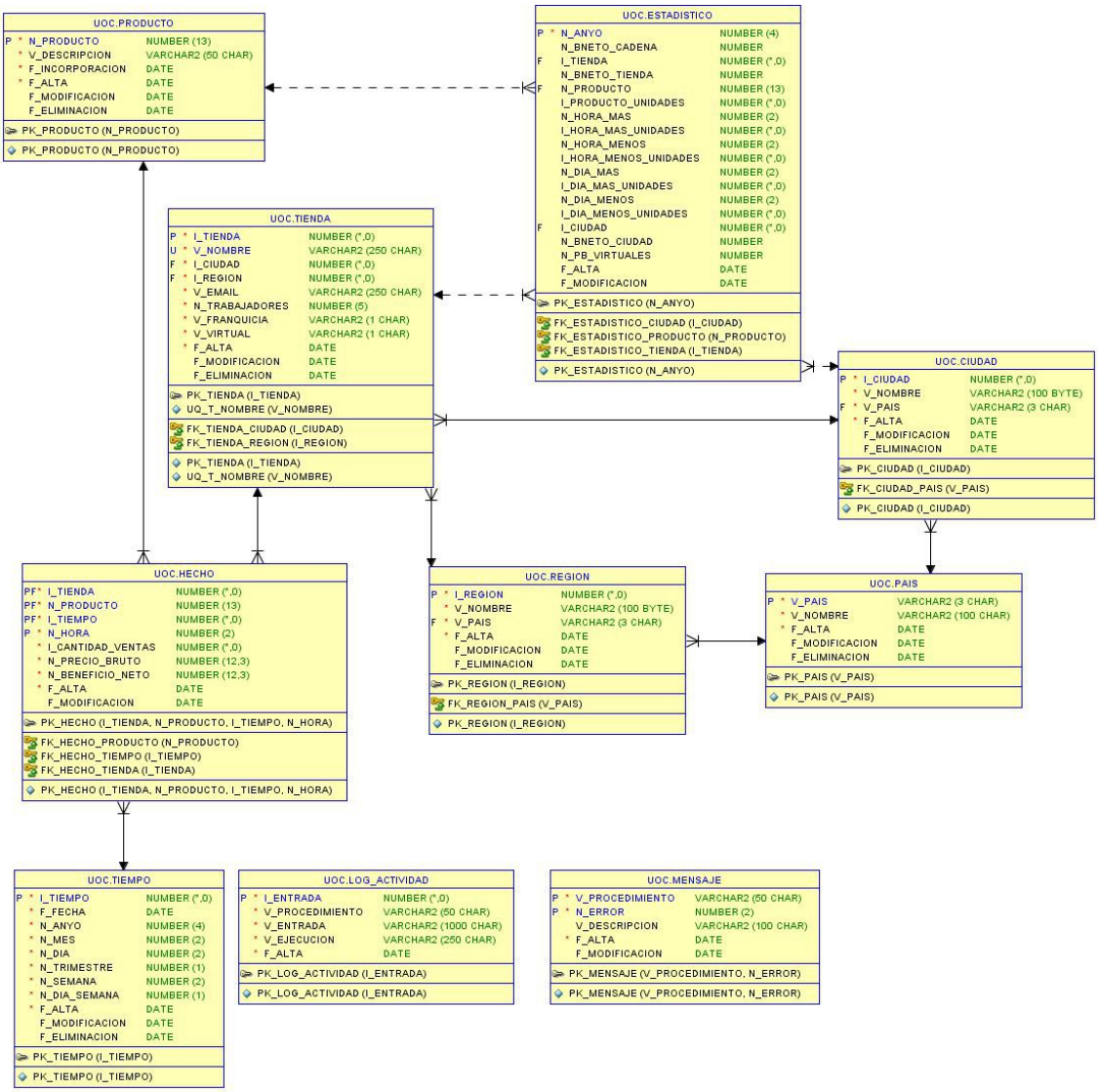
3.- Los procedimientos dispondrán de tratamiento de excepciones.

4.- Se creará un log que registrará la actividad de la base de datos, indicando el procedimiento ejecutado, los parámetros de entrada y la salida de dicha ejecución.

1.5.4. Diseño

En la imagen se puede observar las tablas que forman el diseño, así como las relaciones existentes entre ellas.

También se pueden observar las restricciones de integridad.



1.5.5 Implementación

A continuación se indicarán los diferentes archivos que componen la implementación, los pasos a seguir serán los siguientes:

- Conectarse a la base de datos con un usuario que tenga permisos de administrador.
- Situar los scripts en una carpeta que sea accesible desde Oracle SQL Developer. Se puede configurar en: Menú herramientas – Preferencias – Base de datos – Hoja de trabajo.
- Crear dentro de esta carpeta, una subcarpeta llamada logs para almacenar el resultado de las diferentes ejecuciones que se realice. En función de la configuración del entorno de trabajo es posible que está subcarpeta deba crearse en otro directorio.
- En el caso de querer ejecutar el proceso manualmente, es necesario respetar el orden indicado en cada uno de los archivos.

Para facilitar los puntos anteriores se crea el archivo **01-Generacion.sql**.

Los scripts que contiene son los siguientes:

- @02-Tablas.sql;
- @03-Secuencias_y_triggers.sql;
- @04-Tipos.sql;
- @05-Datos_iniciales.sql;
- @06-Package_Procedimientos.sql;
- @07-Package_Funciones.sql;
- @08-Package_Verificacion.sql;
- @09-Package_Tiendas.sql;
- @10-Package_Productos.sql;
- @11-Package_Hechos.sql;
- @12-Package_Consultas.sql;
- @13-Package_Historicos.sql;
- @14-Pruebas_Realizadas.sql;

La creación del entorno y el usuario debe realizarse antes de proceder con el orden propuesto.

1.5.5.1. Creación Tablespace

Archivo que contiene el script: **00-Tablespace_y_usuario.sql**

Permite configurar la unidad lógica de almacenamiento, permitirá almacenar los objetos del esquema de la base de datos, tablas, índices, etc.

Se definirá para que se gestione de forma automática la asignación de espacio, aumentando en caso de necesitar más recursos.

1.5.5.2. Creación del usuario y asignación de permisos

Archivo que contiene el script: **00-Tablespace_y_usuario.sql**

En este caso se crea uno propio para el proyecto y debe tener permisos suficientes para administrar los elementos que componen la solución entregada (tablas, procedimientos almacenados, funciones, etc.)

1.5.5.3. Creación de tablas

Archivo que contiene el script: **02-Tablas.sql**

Las tablas necesarias son las siguientes, se resaltan las claves primarias.

LOG_ACTIVIDAD: registra la actividad en la base de datos.

Columna	Tipo	NULL	Descripción
I_Entrada	Integer	No	Código de identificación
V_Procedimiento	Varchar2 (50 char)	No	Nombre del procedimiento
V_Entrada	Varchar2 (1000 char)	No	Parámetros de entrada
V_Ejecucion	Varchar2 (250 char)	No	Resultado de la ejecución
F_Alta	Date	No	Fecha de incorporación a la tabla

PAIS: países en los que es posible localizar una tienda.

Columna	Tipo	NULL	Descripción
V_Pais	Varchar2 (3 char)	No	Código de identificación
V_Nombre	Varchar2 (100 char)	No	Nombre del país
F_Alta	Date	No	Fecha de incorporación a la tabla
F_Modificacion	Date	Si	Fecha última modificación
F_Eliminacion	Date	Si	Fecha de eliminación

CIUDAD: ciudades en las que es posible localizar una tienda.

Columna	Tipo	NULL	Descripción
I_Ciudad	Integer	No	Código de identificación
V_Nombre	Varchar2 (100 char)	No	Nombre de la ciudad
V_Pais	Varchar2 (3 char)	No	Nombre del país
F_Alta	Date	No	Fecha de incorporación a la tabla
F_Modificacion	Date	Si	Fecha última modificación
F_Eliminacion	Date	Si	Fecha de eliminación

REGION: regiones en las que es posible localizar una tienda.

Columna	Tipo	NULL	Descripción
I_Region	Integer	No	Código de identificación
V_Nombre	Varchar2 (100 char)	No	Nombre de la región
V_Pais	Varchar2 (3 char)	No	Nombre del país
F_Alta	Date	No	Fecha de incorporación a la tabla
F_Modificacion	Date	Si	Fecha última modificación
F_Eliminacion	Date	Si	Fecha de eliminación

TIENDA: contiene las tiendas que gestiona la cadena.

Columna	Tipo	NULL	Descripción
I_Tienda	Integer	No	Código de identificación
V_Nombre	Varchar2 (250 char)	No	Nombre de la tienda
I_Ciudad	Integer	No	Ciudad en la que está ubicada
I_Region	Integer	No	Región en la que está ubicada
V_Email	Varchar2 (250 char)	No	E-mail del gerente
N_Trabajadores	Number (5)	No	Número de trabajadores
V_Franquicia	Varchar2 (1 char)	No	Indica si es una franquicia (S/N)
V_Virtual	Varchar2 (1 char)	No	Indica si es virtual (S/N)
F_Alta	Date	No	Fecha de incorporación a la tabla
F_Modificacion	Date	Si	Fecha última modificación
F_Eliminacion	Date	Si	Fecha de eliminación

PRODUCTO: productos pertenecientes al catálogo.

Columna	Tipo	NULL	Descripción
N_Producto	Number (13)	No	Código de identificación
V_Descripcion	Varchar2 (50 char)	No	Descripción del producto
F_Incorporacion	Date	No	Fecha de incorporación al catálogo
F_Alta	Date	No	Fecha de incorporación a la tabla
F_Modificacion	Date	Si	Fecha última modificación
F_Eliminacion	Date	Si	Fecha de eliminación

TIEMPO: fechas en las que se realizó alguna venta

Columna	Tipo	NULL	Descripción
I_Tiempo	Integer	No	Código de identificación
F_Fecha	Date	No	Fecha en la que se realizó la venta
N_Anyo	Number (4)	No	Año
N_Mes	Number (2)	No	Mes
N_Dia	Number (2)	No	Día
N_Trimestre	Number (1)	No	Trimestre
N_Semana	Number (2)	No	Número de semana
N_Dia_Semana	Number (1)	No	Día de la semana
F_Alta	Date	No	Fecha de incorporación a la tabla
F_Modificacion	Date	Si	Fecha última modificación
F_Eliminacion	Date	Si	Fecha de eliminación

HECHO: registra las ventas realizadas proporcionadas por el ERP de la compañía.

Columna	Tipo	NULL	Descripción
I_Tienda	Integer	No	Identificador de la tienda
N_Producto	Number (13)	No	Identificador del producto
I_Tiempo	Integer	No	Identificador de la fecha de venta
N_Hora	Number (2)	No	Hora en la que se realizó la venta
I_Cantidad_Ventas	Integer	No	Cantidad total de venta
N_Precio_Bruto	Number (12,3)	No	Precio bruto total
N_Beneficio_Neto	Number (12,3)	No	Beneficio neto total
F_Alta	Date	No	Fecha de incorporación a la tabla
F_Modificacion	Date	Si	Fecha última modificación

ESTADISTICO: almacena la información asociada al módulo estadístico.

Columna	Tipo	NULL	Descripción
N_Anyo	Number (4)	No	Año en que se realizó la venta
N_BNeto_Cadena	Number	No	Beneficio neto total de la cadena
I_Tienda	Integer	No	Tienda con mayor beneficio neto
N_BNeto_Tienda	Number	No	Beneficio neto de la tienda
N_Producto	Number (13)	No	Producto más vendido
I_Producto_Unidades	Integer	No	Total de unidades vendidas
N_Hora_Mas	Number (2)	No	Hora en la que se vende más
I_Hora_Mas_Unidades	Integer	No	Total de unidades vendidas
N_Hora_Menos	Number (2)	No	Hora en la que se vende menos
I_Hora_Menos_Unidades	Integer	No	Total de unidades vendidas
N_Dia_Mas	Number (2)	No	Día del mes en que se vende más
I_Dia_Mas_Unidades	Integer	No	Total de unidades vendidas
N_Dia_Menos	Number (2)	No	Día del mes en que se vende menos
I_Dia_Menos_Unidades	Integer	No	Total de unidades vendidas
I_Ciudad	Integer	No	Ciudad con mayor beneficio
N_Bneto_Ciudad	Number	No	Total del beneficio conseguido
N_PB_Virtuales	Number	Si	% Tiendas virtuales en relación a la cadena.
F_Alta	Date	No	Fecha de incorporación a la tabla
F_Modificacion	Date	Si	Fecha ultima modificación

MENSAJE: contiene los mensajes de error resultado de una posible ejecución.

Columna	Tipo	NULL	Descripción
V_Procedimiento	Varchar2 (50 char)	No	Procedimiento que reporta el error
N_Error	Number (2)	No	Identificador del mensaje
V_Descripción	Varchar2 (100 Char)	No	Descripción del mensaje.
F_Alta	Date	No	Fecha de incorporación a la tabla
F_Modificacion	Date	Si	Fecha última modificación

1.5.5.4. Creación de secuencias

Archivo que contiene el script: **03-Secuencias_y_triggers.sql**

Permiten incrementar los campos que definen la clave principal de las tablas Log_Actividad, Tienda y Tiempo.

Cada una va acompañada de un trigger para incrementar y asociar el nuevo identificador:

- SEQ_LOG_ENTRADA, incrementa el campo I_Entrada en la tabla Log_Actividad.
- SEQ_ID_TIENDA, incrementa el campo I_Tienda en la tabla Tienda.
- SEQ_ID_TIEMPO, incrementa el campo I_Tiempo en la tabla Tiempo.

1.5.5.5. Creación de tipos

Archivo que contiene el script: **04-Tipos.sql**

Define las tablas y tipos de objetos necesarios para almacenar el resultado de las consultas que debe proporcionar la base de datos.

También están contempladas las consultas correspondientes al módulo estadístico.

1.5.5.6. Carga de datos iniciales

Archivo que contiene el script: **05-Datos_iniciales.sql**

Actualiza la información que contienen las tablas auxiliares implicadas: países, regiones, ciudades y mensajes.

1.5.5.7. Package_Procedimientos

Archivo que contiene el script: **06-Package_Procedimientos.sql**

Agrupa los procedimientos auxiliares.

Nombre del procedimiento: usp_insertar_log

Parámetros de entrada:

I_Procedimiento: Nombre del procedimiento ejecutado – Varchar2(50)

I_Entrada: Parámetros del procedimiento – Varchar2(1000)

I_Ejecucion: Resultado de la ejecución – Varchar2(250)

Ejecución:

Almacena el nombre del procedimiento ejecutado, los parámetros empleados y el resultado de la ejecución.

Salida:

Existe un nuevo registro en la tabla LOG_ACTIVIDAD.

Nombre del procedimiento: usp_insertar_tiempo

Parámetros de entrada:

I_Fecha: Fecha a incorporar - Date

Ejecución:

Verifica si la fecha recibida como parámetro existe en la tabla TIEMPO.

En el caso negativo, obtiene el año, mes, día, trimestre, semana y día de la semana a partir de la fecha recibida.

Salida:

La fecha recibida está registrada en la tabla TIEMPO.

Retorna el identificador de tiempo asociado al registro.

1.5.5.7. Package_Funciones

Archivo que contiene el script: **07-Package_Funciones.sql**.

Agrupa las funciones auxiliares.

Nombre de la función: f_es_fecha

Parámetros de entrada:

I_Fecha: Fecha a verificar – Varchar2

Ejecución:

Verifica que la fecha informada es correcta.

Salida:

Boolean, verdadero en el caso que la fecha sea correcta.

Nombre de la función: f_es_numero

Parámetros de entrada:

I_Numero: Número a verificar – Varchar2

Ejecución:

Verifica que un campo numérico es correcto.

Salida:

BOOLEAN, verdadero en el caso de que el número a verificar sea correcto.

Nombre de la función: f_mensaje

Parámetros de entrada:

I_Procedimiento: Nombre del procedimiento – Varchar2

I_Error: Identificador del error solicitado - Number

Ejecución:

Retorna el mensaje correspondiente al procedimiento y número de error indicado.

Salida:

Varchar2

Observaciones

Si no localiza el mensaje, retorna “el mensaje solicitado no existe'

Nombre de la función: f_tiempo

Parámetros de entrada:

I_Fecha: Fecha a localizar – DATE

Ejecución:

Retorna el identificador de tiempo correspondiente a la fecha indicada.

Salida:

Integer

Observaciones:

Si la fecha proporcionada no existe en la tabla, retorna un cero.

1.5.5.8. Package_Verificacion

Archivo que contiene el script: **08-Package_Verificacion.sql**

Agrupa los procedimientos utilizados para verificar el formato y la información de los datos recibidos.

Nombre del procedimiento: usp_verificar_tienda

Parámetros de entrada:

I_Tienda: Código de identificación de la tienda - Integer

O_RSP: Resultado de la ejecución - Varchar2

Ejecución:

Verifica la existencia de la tienda.

Salida:

Si finaliza correctamente la ejecución, confirmará que la tienda informada existe.

Errores tratados:

No se informa el código de identificación de la tienda.

La tienda informada no existe.

La tienda informada está anulada.

SQLCODE – SQLERRM (si se produce un error distinto de los indicados).

Nombre del procedimiento: usp_verificar_nombre

Parámetros de entrada:

I_Tienda: Código de identificación de la tienda. - Integer

I_Nombre: Nombre a verificar – Varchar(250 char)

O_RSP: Resultado de la ejecución - Varchar2

Ejecución:

Verifica el nombre de la tienda para evitar que existan duplicados..

Salida:

Si finaliza correctamente la ejecución, confirmará que el nombre proporcionado es único.

Errores tratados:

No se informa el nombre de la tienda.

El nombre informado está asociado a otra tienda.

SQLCODE – SQLERRM (si se produce un error distinto de los indicados).

Nombre del procedimiento: usp_verificar_ciudad

Parámetros de entrada:

I_Ciudad: Código de identificación de la ciudad - Integer

O_RSP: Resultado de la ejecución - Varchar2

Ejecución:

Verifica la existencia de la ciudad en la que está ubicada la tienda.

Salida:

Si finaliza correctamente la ejecución, confirmará que el código de ciudad proporcionado es correcto.

Errores tratados:

No se informa la ciudad en la que está ubicada la tienda.

La ciudad informada no existe.

La ciudad informada está anulada.

SQLCODE – SQLERRM (si se produce un error distinto de los indicados).

Nombre del procedimiento: usp_verificar_region

Parámetros de entrada:

I_Region: Código de identificación de la región - Integer

O_RSP: Resultado de la ejecución - Varchar2

Ejecución:

Verifica la existencia de la región en la que está ubicada la tienda.

Salida:

Si finaliza correctamente la ejecución, confirmará que el código de región proporcionado es correcto.

Errores tratados:

No se informa la región en la que está ubicada la tienda.

La región informada no existe.

La región informada está anulada.

SQLCODE – SQLERRM (si se produce un error distinto de los indicados).

Nombre del procedimiento: usp_verificar_email

Parámetros de entrada:

I_Email: E-Mail del gerente de la tienda – Varchar2(250 char)

O_RSP: Resultado de la ejecución - Varchar2

Ejecución:

Verifica que el E-Mail tenga la estructura esperada.

Salida:

Si finaliza correctamente la ejecución, confirmará que la dirección de correo sigue el estándar previsto.

Errores tratados:

No se informa el e-mail del gerente de la tienda.

El e-mail informado no es correcto.

SQLCODE – SQLERRM (si se produce un error distinto de los indicados).

Nombre del procedimiento: usp_verificar_trabajadores

Parámetros de entrada:

I_Trabajadores: Indica el número de trabajadores de la tienda – Number(5)

O_RSP: Resultado de la ejecución - Varchar2

Ejecución:

Verifica que el número de trabajadores informado sea numérico y distinto de cero.

Salida:

Si finaliza correctamente la ejecución, confirmará que el número de trabajadores informado es correcto.

Errores tratados:

No se informa el número de trabajadores de la tienda.

El número de trabajadores debe ser numérico.

SQLCODE – SQLERRM (si se produce un error distinto de los indicados).

Nombre del procedimiento: usp_verificar_ubicacion

Parámetros de entrada:

I_Ciudad: Identifica la ciudad donde está ubicada la tienda – Integer.

I_Region: Identifica la región donde está ubicada la tienda – Integer.

O_RSP: Resultado de la ejecución - Varchar2

Ejecución:

Verifica que la ciudad y la región pertenezcan al mismo país.

Salida:

Si finaliza correctamente la ejecución, confirmará que la asociación pertenece al mismo país.

Errores tratados:

El país de la ciudad y la región no es coincidente.

SQLCODE – SQLERRM (si se produce un error distinto de los indicados).

Nombre del procedimiento: usp_verificar_producto

Parámetros de entrada:

I_Producto: Identifica el producto a verificar – Number(13)

I_Comprobacion: Indica si se trata de un alta o eliminación – Varchar2(1 char).

O_RSP: Resultado de la ejecución - Varchar2

Ejecución:

Verifica la existencia y el formato del código de producto.

Salida:

Si finaliza correctamente la ejecución, confirmará que el producto informado es correcto.

Errores tratados:

No se informa el código del producto.

El código de producto informado no es correcto.

El producto informado no existe.

El producto informado está anulado.

SQLCODE – SQLERRM (si se produce un error distinto de los indicados).

Nombre del procedimiento: usp_verificar_anyo

Parámetros de entrada:

I_Anyo: Contiene el año a validar – Varchar2

O_RSP: Resultado de la ejecución - Varchar2

Ejecución:

Verifica que sea numérico y que su longitud sea igual a cuatro.

Salida:

Si finaliza correctamente la ejecución, confirmará que el año informado es correcto.

Errores tratados:

No se informa el año a consultar.

El año recibido como parámetro no es correcto.

SQLCODE – SQLERRM (si se produce un error distinto de los indicados).

Nombre del procedimiento: usp_verificar_mes

Parámetros de entrada:

I_Mes: Contiene el mes validar – Varchar2

O_RSP: Resultado de la ejecución - Varchar2

Ejecución:

Verifica que sea numérico y su valor esté comprendido entre uno y doce.

Salida:

Si finaliza correctamente la ejecución, confirmará que el mes informado es correcto.

Errores tratados:

No se informa el mes a consultar.

El mes recibido como parámetro no es correcto.

SQLCODE – SQLERRM (si se produce un error distinto de los indicados).

Nombre del procedimiento: usp_verificar_tiempo

Parámetros de entrada:

I_Tiempo: Contiene el código de identificación asociado a una fecha – Varchar2

O_RSP: Resultado de la ejecución - Varchar2

Ejecución:

Verifica que se trate de una fecha.

Salida:

Si finaliza correctamente la ejecución, confirmará que la fecha recibida es correcta.

Errores tratados:

No se informa la fecha a verificar.

El fecha recibida como parámetro no es correcta.

SQLCODE – SQLERRM (si se produce un error distinto de los indicados).

Nombre del procedimiento: usp_verificar_hora

Parámetros de entrada:

I_Hora: Hora a verificar – Varchar2

O_RSP: Resultado de la ejecución - Varchar2

Ejecución:

Verifica que se trate de una número y que su valor esté comprendido entre 0 y 23.

Salida:

Si finaliza correctamente la ejecución, confirmará que la hora recibida es correcta.

Errores tratados:

No se informa la hora a verificar.

El hora recibida como parámetro no es correcta.

SQLCODE – SQLERRM (si se produce un error distinto de los indicados).

Nombre del procedimiento: usp_verificar_cantidad_ventas

Parámetros de entrada:

I_Cantidad_Ventas: Cantidad de las ventas realizadas a verificar – Varchar2

O_RSP: Resultado de la ejecución - Varchar2

Ejecución:

Verifica que se trate de una número y que su valor sea mayor de cero.

Salida:

Si finaliza correctamente la ejecución, confirmará que la cantidad recibida es correcta.

Errores tratados:

No se informa la cantidad de ventas a verificar.

La cantidad de ventas recibida como parámetro no es correcta.

SQLCODE – SQLERRM (si se produce un error distinto de los indicados).

Nombre del procedimiento: usp_verificar_precio_bruto

Parámetros de entrada:

I_Precio_Bruto: Precio bruto a verificar – Varchar2

O_RSP: Resultado de la ejecución - Varchar2

Ejecución:

Verifica que se trate de una número y que su valor sea mayor de cero.

Salida:

Si finaliza correctamente la ejecución, confirmará que la cantidad recibida es correcta.

Errores tratados:

No se informa el precio bruto a verificar.

El precio bruto recibido como parámetro no es correcto.

SQLCODE – SQLERRM (si se produce un error distinto de los indicados).

Nombre del procedimiento: usp_verificar_beneficio_neto

Parámetros de entrada:

I_Beneficio_Neto: Beneficio neto a verificar – Varchar2

O_RSP: Resultado de la ejecución - Varchar2

Ejecución:

Verifica que se trate de una número.

Salida:

Si finaliza correctamente la ejecución, confirmará que la cantidad recibida es correcta.

Errores tratados:

No se informa el beneficio neto a verificar.

El beneficio neto recibido como parámetro no es correcto.

SQLCODE – SQLERRM (si se produce un error distinto de los indicados).

1.5.5.9. Package_Tiendas

Archivo que contiene el script: **09-Package_Tiendas.sql**

Agrupa los procedimientos asociados a la gestión de tiendas.

Nombre del procedimiento: usp_insertar_tienda

Parámetros de entrada:

I_Nombre: Nombre de la tienda - Varchar2(250 char)

I_Ciudad: Código de la ciudad en la que está ubicada – Integer

I_Region: Código de la región a la que pertenece – Integer

I_Email: Email del gerente - Varchar2(250 char)

I_Trabajadores: Número de trabajadores – Number(5)

I_Franquicia: Indica si se trata de una franquicia - Varchar2(1 char)

I_Virtual: Indica si se trata de una tienda virtual - Varchar2(1 char)

O_RSP: Resultado de la ejecución - Varchar2

Ejecución:

Permite incorporar una nueva tienda a la base de datos.

Salida:

Si finaliza correctamente la ejecución, existirá un nuevo registro en la tabla TIENDA.

Errores tratados:

No se informa el nombre de la tienda.

No se informa la ciudad en la que está ubicada la tienda.

No se informa la región en la que está ubicada la tienda.

No se informa el Email del gerente de la tienda.

El Email informado no es correcto.

No se informa el número de trabajadores de la tienda.

El número de trabajadores debe ser un valor numérico.

No se informa correctamente si la tienda es una franquicia.

No se informa correctamente si la tienda es virtual.

El nombre informado está asociado a otra tienda.

La ciudad informada no existe.

La ciudad informada está anulada.

La región informada no existe.

La región informada está anulada.

El e-mail informado no es correcto.

El número de trabajadores informado no es correcto.

La tienda ya existe.

El país de la ciudad y la región no es coincidente.

SQLCODE – SQLERRM (si se produce un error distinto de los indicados).

Observaciones:

El número de trabajadores debe ser mayor de cero.

Franquicia y Virtual solo aceptan valores S o N.

1.5.5.10. Package_Productos

Archivo que contiene el script: 1**0-Package_Productos.sql**

Agrupa los procedimientos asociados a la gestión de productos.

Nombre del procedimiento: usp_insertar_producto

Parámetros de entrada:

I_Producto: Código de identificación del producto - Number(13).

I_Descripción: Descripción del producto – Varchar2(50 char)

I_FIncorporacion: Fecha en la que se incorpora al catalogo – Date.

O_RSP: Resultado de la ejecución - Varchar2

Ejecución:

Permite incorporar un nuevo producto a la base de datos.

Salida:

Si finaliza correctamente la ejecución, existirá un nuevo registro en la tabla PRODUCTO.

Errores tratados:

No se informa el código de producto.

El código de producto informado no es correcto.

No se informa la descripción del producto.

No se informa la fecha de incorporacion al catálogo.

La fecha de incorporación informada no es correcta.

El producto ya existe.

El código de producto informado no es correcto.

SQLCODE – SQLERRM (si se produce un error distinto de los indicados).

Nombre del procedimiento: usp_actualizar_producto

Parámetros de entrada:

I_Producto: Código de identificación del producto - Number(13).

I_Descripción: Descripción del producto – Varchar2(50 char)

I_FIncorporacion: Fecha en la que se incorpora al catalogo – Date.

O_RSP: Resultado de la ejecución - Varchar2

Ejecución:

Permite actualizar la información del producto.

Salida:

Si finaliza correctamente la ejecución, se actualizarán los datos correspondientes al producto.

Errores tratados:

No se informa el código de producto.

No existe el producto a modificar.

La fecha de incorporación informada no es correcta.

SQLCODE – SQLERRM (si se produce un error distinto de los indicados).

Nombre del procedimiento: usp_eliminar_producto

Parámetros de entrada:

I_Producto: Código de identificación del producto - Integer

O_RSP: Resultado de la ejecución - Varchar2

Ejecución:

Permite eliminar física o lógicamente el producto informado.

Salida:

Si finaliza correctamente la ejecución, el producto no puede ser utilizado en futuras inserciones.

Errores tratados:

No se informa el código del producto.

El producto informado no existe.

El producto informado está anulado.

El código del producto informado no es correcto.

SQLCODE – SQLERRM (si se produce un error distinto de los indicados).

1.5.5.11. Package_Hechos

Archivo que contiene el script: **11-Package_Hechos.sql**

Agrupa los procedimientos asociados a la gestión de hechos.

Nombre del procedimiento: usp_insertar_hecho

Parámetros de entrada:

I_Tienda: Código de identificación de la tienda - Integer.

I_Producto: Código de identificación del producto – Number(13)

I_Fecha: Fecha en la que se realizaron las ventas – Date.

I_Hora: Hora en la que se realizó la venta – Number(2).

I_Cantidad_Ventas: Cantidad del producto vendida en el día/hora – Integer.

I_Precio_Bruto: P. Bruto de los productos vendidos en el día/hora – Number(12,3)

I_Beneficio_Neto: B. Neto de los productos vendidos en el día/hora – Number(12,3)

O_RSP: Resultado de la ejecución - Varchar2

Ejecución:

Permite añadir un nuevo registro con la información de la venta del día a la base de datos.

Salida:

Si finaliza correctamente la ejecución, existirá un nuevo registro en la tabla HECHO.

Errores tratados:

Las ventas diarias ya existen en la tabla.

SQLCODE – SQLERRM (si se produce un error distinto de los indicados).

Nombre del procedimiento: usp_actualizar_hecho

Parámetros de entrada:

I_Tienda: Código de identificación de la tienda - Integer.

I_Producto: Código de identificación del producto – Number(13)

I_Fecha: Fecha en la que se realizaron las ventas – Date.

I_Hora: Hora en la que se realizó la venta – Number(2).

I_Cantidad_Ventas: Cantidad del producto vendida en el día/hora – Integer.

I_Precio_Bruto: P. Bruto de los productos vendidos en el día/hora – Number(12,3)

I_Beneficio_Neto: B. Neto de los productos vendidos en el día/hora – Number(12,3)

O_RSP: Resultado de la ejecución - Varchar2

Ejecución:

Permite modificar un registro existente con la información de la venta del día.

Salida:

Si finaliza correctamente la ejecución, se actualizaran los datos correspondientes a la venta.

Errores tratados:

Los parámetros recibidos no permiten localizar el registro a modificar.

No existe el registro de ventas diarias.

SQLCODE – SQLERRM (si se produce un error distinto de los indicados).

Nombre del procedimiento: usp_eliminar_hecho

Parámetros de entrada:

I_Tienda: Código de identificación de la tienda - Integer.

I_Producto: Código de identificación del producto – Number(13)

I_Fecha: Fecha en la que se realizaron las ventas – Date.

I_Hora: Hora en la que se realizó la venta – Number(2).

O_RSP: Resultado de la ejecución - Varchar2

Ejecución:

Permite eliminar físicamente el hecho informado.

Salida:

Si finaliza correctamente la ejecución, la venta diaria no se contabilizará.

Errores tratados:

SQLCODE – SQLERRM (el resto de errores se monitorizan en las funciones de verificación).

1.5.5.12. Package_Consultas

Archivo que contiene el script: **12-Package_Consultas.sql**

Agrupa los procedimientos asociados a la gestión de consultas.

Nombre del procedimiento: f_Consulta_A

Parámetros de entrada:

I_Anyo: Identifica el año a consultar - Number(4).

I_Mes: Identifica el mes a consultar – Number(2)

Ejecución:

Lista todas las tiendas de la cadena incluyendo para cada una de ellas:

Número total de productos vendidos durante un mes.

El número de productos diferentes vendidos durante el mes.

El beneficio neto total obtenido durante el mes.

El porcentaje de beneficio que aporta la tienda en relación al total de la cadena durante el mes.

El beneficio neto dividido por el número de empleados de la tienda.

Salida:

Retorna el resultado de la consulta.

Errores tratados:

No se informa el año a consultar.

No se informa el mes a consultar.

El año recibido como parámetro no es correcto.

El mes recibido como parámetro no es correcto.

SQLCODE – SQLERRM (si se produce un error distinto de los indicados).

Nombre del procedimiento: f_Consulta_B

Parámetros de entrada:

I_Anyo: Identifica el año a consultar - Number(4).

I_Mes: Identifica el mes a consultar – Number(2)

Ejecución:

Lista todos los productos del catalogo incluyendo para cada uno de ellos:

Identificador EAN13.

Nombre del producto.

Número de unidades vendidas.

Beneficio neto que ha generado el producto.

Tienda que ha vendido más unidades.

Número de unidades vendidas en dicha tienda.

Salida:

Retorna el resultado de la consulta.

Errores tratados:

No se informa el año a consultar.

No se informa el mes a consultar.

El año recibido como parámetro no es correcto.

El mes recibido como parámetro no es correcto.

SQLCODE – SQLERRM (si se produce un error distinto de los indicados).

Nombre del procedimiento: f_Consulta_C

Parámetros de entrada:

I_Anyo: Identifica el año a consultar - Number(4).

I_Mes: Identifica el mes a consultar – Number(2)

Ejecución:

Lista todos los días del mes, incluyendo para cada uno de ellos:

Total beneficio neto obtenido ese día por toda la cadena.

Identificador EAN13 del producto más vendido.

Número de unidades vendidas del producto más vendido.

Identificador de la tienda que más beneficio neto ha generado.

El valor en euros del beneficio generado por dicha tienda.

Salida:

Retorna el resultado de la consulta.

Errores tratados:

No se informa el año a consultar.

No se informa el mes a consultar.

El año recibido como parámetro no es correcto.

El mes recibido como parámetro no es correcto.

SQLCODE – SQLERRM (si se produce un error distinto de los indicados).

Nombre del procedimiento: f_Consulta_H

Parámetros de entrada:

I_Anyo: Identifica el año a consultar - Number(4)

Ejecución:

Lista la información asociada al histórico para un año o para todos:

Beneficio neto de toda la cadena.

Identificador de la tienda que más beneficio neto ha generado.

Cantidad obtenida por dicha tienda.

Identificador del producto más vendido.

Número de unidades vendidas de dicho producto.

La hora del día en que más productos se han vendido.

Número de unidades vendidas en dicho hora.

La hora del día en que menos productos se han vendido.

Número de unidades vendidas en dicha hora.

El día del mes en que más productos se han vendido.

Número de unidades vendidas en dicho día.

El día del mes en que menos productos se han vendido.

Número de unidades vendidas en dicho día.

Ciudad donde más beneficios se han conseguido.

La cantidad de beneficio conseguida.

Porcentaje de beneficios conseguidos por las tiendas virtuales con respecto a la cadena.

Salida:

Retorna el resultado de la consulta.

Errores tratados:

El año recibido como parámetro no es correcto.

SQLCODE – SQLERRM (si se produce un error distinto de los indicados).

1.5.5.13. Package_Historicos

Archivo que contiene el script: **13-Package_Historicos.sql**

Agrupa los procedimientos asociados a la gestión de históricos.

Nombre del procedimiento: usp_calcular_beneficio_cadena

Parámetros de entrada:

I_Anyo: Año a calcular - Number(4).

O_BNeto_Cadena: Retorna el beneficio neto de la cadena – Number.

O_RSP: Resultado de la ejecución - Varchar2

Ejecución:

Permite obtener el beneficio neto de la cadena, si no recibe el año a consultar realiza la consulta de forma histórica.

Salida:

Si finaliza correctamente la ejecución, entregará el beneficio neto de la cadena.

Errores tratados:

SQLCODE – SQLERRM

Nombre del procedimiento: usp_calcular_beneficio_tienda

Parámetros de entrada:

I_Anyo: Año a calcular - Number(4).

O_Tienda: Identificador de la tienda - Integer.

O_BNeto_Tienda: Retorna el beneficio neto de la tienda – Number.

O_RSP: Resultado de la ejecución - Varchar2

Ejecución:

Permite obtener el identificador de la tienda que más benefició neto obtuvo, así como la cifra de dicho beneficio. Si no recibe el año a consultar realiza la consulta de forma histórica.

Salida:

Si finaliza correctamente la ejecución, entregará el identificador de la tienda y su beneficio.

Errores tratados:

SQLCODE – SQLERRM

Nombre del procedimiento: usp_calcular_producto_unidades

Parámetros de entrada:

I_Anyo: Año a calcular - Number(4).

O_Producto: Identificador del producto – Number(13).

O_Producto_Unidades: Retorna el número de unidades vendidas – Integer.

O_RSP: Resultado de la ejecución - Varchar2

Ejecución:

Permite obtener el identificador del producto más vendido, así como el número de unidades. Si no recibe el año a consultar realiza la consulta de forma histórica.

Salida:

Si finaliza correctamente la ejecución, entregará el identificador del producto y el número de unidades vendidas.

Errores tratados:

SQLCODE – SQLERRM

Nombre del procedimiento: usp_calcular_hmas_unidades

Parámetros de entrada:

I_Anyo: Año a calcular - Number(4).

O_Hora_Mas: Hora del día – Number(2).

O_Hora_Mas_Unidades: Retorna el número de unidades vendidas – Integer.

O_RSP: Resultado de la ejecución - Varchar2

Ejecución:

Permite obtener la hora del día donde más productos se han vendido, así como el número de unidades. Si no recibe el año a consultar realiza la consulta de forma histórica.

Salida:

Si finaliza correctamente la ejecución, entregará la hora del día y el número de unidades vendidas.

Errores tratados:

SQLCODE – SQLERRM

Nombre del procedimiento: usp_calcular_hmenos_unidades

Parámetros de entrada:

I_Anyo: Año a calcular - Number(4).

O_Hora_Menos: Hora del día – Number(2).

O_Hora_Menos_Unidades: Retorna el número de unidades vendidas – Integer.

O_RSP: Resultado de la ejecución - Varchar2

Ejecución:

Permite obtener la hora del día donde menos productos se han vendido, así como el número de unidades. Si no recibe el año a consultar realiza la consulta de forma histórica.

Salida:

Si finaliza correctamente la ejecución, entregará la hora del día y el número de unidades vendidas.

Errores tratados:

SQLCODE – SQLERRM

Nombre del procedimiento: usp_calcular_dmas_unidades

Parámetros de entrada:

I_Anyo: Año a calcular - Number(4).

O_Dia_Mas: Día – Number(2).

O_Dia_Mas_Unidades: Retorna el número de unidades vendidas – Integer.

O_RSP: Resultado de la ejecución - Varchar2

Ejecución:

Permite obtener el día donde más productos se han vendido, así como el número de unidades. Si no recibe el año a consultar realiza la consulta de forma histórica.

Salida:

Si finaliza correctamente la ejecución, entregará el día y el número de unidades vendidas.

Errores tratados:

SQLCODE – SQLERRM

Nombre del procedimiento: usp_calcular_dmenos_unidades

Parámetros de entrada:

I_Anyo: Año a calcular - Number(4).

O_Dia_Menos: Día – Number(2).

O_Dia_Menos_Unidades: Retorna el número de unidades vendidas – Integer.

O_RSP: Resultado de la ejecución - Varchar2

Ejecución:

Permite obtener el día donde menos productos se han vendido, así como el número de unidades. Si no recibe el año a consultar realiza la consulta de forma histórica.

Salida:

Si finaliza correctamente la ejecución, entregará el día y el número de unidades vendidas.

Errores tratados:

SQLCODE – SQLERRM

Nombre del procedimiento: usp_calcular_beneficio_ciudad

Parámetros de entrada:

I_Anyo: Año a calcular - Number(4).

O_Ciudad: Código identificador de la ciudad con mayor beneficio – Integer.

O_BNeto_Ciudad: Retorna el beneficio neto – Number.

O_RSP: Resultado de la ejecución - Varchar2

Ejecución:

Permite obtener el código identificador de la ciudad que más beneficio neto obtuvo, así como la cantidad de dicho beneficio. Si no recibe el año a consultar realiza la consulta de forma histórica.

Salida:

Si finaliza correctamente la ejecución, entregará el código de la ciudad y el beneficio neto conseguido.

Errores tratados:

SQLCODE – SQLERRM

Nombre del procedimiento: usp_calcular_porcentaje

Parámetros de entrada:

I_Anyo: Año a calcular - Number(4).

O_PB_Virtuales: Porcentaje asociado a las tiendas virtuales – Number.

O_RSP: Resultado de la ejecución - Varchar2

Ejecución:

Permite obtener el porcentaje del beneficio de las tiendas virtuales con respecto al total de la cadena. Si no recibe el año a consultar realiza la consulta de forma histórica.

Salida:

Si finaliza correctamente la ejecución, entregará el porcentaje correspondiente a la categoría demandada.

Errores tratados:

SQLCODE – SQLERRM

Nombre del procedimiento: usp_actualizar_historico

Parámetros de entrada:

I_Anyo: Año a calcular - Number(4).

Ejecución:

Actualiza la información correspondiente al módulo estadístico. Si no recibe el año a consultar realiza la consulta de forma histórica.

Errores tratados:

SQLCODE – SQLERRM

2. Gestión de riesgos

Descripción del riesgo	Probabilidad	Impacto	Contramedida
Enfermedad	Baja	Alto	Incrementar el número de horas de dedicación una vez recuperado (dependería de la gravedad)
Carga elevada de trabajo	Alta	Medio	Compensar las horas perdidas trabajando el fin de semana.
Problemas técnicos con el equipo o software.	Baja	Bajo	Gestión de copias de seguridad y disponer de un equipo configurado para sustituir el primero.
Viajes imprevistos	Baja	Medio	Se continuaría con el proyecto a pesar de que el número de horas diarias pudiese sufrir variación.
Incoherencia de los datos, perdida de integridad y redundancia.	Media	Alta	Incrementar el número de pruebas realizadas.

3. Fechas y dedicación

Tarea	Fecha inicial	Fecha final	Nº Días	Horas	Material entregado
PAC 1	16/09/2015	05/10/2015	13	26	Plan de trabajo
PAC 2	06/10/2015	09/11/2015	24	48	Producto
PAC 3	10/11/2015	10/12/2015	22	44	Memoria / Presentación
Entrega final	11/01/2016	11/01/2016	19	38	Correcciones finales

En lo que respecta a la dedicación, se emplearan dos horas diarias de lunes a viernes, no se incluyen días festivos.

4. Coste económico

Los perfiles que intervienen en el proceso de creación son los siguientes:

Jefe de proyecto: será el encargado de coordinar el equipo de trabajo, verificando que se cumplan los requisitos solicitados por el cliente.

Analista: realizará el análisis de los requisitos funcionales del sistema, diseño conceptual y lógico de la base de datos.

Programador: instalación del software (si el proyecto fuese mayor, aparecería la figura del administrador de base de datos), creación de la base de datos, procedimientos almacenados, triggers y juego de pruebas.

Documentalista: será el encargado de redactar la memoria y elaborar la presentación del proyecto.

El calculo del coste del proyecto se basará en los siguientes precios/hora:

- Jefe de proyecto: 60 €.
- Analista: 40 €.
- Programador: 35 €.
- Documentalista: 15 €.

Perfil	Total Horas	Coste / Hora	Importe
Jefe de proyecto	20	60	1200 €
Analista	32	40	1280 €
Programador	56	35	1960 €
Documentalista	48	15	720 €
Total	156		5.160 €

El proyecto tiene una duración de 78 días, con una dedicación prevista de dos horas diarias. El importe resultante bruto es de 5.160 €.

A esta cantidad se se le aplicará el 21 % de IVA. El importe a facturar al cliente será de 6.243 € (IVA 1083 €)

5. Posibles mejoras

Incorporar información asociada a los clientes, se podría estudiar los hábitos de compra, permitiendo de esta forma adaptarse a las necesidades de cada una de las regiones en la que está desplegada la cadena.

Ejemplos:
Se podría analizar las edades de los compradores y los lugares de origen. Esto permitiría adaptar los stocks y las ofertas de cada tienda en base a las costumbres locales, incluso a la climatología del lugar de origen, etc.

Otro posible estudio sería realizar un seguimiento del material sobrante, este podría distribuirse a las tiendas con mayor demanda. Realizando un estudio exhaustivo de este concepto, permitiría acondicionar la logística del negocio, reduciendo considerablemente los gastos de distribución.

6. Conclusión

Considero que el trabajo realizado, es muy importante para conocer los problemas que implica el desarrollo de un proyecto de estas características.

Se confirma la importancia de las primeras etapas, cuanto más tiempo se les dedique mejor será el resultado y el cumplimiento de los objetivos solicitados.

Un factor vital es intentar crear procedimientos sencillos que puedan ensamblarse para conseguir el producto deseado.

La documentación implica gran trabajo y tiempo, puede caerse en la equivocación de no asignarlo correctamente.

7. Bibliografía

- Apuntes de las asignaturas bases de datos I y II.
- Apuntes de la asignatura Ingeniería del software.
- https://es.wikipedia.org/wiki/Wikipedia:Portada
- César Pérez. “Oracle 9i. Administración y análisis de bases de datos”, Ra-Ma, (ISBN: 84-7897-523-3)