

Base de Datos Aplicadas

Curso de Verano
Comisión 1353
Grupo 1353

HNATIUK, JAIR EZEQUIEL



Realizado por:
Barrionuevo Luciano
45429539
Paula Nyszta
45129511

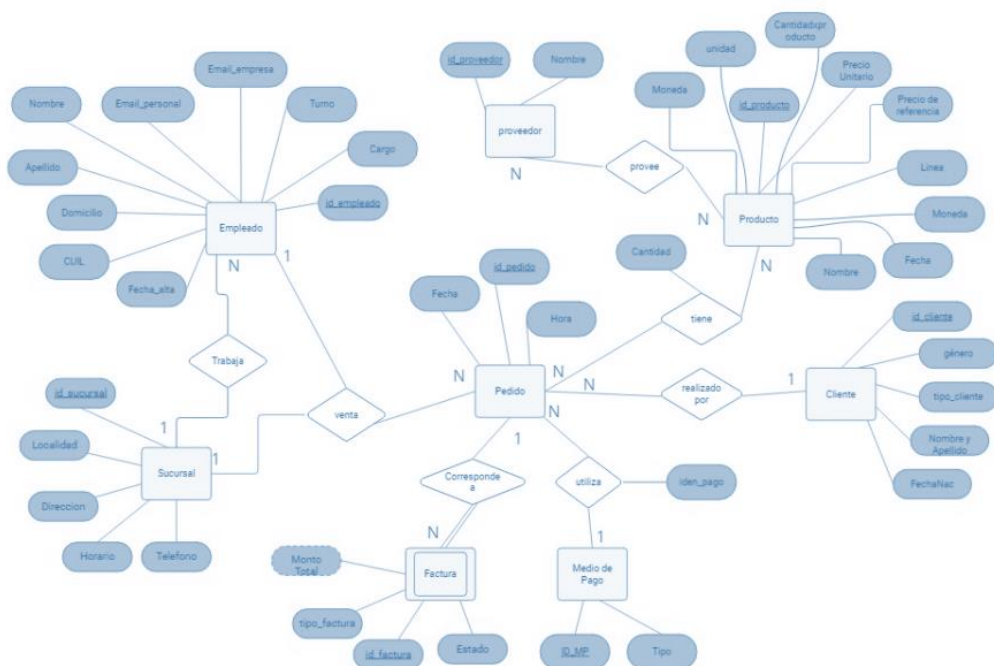
Consignas:

Repositorio Github:

<https://github.com/PaulaNyszta/BBDD-APLICADAS>

Entrega 3

1. Incluya un DER con el diseño de la base de datos. Deben aparecer correctamente las relaciones y las claves, pero el formato queda a su criterio.



Página utilizada para realizar el DER: Miro

https://miro.com/welcomeonboard/VHBkT1F2ZTQzeE9wUXg4T3AveUhXelRc0ZrUUyUzlxbmTzdUgVbVFleXlXQ1pseEd4UVh1akhKROFuVERDQWIRTW02VTJCYYtuU0FTenorOTF1djh6NjNqTCtOSXhFd2VQd1dEMEziV056QXRjRXMrSi9VZk9OWk9ocDZKTStnbHpza3F6REdEcmNpNEFOMmJXWXBPT0h0djE=?share_link_id=102485045162

Adjuntamos archivo con nombre DER.pdf

El diseño debe estar normalizado y en los casos donde el grupo decida normalizar (3ra FN) debe incluirse la fundamentación para tal decisión.

Modelo Relacional

- Clave Primaria: subrayado
- Clave Foránea: negrita
- Clave Primaria + Clave Foránea: subrayado y negrita

Empleado(id_empleado, dni, cargo, turno, email_empresa, email_personal, nombre, apellido, domicilio, cuil, **id_sucursal**)

Sucursal(id_sucursal, localidad, direccion, horario, telefono)

Pedido(id_pedido, fecha, hora, iden_pago, **id_mp**, **id_cliente**)

Venta(**id_sucursal**, **id_pedido**, **id_empleado**)

Factura(id_factura, tipo_factura, **id_pedido**, fecha, estado)

MedioPago(id_mp, tipo)

Cliente(id_cliente, tipo_cliente, genero, nombre, apellido, fecha_nac)

Producto(id_producto, nombre_producto, linea, precio_unitario, precio_referencia, unidad, cantidadPorUnidad, moneda, fecha)

Proveedor(id_proveedor, nombre)

Tiene(**id_pedido**, **id_producto**, cantidad)

Provee(**id_proveedor**, **id_producto**)

¿Por qué está en Tercera Forma Normal (3FN)?

Tablas principales:

MedioPago

- **Clave primaria:** id_mp
- No posee atributos que dependan parcialmente de la clave primaria ni posee dependencias transitivas.
- Cumple con **3FN**.

Producto

- **Clave primaria:** id_producto
- No posee atributos que dependan parcialmente de la clave primaria ni posee dependencias transitivas.
- Cumple con **3FN**.

Tiene

- **Clave primaria compuesta:** id_pedido, id_producto
- Todos los atributos dependen completamente de la clave primaria compuesta.
- Cumple con **3FN**.

Proveedor

- **Clave primaria:** id_proveedor
- No posee atributos que dependan parcialmente de la clave primaria ni posee dependencias transitivas.
- Cumple con **3FN**.

Provee

- **Clave primaria compuesta:** id_proveedor, id_producto
- Todos los atributos dependen completamente de la clave primaria compuesta.
- Cumple con **3FN**.

Pedido

- **Clave primaria:** id_pedido
- No posee atributos que dependan parcialmente de la clave primaria ni posee dependencias transitivas.
- Cumple con **3FN**.

Factura

- **Clave primaria compuesta:** id_factura, id_pedido
- Todos los atributos dependen completamente de la clave primaria compuesta.
- Cumple con **3FN**.

Cliente

- **Clave primaria:** id_cliente
- Atributos como tipo_cliente, género, nombre, apellido, fecha_nac dependen directamente de id_cliente, no hay dependencias transitivas.
- Cumple con **3FN**.

Sucursal

- **Clave primaria:** id_sucursal
- No posee atributos que dependan parcialmente de la clave primaria ni posee dependencias transitivas.
- Cumple con **3FN**.

Empleado

- **Clave primaria:** id_empleado
- No posee atributos que dependan parcialmente de la clave primaria ni posee dependencias transitivas.
- Cumple con **3FN**.

Venta

- **Clave primaria compuesta:** id_sucursal, id_pedido

- Todos los atributos dependen completamente de la clave primaria compuesta.
 - Cumple con **3FN**.
2. Cree la base de datos, entidades y relaciones. Incluya restricciones y claves. Deberá entregar un archivo .sql con el script completo de creación (debe funcionar si se lo ejecuta “tal cual” es entregado en una sola ejecución). Incluya comentarios para indicar qué hace cada módulo de código.
- Genere store procedures para manejar la inserción, modificado, borrado (si corresponde, también debe decidir si determinadas entidades solo admitirán borrado lógico) de cada tabla.



SQLQuery1.sql

Utilizamos el archivo ‘Proyecto’, la Query ‘SQLQuery1.sql’ para crear tanto las tablas pertenecientes al DER como los procedimientos almacenados que se encargaran de verificar que la información enviada sea valida para insertar en las tablas.

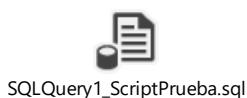
Los nombres del store procederes NO deben comenzar con “SP”.

Las tablas fueron creadas bajo el Schema ddbba y cada una de ellas esta especificada en singular. Por otro lado los procedimientos almacenados que insertar datos a las tablas se nombrar como ddbba.insertar[Tabla]

3. Genere esquemas para organizar de forma lógica los componentes del sistema y aplique esto en la creación de objetos. NO use el esquema “dbo”.

El esquema generado se llama ddbba

4. Todos los SP creados deben estar acompañados de juegos de prueba. Se espera que realicen validaciones básicas en los SP (p/e cantidad mayor a cero, CUIT válido, etc.) y que en los juegos de prueba demuestren la correcta aplicación de las validaciones.



SQLQuery1_ScriptPrueba.sql

Utilizamos el archivo ‘Proyecto’, la Query ‘SQLQuery1_ScriptPrueba.sql’ para generar todas las pruebas necesarias para validar que los Store procedures estén bien creados. Este archivo tiene instrucciones de como ejecutarse

Entrega 4 -----

Se proveen maestros de XXX.

Se requiere que importe toda la información antes mencionada a la base de datos:

- Genere los objetos necesarios (store procedures, funciones, etc.) para importar los archivos antes mencionados. Tenga en cuenta que cada mes se recibirán archivos de novedades con la misma estructura, pero datos nuevos para agregar a cada maestro.
- Considere este comportamiento al generar el código. Debe admitir la importación de novedades periódicamente sin eliminar los datos ya cargados y sin generar duplicados.

- Cada maestro debe importarse con un SP distinto. No se aceptarán scripts que realicen tareas por fuera de un SP.



SQLQuery1_ImportacionArchivos.sql

novedades periódicas.

Utilizamos el archivo 'Proyecto', la Query 'SQLQuery1_ImportacionArchivos.sql' para realizar todas las importaciones necesarias, teniendo en cuenta las

En esta entrega está incluida la generación de los informes en XML mencionados en la introducción del TP.



SQLQuery1_Reportes.sql

informes.

Utilizamos el archivo 'Proyecto', la Query 'SQLQuery1_Reportes.sql' para generar todos los reportes pedidos en la consigna, además en el archivo de Pruebas se generaron las pruebas necesarias para corroborar estos

Entrega 5 -----

1. Cuando un cliente reclama la devolución de un producto se genera una nota de crédito por el valor del producto o un producto del mismo tipo.

En el caso de que el cliente solicite la nota de crédito, solo los Supervisores tienen el permiso para generarla.

Tener en cuenta que la nota de crédito debe estar asociada a una Factura con estado pagada. Asigne los roles correspondientes para poder cumplir con este requisito.

2. Por otra parte, se requiere que los datos de los empleados se encuentren encriptados, dado que los mismos contienen información personal.



SQLQuery1_Seguridad.sql

Utilizamos el archivo 'Proyecto', la Query 'SQLQuery1_seguridad.sql' donde se realizaron los puntos de la entrega N5. En este punto se

3. La información de las ventas es de vital importancia para el negocio, por ello se requiere que se establezcan políticas de respaldo tanto en las ventas diarias generadas como en los reportes generados. Plantee una política de respaldo adecuada para cumplir con este requisito y justifique la misma. No es necesario que incluya el código de creación de los respaldos.

Frecuencia de Respaldos

Para no perder información importante y no sobrecargar los sistemas, los respaldos se harán en diferentes momentos:

Respaldo Incremental (cada hora): Guarda solo los cambios recientes desde el último respaldo. Esto evita que los archivos sean demasiado pesados y permite restaurar datos rápidamente si hay un fallo.

Respaldo Diferencial (cada noche): Antes de finalizar el día, se guarda toda la información de ventas que haya cambiado desde el último respaldo completo.

Respaldo Completo (una vez por semana): Todos los domingos a la madrugada se genera una copia completa de la base de datos, para garantizar que haya una versión intacta de toda la información.

¿Dónde se almacenarán los respaldos?

Almacenamiento Externo (en la Nube): Se usará un servicio seguro en la nube para tener una copia extra, por si ocurre un problema grave en los servidores locales (como un incendio o un ataque cibernético).

Acceso Seguro: Solo el personal autorizado podrá acceder a estos respaldos, evitando modificaciones o robos de datos.

Tiempo de Guardado de respaldos

Respaldos diarios (incrementales): Se conservan por 30 días.

Respaldos semanales (completos): Se guardan por 3 meses.

Respaldos mensuales: Se mantienen por 1 año antes de ser eliminados o archivados.
