Practica 1 - DER

1) En cada producto de una fábrica intervienen distintas partes en distintas cantidades. Cada parte se puede comprar a varios proveedores. Se compra al mejor precio. En la orden de producción figuran: la fecha de la orden, un nro. que la identifica ; el producto ; las partes que la componen y en que cantidad ; el proveedor y el precio de cada parte. Diseñar un modelo E/R que permita confeccionar esa orden y enviar por correo los pedidos a los proveedores para confeccionar cada orden.

2) Construir un modelo E/R para una empresa distribuidora de zapatillas. Tiene varios depósitos en distintos lugares del país. Según el lugar de residencia del comprador se le envía el pedido desde el depósito más cercano, previamente determinado. Si ese modelo no está disponible en ese momento se recurre a un depósito que complementa el anterior y si allí no hay se posterga la entrega. Se necesita un inventario actualizado por modelo de zapatilla en cada depósito, donde puede haber varios modelos.

3) Construir un modelo E/R para una universidad. Es necesario conocer donde (facultad, aula) ; cuando (día, hora) y quien dicta (profesor, ayudante) cada materia en cada momento. Que alumnos la están cursando. Para inscribir a cada alumno se investiga si tiene aprobadas todas las materias correlativas dentro del plan que le corresponde y se informa con quien la cursó.

4) En una línea de producción trabajan varios obreros. Cada uno en una máquina de esa línea. Los obreros están capacitados para usar varios tipos de máquinas, por lo que pueden estas asignados a una u otra máquina , en una u otra línea. Las máquinas dentro de las líneas se identifican por un número. En el inventario las máquinas tienen un código que las identifica, tipo, descripción, estado, fecha... Se pueden cambiar de línea y/o reemplazar por desperfectos. Todos los días se asignan las líneas y dentro de ellas las máquinas para resolver el problema de las inasistencias de los operarios y/o desperfectos en las máquinas. Si una máquina tiene dentro del mes mas de 10 días fuera de servicio se la reemplaza por una nueva.

5) Una base de datos de personal de una compañía con lo siguientes datos
Para cada departamento : número del departamento (único) , presupuesto y la identificación del gerente.
Para cada empleado : número del empleado (único) , número del proyecto actual, número de oficina y número telefónico. Además el título de cada uno de los trabajos que ha desarrollado y entre que períodos, indicando para que proyecto efectuó el trabajo.
Para cada proyecto : número del proyecto y presupuesto.
Para cada oficina : número de oficina, área y números de todos los teléfonos de la misma.
Con la información almacenada se tienen que poder responder las siguientes preguntas:
Cuanto gasta cada departamento
El departamento al que pertenece un empleado
Los empleados que intervienen en un proyecto
Los teléfonos donde se puede ubicar a un empleado
Quién es el gerente de un empleado
Poder contestar si el empleado XX trabajó en el proyecto A.
Saber quienes dependen del empleado XX.
Saber quienes son los empleados de un departamento

6) Una base de datos de recepción de pedidos con información acerca de clientes, artículos y órdenes.
Para cada cliente debe tener : número del cliente, direcciones de envío (varias por cliente) , saldo. límite de crédito, descuento.

Para cada pedido :
información de cabecera : número de cliente, dirección de envío, fecha del pedido.
renglón de detalle (varios) : número de artículos, cantidad pedida.
Para cada artículo : número de artículo, plantas manufactureras, cantidad en existencia en cada planta, punto de reorden para cada planta (stock mínimo) , descripción del artículo.

7) Una empresa brinda servicios en telefonía y transmisión de datos para un conjunto de clientes con oficinas ubicadas en un edificio. La empresa usa centrales ubicadas en las oficinas de las que se conoce el id de la central, la fecha del último service, a que cliente pertenece y en qué oficina está. Cada central administra un conjunto de líneas externas e internas. De cada línea se sabe el número (interno o externo), la oficina a la que pertenece, y si es externa el nombre de un responsable.
Las centrales proveen información de cada comunicación que se realiza. Si es llamada telefónica o transmisión de datos. Si es llamada telefónica de que línea se hizo y a qué número y si es comunicación digital desde que terminal, que línea, que duración y horario.

8) Una aerolínea maneja información de pasajeros, vuelos y personal. Para los pasajeros se considera de interés el #pasaporte y #vuelo. Para los vuelos #vuelo, fecha, hora, ciudad donde hace escala, personal asignado, #avión. Para los aviones se considera modelo, fabricante, capacidad, hangar. Por último para el personal se tiene en cuenta el nombre y apellido, área asignada, y en particular para los pilotos se conoce la cantidad de horas de vuelo y el tipo de avión que pilotea.

9) Una organización desea desarrollar un sistema de fidelización (CRM - Customer Relationship Management). El objetivo del sistema es el de gestionar los puntos obtenidos por los consumos realizados con la tarjeta otorgada y el canje de los mismos por premios administrando su stock.

Para ello el sistema deberá administrar los clientes que tienen un identificador, apellido, nombre, fecha de nacimiento, domicilio, localidad, y teléfono. Además debe indicar si el cliente es el titular de la tarjeta y si esta habilitado para canjear los puntos por premios.

También administrará un maestro de tarjetas que tendrá un identificador unívoco para cada tarjeta que viene grabado de fábrica, un número que es el que le otorga el sistema al ingresarla, y una fecha de baja que se usara para cuando la tarjeta sea extraviada o destruida. Cada cliente puede tener una única tarjeta en su poder en determinado momento, sin embargo, varios clientes pueden conformar un grupo, que registra el total de los consumos efectuados históricamente y el total de los consumos actuales de todos sus miembros. Este total actual de puntos acumulados es el que se puede canjear y solo lo podrán canjear aquellos miembros del grupo que estén autorizados.

Los premios tienen un identificador único, y una descripción del mismo. También se debe conocer el stock actual de cada premio, un stock de reposición y un stock mínimo del mismo porque se deberá informar semanalmente aquellos premios que estén con un stock actual menor o igual al stock de reposición.

La organización a su vez define distintas políticas de fidelización, puesto que a lo largo del año van implementando distintas estrategias de marketing, y cada una de estas políticas tiene un comienzo y fin. A su vez los premios tienen asignado un valor en puntos que depende de cada política. Es decir que una cena para dos personas en un restaurant puede tener un valor de canje de 2000 puntos para la política “Consumos en shoppping” y 500 puntos para la política “Compras de libros”. También la política debe indicar cada cuantos pesos gastados se otorga un punto.

Los grupos están asociados a varias políticas de fidelización.

Es necesario registrar los consumos de cada cliente, conociendo la fecha y hora de realización, el monto, el número de factura relacionado al consumo efectuado, bajo que política de fidelización se efectuó y los puntos obtenidos por dicho consumo.

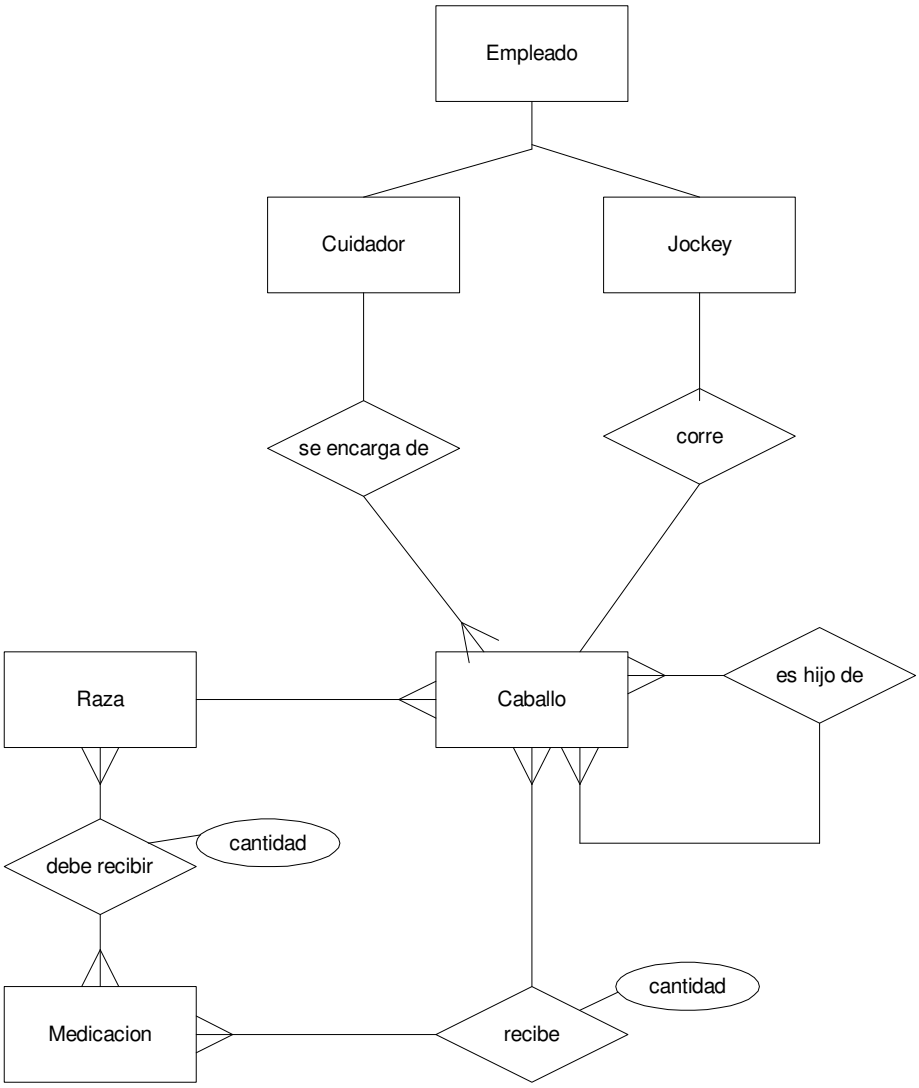
Finalmente se debe tener en cuenta el canje de puntos. Para ello el cliente debe tener autorización a realizar el canje, se le deben ofrecer solo los premios de las políticas que tiene

asignadas a su grupo con un valor de canje menor o igual al total de puntos acumulados actual y cuyo stock actual sea mayor al stock mínimo. Para ello se registra, el cliente que realiza el canje, la fecha y la hora del mismo, que política de fidelización está usando cual es el premio por el que esta efectuando el canje y cual es el valor en puntos. Luego de esto el sistema deberá actualizar el stock de los premios como así también el total de puntos acumulados descontando los puntos canjeados.

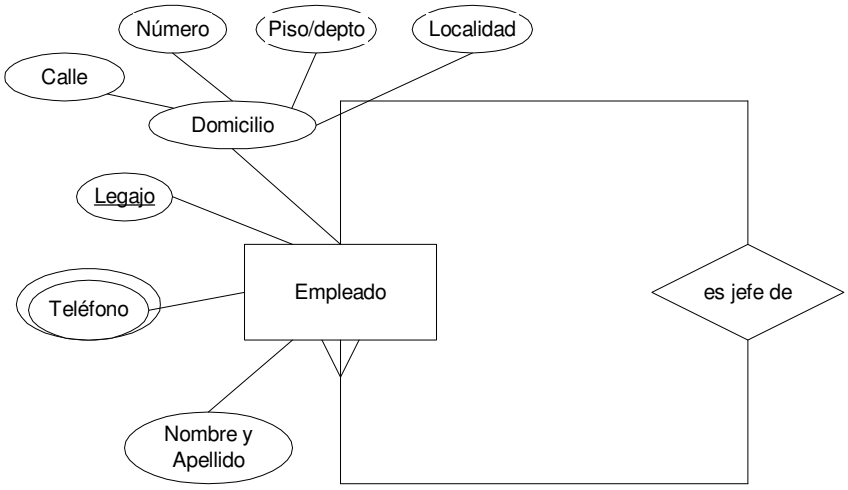
10) Una base de datos para los afiliados a obra sociales. De cada filiado nos interesa conocer su nombre, dirección(le vamos a mandar correspondencia) , teléfonos y el plan al que pertenece.
Además un afiliado puede tener varios afiliados a cargo.
Del plan nos interesa conocer el nombre, la fecha de creación, la obra social a la que corresponde y el coordinador de ventas responsable del mismo.
De los empleados el número de interno, el nombre y el cuil. En el caso de los coordinadores de ventas además queremos conocer su número de celular.
Cada plan incluye varias prestaciones que tienen una tarifa distinta según el plan y la obra social.
De las prestaciones nos interesa conocer el nombre ,categoría a la que pertenece y afiliados que las utilizaron.
Por otro lado la obra social tiene a diferentes médicos en cartilla. Cada médico tiene una especialidad y de los mismos nos interesa conocer su nombre, dirección, teléfonos y su número de matrícula médica.
Los afiliados efectúan visitas a los médicos. Necesitamos saber a que médicos visitó cada afiliado y en qué fecha para poder hacer la liquidación correspondiente de los honorarios del médico.
Tener en cuenta que un afiliado puede pertenecer a más de una obra social y utilizarlas en forma indistinta.

11) Diseñar una base de datos para almacenar los errores que se producen en programas de software que ya se encuentran en producción. De cada programa nos interesa conocer su nombre, su código y quienes fueron sus autores. Los autores son empleados de la empresa, a los que identificamos por su número de legajo. De todos los empleados tenemos su nombre y apellido, una dirección y uno o más teléfonos donde ubicarlos en caso de una emergencia. Además en el caso de los programadores sabemos que lenguajes dominan junto con una "calificación" en los mismos. Tener en cuenta que esta calificación se encuentra tabulada y la calificación de un empleado varía de acuerdo al lenguaje.
Los programadores nuevos tienen asignado otro programador que hace las veces de "tutor".
Cada programa tiene varias versiones, las mismas se identifican por un número, además de por el código del programa, y tienen una fecha en la que fueron puestas en producción. De los defectos que encontramos nos interesa conocer una descripción, quien y cuando lo detectó, en que momento del desarrollo fue introducido y en que versión fue detectado. Los defectos se encuentran clasificados por un tipo y una gravedad.
Las personas que encuentran los errores pueden ser tanto empleados del área de sistemas como otros empleados de la empresa que son usuarios finales de los programas.
Cada error da origen a un incidente que es asignado a un programador. Cuando éste termina la corrección indica la fecha y el tiempo que le llevó efectuar la misma.
Para poder mejorar la calidad de nuestro software nos interesa poder efectuar estadísticas sobre los tipos de errores más frecuentes y en que fase del desarrollo se introdujeron.

12) Escriba un enunciado que sirva como descripción del siguiente DER:

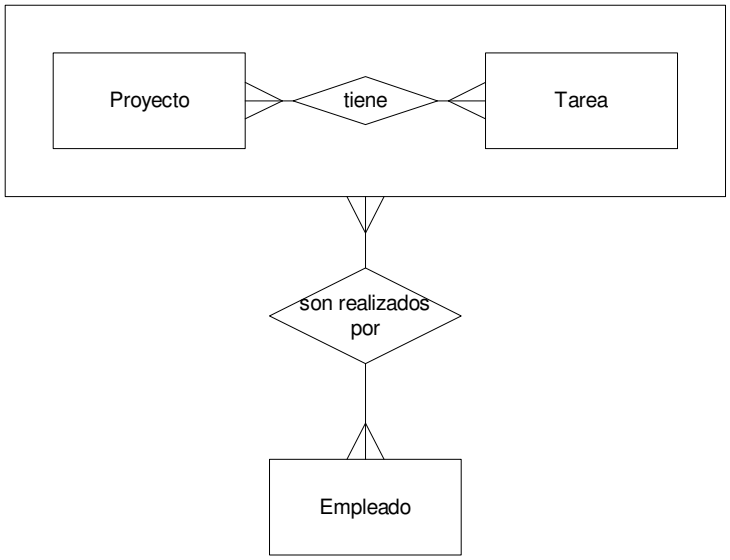


13) Dada la siguiente parte de un DER indicar cuáles de las afirmaciones detalladas son verdaderas. Tener en cuenta que hay que indicar lo que se desprende del DER y no la interpretación que se le pueda dar.



a) Cada empleado tiene varios domicilios
b) Cada empleado tiene un domicilio
c) Un empleado es jefe de varios otros
d) Un empleado tiene varios jefes
e) Un empleado tiene varios jefes y un jefe tiene varios empleados a cargo
f) Cada empleado tiene varios teléfonos
g) Cada empleado tiene un único jefe.
h) El nombre y apellido identifican al empleado
i) El legajo identifica al empleado
j) El domicilio es un atributo compuesto
k) La localidad no es un atributo del empleado

14) Dada la siguiente parte de un DER indicar cuáles de las afirmaciones detalladas son verdaderas. Tener en cuenta que hay que indicar lo que se desprende del DER y no la interpretación que se le pueda dar.



a) Un proyecto tiene varias tareas
b) Una misma tarea esta en un único proyecto
c) Una misma tarea puede estar en varios proyectos
d) Cada proyecto tiene una única tarea
e) Un empleado hace siempre las mismas tareas, no importa de qué proyecto se trate.
f) Un empleado hace un único par proyecto-tarea
g) Un empleado trabaja para un único proyecto
h) La relación proyecto – tarea se conoce previamente a saber quién va a ejecutarla.
i) Al mismo tiempo que se sabe cuáles son las tareas de un proyecto se conoce quién la va a llevar a cabo.

15) Para cada uno de las afirmaciones de los ejercicios E) y F) que haya resultado falsa indicar como debería haber sido el DER para indicar esa situación.

Practica 2 - Creación de Tablas

Crear las siguientes tablas :

EMP (codemp, nombre, coddep, sueldo)

DEP (coddep, descripcion)

Los campos subrayados corresponden a la clave primaria de la tabla.

1. Definir el campo coddep de la tabla EMP como clave foránea con respecto a la tabla DEP. Marcar que exija integridad referencial.
2. Insertar los siguientes datos en la tabla EMP (01, "Jose", 2, 800). Qué sucede ?
3. Insertar el siguiente registro en la tabla DEP (2, "Marketing")
4. Repetir el punto 2. Cuál fue el resultado ahora ? Por qué ?
5. Borrar el registro de la tabla DEP. Qué sucede ? Por qué ?
6. Modificar el código del departamento 2, sustituyéndolo por 4. Qué sucede ? Por qué ?
7. Modificar el diseño de la tabla EMP, de tal forma que el campo sueldo tenga una regla que le exija que el mismo sea mayor de 0.
8. Insertar los siguientes datos en la tabla EMP (02, "Juan", 2, -100). Qué sucede ? Por qué ?
9. Editar la relación entre las tablas EMP y DEP. Marcar la opción de update en cascada. Repetir el punto 6. Verificar que en la tabla EMP también cambió el código del depósito.
10. Editar la relación entre las tablas EMP y DEP. Marcar la opción de delete en cascada. Repetir el punto 5. Cuántos registros hay en la tabla EMP ?

Practica 3 – Lenguajes de consulta

Dadas las siguientes tablas:

Empleado (EMP)	**Artículo (ART)**
Cod_Empleado(PK)	Cod_Articulo (PK)
Nombre	Descripción
Apellido	Tipo_Articulo(A,B,C)
Dirección	Precio
Codigo_Postal	
Cod_Departamento (FK-Departamento)	**Artículo_Deposito (ART_DEP)**
Sueldo_basico	Cod_Articulo (PK) (FK-Articulo)
Fecha_ingreso	Cod_Deposito(PK) (FK-Deposito)
Fecha_nacimiento	Stock_actual
Teléfono	Punto_Reorden
Jefe (FK –Empleado)	
	Pedido (PED)
Cliente (CLI)	Nro_pedido(PK)
Cod_Cliente (PK)	Cod_Cliente(FK-Cliente)
Razon_Social	Cód_Empleado(FK-Empleado)
Dirección	Fecha_pactada_entrega
	Fecha_real_entrega
Depósito (DEP)	Deposito_entrega (FK-Deposito)
Cod_depósito (PK)	
Ubicacion_deposito	**Detalle_de_pedido (DET)**
	Nro_pedido(PK)(FK-Pedido)
Departamento (DEPA)	Cod_articulo(PK)(FK-Articulo)
Cod_departamento (PK)	Cantidad
Descripción	
Gerente(FK-Empleado)	
Cod_Dep_Padre (FK -Departamento)	

El cod_dep_padre representa al departamento del que depende un departamento de acuerdo a la estructura de la organización.
Se pide escribir las sentencias SQL correspondientes a las siguientes consultas. En algunos de los ejercicios se aclara entre paréntesis que esas mismas consultas deben ser expresadas en álgebra relacional (AR)

Simples

1. Recuperar los números de pedidos y los nombres de los clientes para los pedidos que vencen mañana. La fecha de entrega representa la fecha en que vence el plazo comprometido con el cliente (AR)
2. Recuperar los renglones de todos los pedidos, incluyendo la descripción del artículo
3. Calcular para cada renglón de pedido el costo del mismo (total de renglón más 21% de IVA).
4. Listar los datos de los artículos que se encuentren a menos de un 10% de su punto de reorden.

Manejo de texto y rangos

5. Recuperar los apellidos de los empleados (sin repeticiones).

6. Recuperar todos los números de los clientes cuyo nombre empiece con M.
7. Recuperar todos los datos de los empleados de apellido Perez.
8. Recuperar los números de los pedidos que compraron el artículo, 2, 4, 6,8 ó 10 (AR)
9. Listar los artículos cuyo precio esté entre 20.000 y 50.000.
10. Listar los pedidos que vencen la semana próxima.
11. Recuperar los pedidos para los cuales no se informó el cliente.
12. Recuperar los artículos cuya primera letra sea una R o una T y que luego continúan con S500.

Obtener con subconsultas

13. Recuperar los nombres de los diferentes artículos que tienen pedidos. (AR)
14. Recuperar los artículos que nunca fueron pedidos. (AR)
15. Recuperar el nombre de los empleados que no efectuaron ningún pedido esta semana. (AR)
16. Crear una tabla con nuevos pedidos. Listar los pedidos "viejos" y los nuevos
17. Listar todos los pedidos de la tabla anterior ordenados por fecha de entrega decreciente

Funciones de agregado

18. Recuperar el total de sueldos y el promedio de sueldos para cada departamento.
19. Recuperar el costo total de cada pedido.
20. Recuperar el total de unidades de cada artículo que hay que entregar la próxima semana.
21. Recuperar los datos de los empleados que tienen más de 3 pedidos pendientes de entrega.
22. Recuperar la cantidad de pedidos para cada empleado indicando el código y el nombre del mismo.
23. Recuperar la cantidad pedida pendiente de entrega para cada artículo.
24. Recuperar los departamentos para los cuales el promedio de sueldo de sus empleados sea superior a 3000.
25. Recuperar los artículos para los cuales la cantidad pendiente de entrega supere el stock. Los pedidos pendientes de entrega son los que no tienen informado el campo fecha_real_entrega.

Sentencias de Actualización

26. Insertar en la tabla Empleados un empleado con todos sus datos.
27. Insertar en la tabla Empleados un empleado indicando su código de empleado, su nombre y nulos o defaults en el resto de sus campos.
28. Elegir un empleado y actualizar su domicilio.
29. Agregar los empleados de la tabla Nuevos_empleados a la tabla Empleados. La tabla nuevos_empleados tiene la misma estructura que la tabla Empleados.
30. Eliminar todos los empleados con código postal 9999
31. Dar de alta 3 artículos del tipo C con todos sus datos y uno del tipo B, indicando para este último un stock de 2126 unidades.
32. Todos los artículos del tipo C pasaron a formar parte del tipo A. Actualizar la tabla con 1 instrucción.
33. Restar 268 unidades al stock de los artículos de tipo B
34. Aumentar en un 5,5% el precio de los artículos del grupo A
35. Disminuir en un 10% el precio de los artículos con el mayor stock.
36. Aumentar un 20% el sueldo básico de los empleados con el menor sueldo básico.
37. Aumentar un 15% a los empleados con más de 20 años en la empresa.
38. Aumentar en $500,00 el sueldo de los jefes de departamento que tengan el menor sueldo básico.
39. Actualizar el precio de todos los artículos que no tienen pedidos reduciéndolo en un 10%
40. Borrar todos los artículos cuyo stock es 0 y nunca han tenido pedidos.

Avanzados

41. Recuperar los artículos cuya descripción tiene al comienzo la sílaba MA y luego continúa con S550 ó S750.
42. Listar el sotck de los artículos cuya primera sílaba es ME o TE y luego continúan con R200 o R980.
43. Recuperar toda la estructura del departamento A (con todos sus subniveles)
44. Recuperar toda la línea jerárquica que se encuentra sobre el empleado 28.
45. Recuperar el nombre de todos los empleados que dependen de XXXX (en todos los niveles).
46. Recuperar el árbol que corresponde a la estructura de departamentos de la organización.
47. Recuperar el árbol que corresponde a la estructura de personal de la organización.
48. Recuperar el nombre y el departamento de los empleados de mayor sueldo.
49. Recuperar el nombre y el departamento de los empleados de mayor sueldo y de los de menor sueldo indicando en una columna la condición de mayor o de menor según corresponda.
50. Obtener el departamento de los empleados que cobran, al menos, el doble que el sueldo promedio de los empleados.
51. Listar los empleados que no son jefes de departamento y tienen sueldo mayor que el sueldo más alto de los jefes
52. Generar el pedido 24 que es igual al 13.
53. Listar los números de pedido que vendieron el artículo 23 y el 54. Resolver este ejercicio de 3 formas diferentes. (AR)
54. Listar los números de pedido que vendieron el artículo 23 o el 54. Resolver este ejercicio de 3 formas diferentes. (AR)
55. Listar los números de pedido que vendieron el artículo 23, pero no el 54. Resolver este ejercicio de 3 formas diferentes. (AR)
56. Repetir los 3 ejercicios anteriores, pero ahora recuperar los nombres de los empleados en cada una de esas situaciones.
57. Indicar para cada artículo su nombre, la cantidad total pendiente de entrega y la cantidad total en depósito.
58. Indicar el nombre de los artículos que tienen stock en todos los depósitos.

Uso de Null

59. Informar el sueldo actual de los empleados y junto con el sueldo incrementado en $500 para los empleados del departamento de Investigacion y desarrollo
60. Informar el precio actual de los articulos con stock menor a 50 unidades y el precio con un descuento del 15%
61. Informar todos los articulos con el precio menor o igual al precio del articulo 121.
62. Informar todos los articulos con los detalles de pedidos para todos independientemente que no tengan y cuya cantidad del pedido sea menor a 2
63. Obtener el monto total de sueldos por departamento.
64. Informar los artículos con el menor precio
65. Informar todos los empleados que no tienen jefe
66. Informar todos los departamentos que no tienen empleados

Practica 4 - Stored Procedures y Triggers

Crear una tabla factura con la siguiente estructura

nrofactura int
cliente varchar(100)
monto numeric (10,3)
saldo numeric(10,3)

Crear una tabla pago con la siguiente estructura

nropago int
nrofactura int (se refiere al numero de la tabla anterior)
monto numeric (10,3)

1. Crear un stored procedure para insertar en la tabla facturas y en la de pagos.
2. Crear un trigger sobre la tabla pagos que cuando se carga un pago sobre una factura actualice el saldo de la misma y controle que el monto pagado no exceda el saldo pendiente de pago.
3. Crear una tabla aud_factura (de auditoria) con una estructura igual a la de factura más usuario, fecha, operación.
4. Crear una tabla aud_pagos (de auditoria) con una estructura igual a la de pago más usuario, fecha, operación.
5. Crear los triggers necesarios para que se carguen dichas tablas cuando se producen las distintas operaciones.
6. Probar todo lo anterior.
7. Crear un stored procedure que reciba como parámetro el nombre de una tabla y devuelva las primary y foreign key sobre la misma.
8. Crear un stored procedure que dada una tabla devuelva todas las tablas que tienen definida una foreign - key hacia ella.

Práctica 5 - System Catalog

Consulta de objetos del catálogo

1. Escribir un query que informe todos los objetos disponibles en la base de datos
2. Escribir un query para obtener todos los objetos creados luego del primero de mes
3. Escribir un query que informe todas las tablas existentes en la BD
4. Escribir un query que informe los constraints de una tabla dada
5. Escribir un query que informe las columnas de una tabla dada
6. Escribir un query que informe las columnas que componen la clave primaria de una tabla dada

Obtener información del catálogo

7. Crear un stored procedure que reciba como parámetro el nombre de la tabla y retorne los atributos de la misma indicando su tipo
8. Modificar el stored procedure para que informe el nombre de la clave primaria de la tabla
9. Escribir un stored procedure que dado el nombre de una tabla indique el nombre y el tipo de los atributos que conforman la clave primaria en el orden correcto
10. Crear un stored procedure para obtener los parámetros y el código fuente de un procedimientos determinado

Armar instrucciones dinámicamente usando el catálogo

11. Escribir un stored procedure para que genere un listado de instrucciones de DROP TABLE para cada tabla de usuario existente. **NO EJECUTAR EL RESULTADO**.
12. Crear un stored procedure para realizar un SELECT * de todas las tablas de usuario de la BD
13. Crear un stored procedure que genere todos los permisos (GRANT) correspondientes de las tablas y de los stored procedures existentes para un usuario determinado.

Práctica 6 - Integración

Ejercicio 1:

- Crear una tabla de nombre audit con los siguientes campos: tabla ,operación, campo ,dato ,fecha ,usuario
- Crear un stored procedure de nombre PRauditoria con parámetros de entrada para tabla ,operación ,campo y dato y que realice la inserción de dichos datos en la tabla audit.

Ejercicio 2:

- Sobre la base Northwind, crear una vista de nombre O_ODView que este formada por los campos orderid , productid, quantity, unitprice, discontinued.
- Realizar una inserción de los siguientes datos en dicha vista: 10248,25,14,2.5,0. Que sucedió? Por que?
- Crear un trigger instead of Insert para dicha vista, de nombre TRIns_O_ODView con el objetivo de solucionar el problema anterior. Dentro del trigger llamar al procedimiento PRauditoria con los parámetros correspondientes (para esto conviene utilizar el system catalog). Repetir la inserción. Que sucedió? Por que? Explique detalladamente los momentos en que ocurre cada acción.

Ejercicio3:

- Crear un trigger for insert para la tabla [order details] de nombre TRIns_OD que verifique el stock de products para el producto insertado (si es mayor al pedido) y luego actualice el mismo a stock=stock-quantity. Dentro del trigger llamar al procedimiento PRauditoria con los parámetros correspondientes.
- Realizar la inserción de datos en la vista (tener en cuenta que se disparara el trigger anterior). Comentar la sucesión de eventos a partir de dicha acción. Comente la utilización de la tabla inserted.

Ejercicio 4:

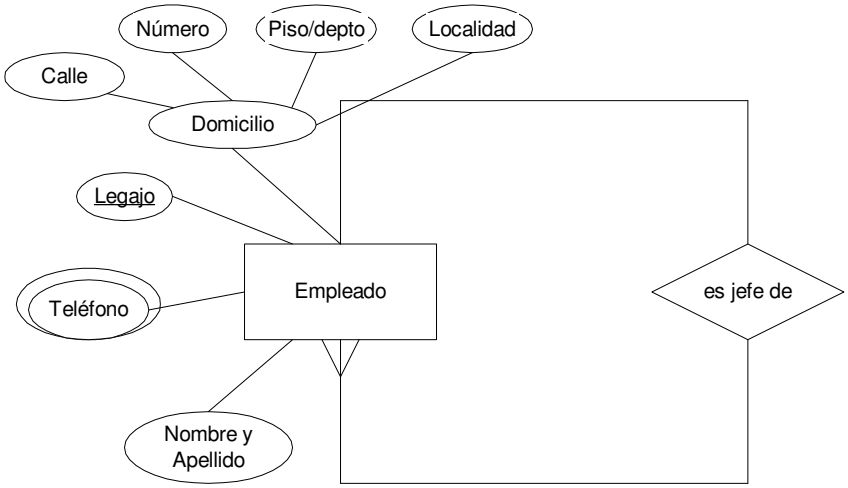
- Crear un trigger para update en la tabla products con el nombre TRUpd_Prod_Stock para el campo unitsinstock, que verifique que la nueva cantidad en stock (luego del update) sea superior a la suma de las cantidades pendientes de entrega en [order details] para el producto modificado. Dentro del trigger llamar al procedimiento PRauditoria con los parámetros correspondientes.
- Realizar una nueva inserción en la vista del ejercicio 1 con otros datos y quantity=400. Comentar lo sucedido
- Repetir el paso anterior pero con quantity=5. Comentar lo sucedido.
- Explique la utilización de las tablas inserted y deleted.
- Leer la información en la tabla audit. Mostrar y explicar los registros encontrados.

Conclusiones.:
Explique lo que sucedió al realizar el ejercicio nro 5, que triggers se dispararon? Como? Antes o después de que evento?

Práctica 7- Extensiones para Objetos

Elegir un motor de base de datos y :

1) Implementar el siguiente modelo de datos utilizando las extensiones para manejo de objetos del mencionado motor:



2) Escribir las instrucciones SQL que permiten resolver las siguientes situaciones:

a) Agregar un empleado especificando su domicilio
b) Agregar un teléfono para un empleado
c) Modificar la localidad de la dirección de un empleado
d) Modificar un número de teléfono asociado a un empleado
e) Borrar un número de teléfono de un empleado
f) Crear un registro de un empleado igual a otro (tener en cuenta que es también es necesario copiar los teléfonos y las direcciones)
g) Controlar si existen dos empleados con igual número de teléfono
h) Dado el legajo de un empleado mostrar sus datos personales y todos los números de teléfono que tiene asociado

Practica 8 – Transacciones

Definición de transacciones. Propiedades de las transacciones. Schedules y recuperación. Seriabilidad de los schedules.

1. Dadas las siguientes transacciones

T1 = {r1(A), r1(B), r1(C), w1(C), w1(A), w1(B),c1}
T2 = {r2(C), r2(B), w2(C), w2(B), c2}

   a) Construir el Schedule serial correspondiente
   b) Construir 2 schedules serializables aunque no seriales
   c) Construir 1 schedule no serializable e indicar porque no lo es
   d) Construir 1 schedule que evite el rollback en cascada
   e) Construir 1 schedule que no evite el rollback en cascada

2. Dadas las siguientes transacciones

T1 = {r1(C), r1(A), w1(C), w1(A), c1}
T2 = {r2(C), r2(B), w2(C), w2(B), c2}

   a) Construir el Schedule serial correspondiente
   b) Construir 2 schedules serializables aunque no seriales
   c) Construir 1 schedule no serializable e indicar porque no lo es
   d) Construir 1 schedule que evite el rollback en cascada
   e) Construir 1 schedule que no evite el rollback en cascada

3. Dadas las transacciones T1 y T2 y los schdules S1 y S2. Dibujar los grafos de seriabilidad de S1 y S2 e indicar si los mismos son o no serializables.

T1 = { r1(D), r1(E), w1(D), w1(E), c1}
T2 = { r2(E), r2(D), w2(E), w2(D), c2}
S1 = { r1(D), r1(E), r2(E), r2(D), w1(D), w1(E),c1, w2(E), w2(D), c2 }
S2 = { r1(D), r1(E), w1(D), w1(E), r2(E), r2(D), w2(E), w2(D),c1,c2}

4. Dadas las transacciones T1, T2 y T3 y los schdules S1 y S2. Dibujar los grafos de seriabilidad de S1 y S2 e indicar si los mismos son o no serializables.

T1 = { r1(D), r1(E), w1(D), c1}
T2 = { r2(F), r2(D), w2(F), c2}
T3 = { r3(D), r3(F), w3(D), c3}

S1 = { r1(D), r1(E), r2(F), w1(D), r2(D), w2(F), r3(D), r3(F), w3(D), c1, c2, c3}
S2 = { r1(D), r1(E), r2(F), r2(D), r3(D), r3(F),w1(D), w2(F), w3(D), c1, c2, c3}

5. Tomando como base las transacciones del ejercicio anterior
   a. Construir el Schedule serial correspondiente
   b. Construir 2 schedules serializables aunque no seriales
   c. Construir 1 schedule no serializable e indicar porque no lo es
   d. Construir 1 schedule que evite el rollback en cascada
   e. Construir 1 schedule que no evite el rollback en cascada

Práctica 9 - XML

Dadas las siguientes tablas de una Base de Datos de una empresa Emisora de Tarjetas de Crédito :

CUENTA	**TARJETA**	**DEUDA_CUENTA**	**BANCO**
(**PK**) **Nro_Cuenta** numeric(10,0)	(**PK**) **Nro_Tarjeta** numeric(16,0)	(**PK**) **Anio** char(4)	(**PK**) **Cod_Banco** int
Nombre_Apellido varchar(40)	**Nro_Cuenta** numeric(10,0) (FK - CUENTA)	(**PK**) **Mes** char(2)	**Nombre** varchar(30)
Ciclo int check(in(4,3,2,1))	**Cod_Categoria_Tarjeta** int check(in(0,1))	(**PK**) **Nro_Cuenta** numeric(10,0) (FK - CUENTA)	
Cod_Banco int (FK – BANCO)	**Cod_Banco** int (FK – BANCO)	**Deuda_Impaga** numeric(8,2)	
Cod_Producto int (FK - PRODUCTO)	**Cod_Estado** int (FK – ESTADO)		
Cod_Estado int (FK – ESTADO)	**Cod_Inhabilitacion** int NULL		
Limite_Compra numeric(8,2)			
Fecha_Alta datetime			

PRODUCTO	**ESTADO**	**RUBRO**	**COMERCIO**
(**PK**) **Cod_Producto** int	(**PK**) **Cod_Estado** int	(**PK**) **Cod_rubro** int	(**PK**) **Nro_Comercio** numeric(10,0)
Descripcion varchar(20)	**Descripcion** varchar(20)	**Descripcion** varchar(20)	**Nombre_Comercio** varchar(50)
			Cod_Rubro int (FK – RUBRO)

MOVIMIENTO
(**PK**) **Nro_Tarjeta** numeric(16,0) (FK – TARJETA)
(**PK**) **Nro_Cuenta** numeric(10,0) (FK – CUENTA)
(**PK**) **Nro_Cupon** int
(**PK**) **Fecha_Movimiento** datetime
(**PK**) **Cantidad_Cuotas** int check(between 1 and 12)
(**PK**) **Cod_Tipo_Operación** int (FK – TIPO_OPERACION)
(**PK**) **Nro_Comercio** numeric(10,0) (FK – COMERCIO)
Importe numeric(7,2)

El cod_tipo_operacion tiene la siguiente codificación: 1 : retiro en efectivo, 2 pago, 5 consumo

6. Escribir un stored procedure que permita exportar los consumos de los clientes de cada banco para un rango de fechas indicadas. El stored procedure recibe como parámetro el rango de fechas y un string que representa un directorio y debe generar un archivo XML para cada banco. El nombre del archivo tiene el siguiente formato: codbanco_fdesde_fhasta.xml. El archivo de exportación debe seguir la estructura del siguiente ejemplo. También definir el .xsd correspondiente

< Banco id = >
<Numero cuenta id = >
<Numero tarjeta id = >
<Movimientos>
<Movimiento fecha = .. Nombre_comercio= cuotas= importe=/>
</Movimientos>
</Numero tarjeta>
</Numero cuenta >
<Totales registros = ….. Importe = ……>
</Banco>

7. Diariamente recibimos de los bancos la información sobre los pagos que efectuaron nuestros clientes. Escribir un stored procedure que permita incorporar la información mencionada en la tabla de movimientos. El stored procedure debe verificar como mínimo que la cuenta para la que se informa el pago exista, no esté dada de baja y corresponda al banco que remitió el pago. El procedimiento recibe como entrada el nombre del archivo que tiene que procesar que tiene el siguiente formato CodBanco_fecha. Un ejemplo del formato del archivo es el siguiente:

```
<Registros>
- <R1s>
- <R1>
  <Numero_de_cuenta>6</Numero_de_cuenta>
  <Fecha_movimiento>20110605</Fecha_movimiento>
  <Importe>500000</Importe>
    </R1>
- <R1>
  <Numero_de_cuenta>3</Numero_de_cuenta>
  <Fecha_movimiento>20110605</Fecha_movimiento>
  <Importe>200000</Importe>
    </R1>
    </R1s>
- <R2>
  <Total_de_movimientos>2</Total_de_movimientos>

     <Importe_Total_movimientos>400000</Importe_Total_movimient
     os>
     </R2>
     </Registros>
```

REGISTRO DE TIPO 1

Campo	Tipo	Longitud
Tipo Registro	char	1

Numero de cuenta	numerico	10
Fecha Movimiento	Char	8 (yyyymmdd)
Importe	numerico	9 (los últimos 2 se asumirán como decimales)

REGISTRO DE TIPO 2

Campo	Tipo	Longitud
Tipo Registro	char	1
Total de movimientos	numerico	10
Importe Total Movimientos	numerico	16
Filler	char	1

Además debe generar un archivo llamado CodBanco_fecha.xml.err con los pagos que fueron rechazados. El archivo debe tener el número de cuenta y el código de error del rechazo:

01 Cuenta inexistente
02 Cuenta dada de baja
03 Cuenta correspondiente a otro banco

8. Programar los stored procedures de los puntos 1) y 2) para que sean ejecutados todos los días a las 21:00 hs.

9. Crear una tabla que contenga un campo de tipo XML y cargar en el mismo el contenido del archivo libros.xml

10. Resolver en SQL las siguientes operaciones sobre la tabla creada en el punto 5.

    a. Recuperar el id y el título de todos los libros
    b. Recuperar el título de todos los libres cuyo precio unitario sea mayor a 40
    c. Recuperar el autor de los libros del género "**Computer"**
    d. Recuperar la cantidad de libros por género y su precio promedio
    e. Modificar el precio del libro cuyo id es bk102 a 10.
    f. Agregar un nuevo libro al XML existente
    g. Borrar un el libro cuyo id es bk101 del XML

11. Escribir las sentencias Xquery que permiten resolver el ejercicio 6.

12. Definir un .xsd que refleje el esquema del ejercicio 1 de la práctica 7 a los fines de poder intercambiar esos datos con otra empresa. El formato de los datos debe definirse utilizando el "XML Naming and Design Rules", que es un standard de las

Naciones Unidas, donde sea posible se deberán usar los tipos básicos definidos en el mismo.

Observaciones
En la página de la materia hay un directorio con archivos de prueba para los ejercicios referidos a la importación de datos, tanto desde un archivo plano como desde un .xml

Anexo I: SQL Server 2000

a) Nomenclatura del SQL-92-standard names

Observaciones

El MS SQL Server 2000, implementa una serie de vistas que son compatibles con el Standard SQL-92 y que además independiza de los cambios internos que puedan sufrir las tablas del catálogo del sistema. Estas vistas son almacenadas en cada BD bajo el nombre de INFORMATION_SCHEMA.

Objetos a usar

INFORMATION_SCHEMA.TABLES, INFORMATION_SCHEMA.TABLE_CONSTRAINTS, INFORMATION_SCHEMA.COLUMNS, INFORMATION_SCHEMA.KEY_COLUMN_USAGE

Ejercicios

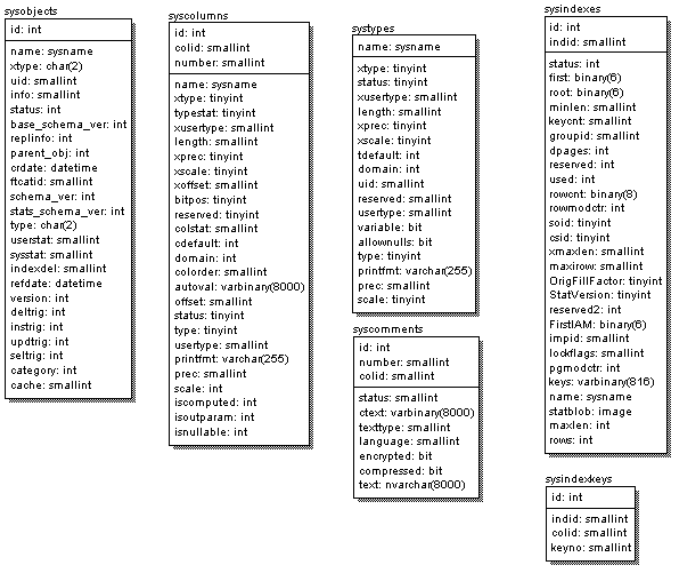
1. Escribir un query que informe todas las tablas existentes en la BD
2. Escribir un query que informe los constraints de una tabla dada
3. Escribir un query que informe las columnas de una tabla dada
4. Escribir un query que informe las columnas que componen la clave primaria de una tabla dada

b) Nomenclatura SQL Server nativo
Objetos a usar

Tablas: SYSOBJECTS, SYSCOLUMNS, SYSTYPES, SYSCOMMENTS, SYSINDEXES, SYSINDEXKEYS

Funciones: user_name(), object_name(), stats_date(), object_id()

Instrucciones: UPDATE STATISTICS Tabla, DBCC SHOW_STATISTICS (Tabla, Indice)



Ejercicios

1. Observar la estructura de las tablas SYSOBJECTS, SYSCOLUMNS, SYSTYPES, SYSCOMMENTS, SYSINDEXES, SYSINDEXKEYS, ver los datos almacenados, determinar que información se almacena en cada una.
2. Verificar que representan los atributos Uid, Type, crDate, parent_Obj en la tabla SYSOBJECTS
3. Que información almacena la tabla SYSCOLUMNS en sus atributos Id, xType, Length, isNullable
4. Informar todos los objetos disponibles en la base de datos y usando la función username() identificar los propietario de los objetos, y usando la función object_name() identificar el nombre de los objetos de los que dependen.
5. Verificar cuando fueron actualizadas las estadísticas por última vez para determinado índice de la tabla *Customers*
6. Actualizar las estadísticas de la tabla *Customers*
7. Ver las estadísticas para determinado índice de la tabla[1] *Authors*
8. Verificar que representan los atributos dpages, rowmodctr, pgmodctr en la tabla SYSINDEXES