UNIVERSIDAD ORT Uruguay

Facultad de Ingeniería
Bernard Wand-Polak
Cuareim 1451
11.100 Montevideo, Uruguay
Tel 902 15 05 Fax 908 13 70

EVALUACIÓN	Obligatorio	**GRUPOS**		**FECHA**	Agosto 2020
MATERIA	Bases de Datos 2				
CARRERA	Analista en Tecnologías de Información / Analista Programador				
CONDICIONES	**- Entrega:** 19/NOV **- Puntos:** Máximo: 40. Mínimo: 0. **IMPORTANTE** - Los grupos deben estar conformados por hasta un máximo de dos personas. - Inscribirse (sacar la "*boleta de entrega*").				

En una empresa importante del mercado se llevan controles de seguridad tecnológica sobre la red informática, a continuación se presentan las tablas que son usadas para tales efectos.

ZONAS (ZonaId, ZonaNom, ZonaDescrip)

Registra las Zonas que conforman la Red de la Empresa. Las zonas se identifican por un Numero. El nombre de las zonas es un dato obligatorio y no existen dos zonas con igual nombre.
Al momento de ingresar los datos suponer que hay una zona por cada área de la Empresa, y una zona llamada DMZ donde están los equipos que conforman la Red Perimetral.

USUARIOS (Usuario, UsuPsw, UsuNomApp, UsuMail)

Registra los Usuarios de Red de la Empresa los cuales se identifican mediante el campo *usuario* que es de hasta 50 caracteres. También se registra la clave *usuPsw*, dato obligatorio, el cual debe tener un máximo de 100 caracteres, y deben contener por lo menos un digito.
Finalmente se guardan, si se conocen, los datos de nombre y apellido *UsuNomApp*, así como el mail *UsuMail*.

EQUIPOS (EqpIP, EqpNom, EqpTipo, EqpSO, *ZonaId*)

Registra el equipamiento informático de la empresa, los cuales se identifican por su IP (*EqpIP*), la cual está formado por 4 secuencias de 3 dígitos, cada secuencia puede tomar valores entre 0 y 255, y están separadas por un punto, ejemplo 192.168.045.123
De los equipos se registra su nombre *EqpNom* que no puede repetirse, el tipo de equipo *EqpTipo* que puede tomar los valores: Terminal, Servidor, Tablet o Impresora, y la zona donde se encuentra *ZonaID*.

Los nombres de los equipos deben tener por prefijo:
WKS si son Terminales de Trabajo
SRV si son Servidores

IMP si son Impresoras
TBL si son Tablets
Todos los datos de esta tabla son requeridos excepto el Sistema Operativo

PERMISOSCNX (*Usuario*, *ZonaId*, Habilitado)

En esta tabla se indica las zonas a las que puede acceder cada usuario. El campo *Habilitado* puede tomar valores SI o NO según dicho permiso este habilitado o no, por defecto las conexiones no están habilitadas.

CTRLCONEXIONES (CnxId, *Usuario*, *EqpIP*, CnxFchHr, CnxPermitida, *TarID*)

En esta tabla se registran todos los intentos de conexión de los usuarios en los equipos. Cada intento de conexión se identifica por un auto numérico, y se guarda la fecha hora en que ocurrió *CnxFchHr*, dato obligatorio.
Si el usuario está intentando conectarse a una zona para la cual tiene permiso entonces se permite la conexión, si una conexión fue permitida o no se registra en el campo *CnxPermitida*
En el campo *TarID* indica la tarea que analizara el caso, en caso de ser necesario.

ESCANEOS (ScnHerr, ScnVulnNom, ScnDescrip)

En esta tabla se registran las herramientas usadas para realizar escaneos de vulnerabilidades y las vulnerabilidades que éstas son capaces de detectar. *scnHerr* es el nombre de la herramienta y *ScnVulnNom* es el nombre de vulnerabilidad que es capaz de detectar. También se guarda una descripción. Ejemplo (“McAfee”, “Virus”, “Escaneo para detección de virus”), (“Crowdstrike”, “malware”, “Escaneo para detección de malware”), etc.

CTRLVULNERABILIDADES (*ScnHerr*, *ScnVulnNom*, *ZonaId*, VulnFchScanO, VulnFchScanU, VulnCriticidad, *TarID*)

En esta tabla se registran las vulnerabilidades detectadas por los escaneos. Estas vulnerabilidades se identifican con la herramienta y el nombre de la vulnerabilidad, la zona en la que se detecto la vulnerabilidad y la fecha en que se detecto.

Una vulnerabilidad se mantiene abierta mientras siga siendo detectada registrándose en el campo *VulnFchHScanU* la fecha de ultimo escaneo que la detecto, y si hay una tarea que este analizando esta situación esta es indicada en el campo *TarID*
Una vez que una vulnerabilidad es resuelta, es decir la tarea que la estaba analizando la resolvió, si vuelve a aparecer se trata como una nueva vulnerabilidad, es decir un nuevo registro en esta tabla

Finalmente se registra la criticidad de cada vulnerabilidad detectada, la que puede tomar los valores: BAJA, MEDIA, ALTA

Facultad de Ingeniería
Bernard Wand-Polak
Cuareim 1451
11.100 Montevideo, Uruguay
Tel 902 15 05 Fax 908 13 70

TAREAS (TarId, TarEstado, TarHrsAcum, TarFchIni, TarFchFin, TarFchFPrev, TarDescrip)

En esta tabla se registran las Tareas que realiza el área de Tecnología. Las tareas se identifican por un autonumerico, y tienen un Estado que puede tomar valores: EN ESPERA, EN DESARROLLO, RESUELTO, CANCELADA. También se guarda la cantidad de horas de trabajo acumuladas en atención a cada tarea, así como la fecha de inicio, la fecha de finalización y la fecha de finalización prevista.
Finalmente, todas las tareas tienen una descripción.
Excepto el campo fecha de finalización todos los demás son obligatorios.
Debe controlarse que la fecha de inicio no sea mayor a la fecha prevista de finalización
Se considera que la cantidad de horas de trabajo acumuladas debe ser mayor a 1 hr, debido que la sola gestión inicial lleva por lo menos ese tiempo.

RACI (*RaciTarId*, *RaciUsuario*, RaciRol)

En esta tabla se guardan los usuarios vinculados a cada tarea y el rol que tienen en dicha tarea. Los Roles posibles son: Responsable de Ejecución (R), Administrador (A), Consultor (C) aporta información requerida para realizar la tarea, y finalmente Informativo persona que debe estar informada del cambio (I)

Se pide

1. Crear las restricciones de integridad que surjan del análisis de la letra, sobre el script de creación de tablas proporcionado. **(5 puntos)**

2. Especificar las restricciones de integridad que surjan de la letra, que no pueden ser implementadas con PK, FK, Check, Unique, NOT NULL. **(1 punto)**

3. Creación de índices que considere puedan ser útiles para optimizar las consultas (según criterio establecido en el curso). **(2 puntos)**

4. Ingreso de un juego completo de datos de prueba (será más valorada la calidad de los datos más que la cantidad. El mismo debería incluir ejemplos que deban ser rechazados por no cumplir con las restricciones implementadas. **(2 puntos)**

5. Resolver mediante consultas SQL **(6 puntos)**:

a) Mostrar los datos de las ultimas Vulnerabilidades de criticidad ALTA que no hayan sido resueltas aún. En el resultado debe aparecer también el nombre de la zona en la cual se detectó la Vulnerabilidad

b) Mostrar los datos de los usuarios que pueden acceder a todas las zonas.

Facultad de Ingeniería
Bernard Wand-Polak
Cuareim 1451
11.100 Montevideo, Uruguay
Tel 902 15 05 Fax 908 13 70

c) Mostrar los datos de las zonas mas seguras de la red, siendo estas aquellas que no han tenido vulnerabilidades ALTA los últimos tres meses, y que tienen menos de 3 usuarios con conexiones no permitidas en el último mes

d) Se quiere los usuarios que hace mas de 180 dias que no se conectan. En el resultado debe aparecer la cantidad de días que hace que no se conectan y el nombre del equipo al que se conectó por última vez

e) Para cada usuario que es responsable de mas de 3 tareas no resueltas, mostrar el usuario y el promedio de horas que están insumiendo estas tareas

f) Para cada Zona de la Red indicar la cantidad de conexiones no permitidas a equipos de la zona, y la cantidad de vulnerabilidades encontradas en la zona en los últimos 30 días. Usar la función 6b) en la solución implementada

6. Crear procedimientos o funciones según corresponda **(10 puntos)**:

a. Crear una función o procedimiento que dado un usuario y un equipo, devuelva una indicación de si el usuario tiene permiso de conexión a ese equipo según la zona en la que se encuentra este último.

b. Crear una función que dada una zona y una cantidad de días X, devuelva la cantidad de vulnerabilidades encontradas en dicha zona en los últimos X días indicados por los parámetros

c. Crear una función que dada una herramienta de escaneo devuelva el nombre de la zona con más vulnerabilidades criticas, altas, encontradas por escaneos realizados por dicha herramienta. Si hay más de una zona en dichas condiciones devolver la que tenga la vulnerabilidad más reciente

d. Crear un procedimiento almacenado, que reciba por parámetro un usuario y un rol y devuelva la cantidad de tareas sin resolver en las cuales dicho usuario tiene el rol indicado, también devolver cuantas de esas tareas están atrasadas, es decir ya debieron haber finalizado según lo previsto

e. Crear un procedimiento almacenado que dado el nombre de una vulnerabilidad devolver la cantidad de veces que se detectó esta vulnerabilidad, la primer y ultima ves que se detectó, el promedio de horas dedicadas a tareas relacionadas a esta vulnerabilidad, y si alguna de esas tareas se resolvió.

Facultad de Ingeniería
Bernard Wand-Polak
Cuareim 1451
11.100 Montevideo, Uruguay
Tel 902 15 05 Fax 908 13 70

7. Crear disparadores necesarios para realizar las siguientes acciones **(10 puntos)**:

a. Mediante un disparador, controle que solo se puedan insertar de a un equipo por vez

   Y cuando esto ocurra controle que se cumplan las restricciones que debe cumplir esta tabla y que no pudieron ser implementadas en el créate de la misma

   (este disparador debe tener en cuenta inserciones SIMPLES)

b. Crear un disparador que cada vez que se ingrese un registro a la tabla de Control de Conexiones determine los valores que corresponden para los campos fecha-hora y conexión permitida. Usar el procedimiento o función implementado en el punto 6a)

   (este disparador debe tener en cuenta inserciones MULTIPLES)

c. Crear un disparador que cada vez que se ingresen controles de vulnerabilidades, resultados de escaneos, haga lo siguiente

   - Si esa vulnerabilidad esta abierta, actualice la fecha de ultima vez que fue encontrada la vulnerabilidad con la fecha del dia.
   - Si esa vulnerabilidad no esta abierta agregue el registro (inicialmente sin tarea asociada)

   Se considera que una vulnerabilidad esta abierta si dicha herramienta ya la detecto en la misma zona, y no tiene tarea asociada o tiene una tarea que no esta RESUELTA

   (este disparador debe tener en cuenta modificaciones SIMPLES)

d. Crear un disparador que controle que solo se puedan eliminar control de conexiones que no tengan tareas asociadas y que sean de una antigüedad mayor a 1 año.

   (este disparador debe tener en cuenta eliminaciones MULTIPLES)

e. Crear un disparador que controle que si una tarea es marcada como CANCELADA deje en null los controles de conexiones o vulnerabilidades que tenga asociados.

   (este disparador debe tener en cuenta modificaciones MULTIPLES)

Facultad de Ingeniería
Bernard Wand-Polak
Cuareim 1451
11.100 Montevideo, Uruguay
Tel 902 15 05 Fax 908 13 70

8. Implementar las siguientes Vistas **(4 puntos)**:

a. Crear una vista que muestre para cada herramienta de escaneo mostrar la cantidad de vulnerabilidades que ha detectado de criticidad alta, media y baja en el mes actual. En el resultado de deben aparecer todas las herramientas de escaneo

b. Crear una vista que muestre los datos de los usuarios (usuario, nombre) más críticos siendo estos los que son responsables de ejecutar más cantidad de tareas que no están resueltas ni en espera. En el resultado también debe aparecer el promedio de horas dedicadas a esas tareas.

Se debe entregar:

a. Script: *Restricciones.sql* con las restricciones de integridad creadas sobre el script de creación de tablas, índices
b. Script: *Datos.sql* el ingreso de datos de prueba.
c. Script: *Datos_NoValidos.sql* el ingreso de datos de prueba de control de restricciones de integridad.
d. Archivo con especificación de restricciones de integridad del punto 2.
e. Script: *Consultas_Vistas.sql* con las consultas y vistas pedidas
f. Script: *Procedimientos.sql* con los procedimientos almacenados
g. Script: *Funciones.sql* con las funciones solicitadas
h. Script: *Disparadores.sql* con el código de cada disparador implementado

Consideraciones generales:

1. Los docentes de la materia cumplirán el rol de usuario final del producto a los efectos de evacuar las dudas que puedan surgir a los estudiantes en detalles que no estén incluidos explícitamente en la letra. Independientemente de esto, los alumnos podrán investigar sobre sistemas existentes, así como aportes basados en su propia experiencia o relevamiento con terceros para enriquecer la solución a los problemas planteados siempre que no contradiga lo explicitado en la letra. Cualquier agregado deberá documentarse claramente en la solución y será considerado positivamente en la evaluación. Modificaciones de la letra que puedan surgir durante el curso, serán publicadas en aulas y deberán considerarse en la entrega final.

2. La corrección del obligatorio se hará en base a la estructura entregada junto con la letra del mismo, por lo que los puntos desarrollados deben ser testeados sobre esta estructura. Soluciones a los puntos del obligatorio que no ejecuten correctamente sobre la estructura proporcionada serán evaluados como incorrectos.

3. Durante la última semana los docentes no contestarán dudas del Obligatorio por ningún medio. Esta consideración intenta evitar que los alumnos dejen la implementación del obligatorio para último momento. Se insta a los estudiantes a desarrollar el obligatorio durante el transcurso del semestre para entregar un trabajo de calidad.

Anexo Script

```
Create Database OBLSem2
GO

Use OBLSem2
GO


Create Table TAREAS (
       TarId int identity(1,1)    not null,
       TarEstado varchar(15) ,
       TarHrsAcum int ,
       TarFchIni date ,
       TarFchFin date,
       TarFchFPrev date,
       TarDescrip varchar(200) )
GO

Create Table ZONAS (
       ZonaId int not null,
       ZonaNom varchar(50) ,
       ZonaDescrip varchar(100)
       )
GO

Create Table USUARIOS (
       Usuario varchar(50) not null,
       UsuPsw varchar(200) ,
       UsuNomApp varchar(200),
       UsuMail varchar(200)
       )
GO

Create Table EQUIPOS (
       EqpIP char(15) not null,
       EqpNom varchar(50) ,
       EqpTipo varchar(10),
       EqpSO  varchar(10),
       ZonaId int
       )
GO

Create Table PERMISOSCNX (
       Usuario varchar(50) ,
       ZonaId int ,
       Habilitado char(2),
       )
GO
```

```
Create Table CTRLCONEXIONES (
       CnxId int identity(1,1) not null,
       Usuario varchar(50) ,
       EqpIP char(15),
       CnxFchHr datetime ,
       CnxPermitida bit,
       TarID int)
GO

Create Table ESCANEOS (
       ScnHerr varchar(100),
       ScnVulnNom varchar(100),
       ScnDescrip varchar(200),
       )
GO

create table CTRLVULNERABILIDADES (
       ScnHerr varchar(100),
       ScnVulnNom varchar(100),
       ZonaId int ,
       VulnFchScanO date,
       VulnFchScanU date,
       VulnCriticidad varchar(5),
       TarID int
       )
GO

Create Table RACI (
       RaciTarId int,
       RaciUsuario varchar(50) ,
       RaciRol char(1),
       )
GO
```