Bases de Datos III

Práctica de Laboratorio

Gonzalo Senovilla Minguela, Miguel Vítores Vicente

UEMC-3º Ingeniería Informática

Bases de Datos II

Índice

Introducción

Enunciado

Los ayuntamientos de los municipios de la comunidad autónoma de Castilla y León desean mantener información actualizada de las viviendas ubicadas en zonas urbanas. Se desea diseñar un sistema de bases de datos que incluya las características de las viviendas, su ubicación, propietarios, personas que las habitan, etc. Esta información se utilizará con fines administrativos (impuestos y otros) y estadísticos.

A finales de año, el ayuntamiento de cada municipio debe cobrar a cada propietario un impuesto por las viviendas que son de su propiedad en la actualidad. Así, emite un recibo para cada vivienda donde figura el número de registro catastral, la dirección donde se ubica la vivienda (calle, número y piso), el número de metros cuadrados y el dni y nombre del propietario (aunque la propiedad de una vivienda puede ser compartida por varias personas, a efectos de cobros de impuestos consideramos sólo uno de ellos), además del importe del impuesto y el intervalo de fechas en las que debe ser pagado en primera instancia. Este recibo se le remitirá a la dirección del propietario, que por supuesto, no tiene por qué coincidir con la de la vivienda de la que debe pagar el impuesto.

El importe del impuesto, de cada vivienda, depende de múltiples factores que deben considerarse en su cálculo. Entre ellos está el municipio y el barrio o zona urbana donde se ubica la vivienda, los m2 de la vivienda, y el precio de tasación de la vivienda. A estos efectos cada provincia consta de una serie de municipios, de los que hay que mantener su nombre, el área y la provincia a la que pertenecen; además para identificar a cada municipio se utiliza un código único a nivel regional. Y a su vez cada municipio está dividido en una serie de barrios o zonas urbanas claramente delimitadas. A la hora de calcular el impuesto debe usarse el precio medio del m2 en la zona urbana.

El propietario puede realizar el pago del impuesto en efectivo, dirigiéndose al ayuntamiento, una vez que le ha llegado la notificación de que debe pagar. Si el propietario ha excedido la fecha de vencimiento del impuesto se le aplica un 25% sobre el importe total.

Por otra parte, cada cierto tiempo desde la Junta de Castilla y León se solicitan una serie de informes destinados a distintas consejerías.

La consejería de vivienda y urbanismo suele solicitar:

> *1.- Una lista de todas las zonas urbanas indicando el precio medio del m2, su nombre, área, coordenadas geográficas y el municipio y provincia al que pertenecen, ordenados por provincia y municipio.*
>
> *2.- El mismo listado ordenado por el precio medio del m2 en cada zona urbana.*
>
> *3.- Un listado con las viviendas vacías en cada zona urbana y municipio.*

La consejería de bienestar social solicita:

1.- Un listado de las viviendas habitadas por una única persona de más de 70 años y los datos del barrio donde se ubican.

La consejería de economía y hacienda solicita:

1.- Un listado de los propietarios que no están al corriente de pagos de impuestos (morosos) por zona urbana y municipio.

Problemática a resolver

Las reladatos realizada el cuatrimestre anterior, enfocando nuestros esfuerzos en optimizarla, en implementar disparadores y rutinas, y añadiremos usuarios como ejemplo de un uso común de los sistemas gestores.

Además, y de manera teórica, trataremos ciertos temas como la apropiada indexación para optimizar las consultas o la seguridad en las bases de datos, tanto ciertas medidas que habría que aplicar, como los problemas asociados al entorno web.

Optimización

Índices creados

Como fruto de la práctica del cuatrimestre pasado ya creamos ciertos índices, diferentes unos de otros en estructura, pero todos ellos claves primarias de sus respectivas tablas. Así, tenemos cadenas de caracteres de longitud variable como claves primarias en algunas tablas, en relacionadas con personas, como Ocupantes o Propietarios, tenemos como índice unívoco de cada persona un dni, cadena de caracteres de longitud fija, lo que teóricamente supone una optimización.

Por último, teníamos en la tabla Viviendas como identificador el número de catastro, lo que consideremos que es correcto, ya que es una cadena de caracteres que identifica de forma oficial una vivienda, pero es una cadena de caracteres de un tamaño considerable, lo que puede influir en las consultas, haciéndolas más lentas e ineficaces, sobre todo teniendo en cuenta que es clave foránea en Impuestos y en Ocupantes. Para solucionar esto decidimos indexar esta tabla nuevamente, pero esta vez con números enteros autoincrementales y únicos, dejando aun así el número de catastro como una columna de viviendas por la que se pueden seguir realizando búsquedas, pero al realizar un join de alguna tabla, como Impuestos u Ocupantes con Viviendas, el sistema gestor hará uso de la nueva clave primaria que es un número entero.

Cabe añadir que para la creación de un trigger que creara tuplas en la tabla Impuestos como y por facilidad decidimos cambiar el la clave primaria que identificaba cada impuesto de una cadena de caracteres de longitud variable, cosa que no es sencilla de autogenerar en el disparador, a una clave entera autoincremental, simplificando la creación de nuevos impuestos.

Justificación uso índices

EL uso de índices para nuestras tablas tiene principalmente una función, ya no solo que podamos identificar unívocamente cada tabla simplemente fijándonos en el valor de su primera columna, sino que supone una agilización de las consultas realizadas con aquella tabla que esté indexada.

Programación de la Base de Datos

Triggers

Enunciado en formato textual

1. Al actualizar un barrio, si el barrio crece, entonces el precio del metro cuadrado aumenta. Así, si el área de dicho barrio aumenta en 200 o más metros cuadrados, entonces el precio medio del metro cuadrado aumenta un 6%.
2.
3. Si se paga un impuesto fuera de plazo, entonces se multa al propietario, creando un nuevo impuesto para este.
4. Al insertar una nueva vivienda, si está en un barrio con un área menor que 40m2, su precio de tasación disminuye.
5. Al actualizar los m2 de una vivienda, el precio de tasación se modifica en función de la diferencia de m2.

Código SQL asociado

1.

```
/*1. Si el barrio crece el precio del m2 aumenta*/
delimiter //
CREATE TRIGGER valBarrio BEFORE UPDATE ON Barrios
FOR EACH ROW
BEGIN
    IF  NEW.area - OLD.area > 200 THEN
        SET NEW.avgM2price = OLD.avgM2price * 1.06;
    END IF;

END//
```

2.
3.
4.

```
delimiter //
CREATE TRIGGER depreciacionVivienda BEFORE INSERT ON Viviendas
FOR EACH ROW
BEGIN
    declare areaBarrio decimal(10,3);
    select area into areaBarrio
    from Barrios b
    where b.idBarrios = new.idBarrios;
    if(areaBarrio < 40) then
        set new.precioTasacion = new.precioTasacion - ( new.precioTasacion * 0.05 );
    end if;
END//
```

5.

```
delimiter //
CREATE TRIGGER cambioMetrosCuadrados BEFORE UPDATE ON Viviendas
FOR EACH ROW
BEGIN
    if( new.m2 > 0 ) then
        set new.precioTasacion = new.precioTasacion * new.m2 / old.m2;
    end if;
END//
```

Resultados

1. Aumentaremos el área de un barrio en concreto en 250 metros cuadrados para

```
select * from Barrios where idBarrios=7504;
update Barrios set area = area + 250 where idBarrios=7504;
```

comprobar que el precio medio del metro cuadrado se incrementa.

Aquí tenemos esa tupla antes de ser actualizada

idBarrios	nombre	area	coord	zipCode	avgM2price	idMunicipio
7504	Rondilla	49.360	7.71844,42.85924	47291	1656.899	40200

Y aquí después.

idBarrios	nombre	area	coord	zipCode	avgM2price	idMunicipio
7504	Rondilla	299.360	7.71844,42.85924	47291	1756.313	40200

2.
3.
4. Insertaremos una vivienda en un barrio con menos de 40m2.

idBarrios	nombre	area	coord	zipCode	avgM2price	idMunicipio
7536	Cristo del Mercado	9.861	39.32441,-35.78522	40248	1592.886	40300

El precio de tasación era de 200000€ pero se reduce un 5% debido a lo anterior.

```
select * from barrios where idBarrios=7536;
INSERT INTO `AdminViviendas`.`Viviendas` (`nºCatastro`, `calle`, `num`, `piso`, `m2`, `precioTasacion`, `idBarrios`, `dni`)
values ('2688754 CY7466H 0666 IO', 'Avda. Ejemplo', 1, '1 A', 200, 200000, '7536', '64105041Q');
```

18	2688754 CY7466H 0666 IO	Avda. Ejemplo	1	1 A	200.000	190000.000	7536	64105041Q

5. Tenemos esta vivienda con 200m2, si cambiamos ese valor, se modifica también su precio de tasación.

idViviendas	nºCatastro	calle	num	piso	m2	precioTasacion	idBarrios	dni
1	9872023 VH5797S 0001 WX	Calle Luna	7	1 A	200.000	200000.000	7500	73793158G

idViviendas	nºCatastro	calle	num	piso	m2	precioTasacion	idBarrios	dni
1	9872023 VH5797S 0001 WX	Calle Luna	7	1 A	210.000	210000.000	7500	73793158G

Procedimientos

Enunciado en formato textual

1. Calcula el número de impuestos sin pagar vinculados a un dni
2. Listar casas de un barrio
3. Subir los impuestos a casas con más de X m^2
4. Encuentra las viviendas que pertenecen a un nombre y apellidos
5. Aumenta en un porcentaje el precio de tasación de las viviendas ubicadas en un municipio
6. Un listado de los habitantes de un municipio pasado por argumentos con nombre, dni y apellidos como atributos a mostrar y ordenados alfabéticamente.

Código SQL asociado

1.

```
-- 1. Calcula el numero de impuestos sin pagar vinculados a un dni
delimiter //
CREATE PROCEDURE numImpuestosDeuda(iN elDni VARCHAR(15) , OUT total integer)
begin
    SELECT COUNT(*) into total FROM Impuestos i, Propietarios p WHERE i.dni = p.dni AND p.dni = elDni AND i.fechaActualPago = null;
end//
```

2.

```
-- 2. Listar casas de un barrio
delimiter //
CREATE PROCEDURE listarCasas(iN  barrio VARCHAR(45))
begin
    SELECT v.nºCatastro, v.calle, v.num, v.piso, v.m2, v.precioTasacion, v.dni, p.nombre, p.apellidos
    FROM Viviendas v, Barrios b, Propietarios p
    WHERE b.nombre = barrio AND v.idBarrios = b.idBarrios AND v.dni = p.dni;
end//
```

3.

```
-- 3. Subir los impuestos a casas con mas de X m^2
delimiter //
CREATE PROCEDURE subirImpuestosM2(iN incremento DECIMAL(10,3), IN tamaño DECIMAL(10,3))
begin
    UPDATE Impuestos i, Viviendas v SET i.importe = (i.importe + incremento)
    WHERE i.idViviendas = v.idViviendas AND v.M2 >= tamaño;
end//
```

4.

```
-- 4. Encuentra las viviendas que pertenecen a un nombre y apellidos
delimiter //
CREATE PROCEDURE encuentraViviendas(iN nombreProp VARCHAR(20), IN apellidosProp VARCHAR(40))
begin
    SELECT * FROM Viviendas v, Propietarios p WHERE v.dni = p.dni AND p.nombre = nombreProp AND p.apellidos = apellidosProp;
end//
```

5.

```
-- 5. Aumenta en un porcentaje el precio de tasación de las viviendas ubicadas en un municipio
delimiter //
CREATE PROCEDURE incPrecioViviendasMunic(IN municipio varchar(45), IN porcentaje decimal)
begin
    update Municipios m, Barrios b, Viviendas v
    set precioTasacion = precioTasacion + precioTasacion * porcentaje/100
    where m.nombre = municipio and m.idMunicipio = b.idMunicipio and v.idBarrios = b.idBarrios;
end//
```

6.

```
-- 6. Listado de habitantes de un municipio con nombre apellido dni y ordenados afabéticamente
DELIMITER //
CREATE PROCEDURE listarHabitantesMunicipio(IN muni VARCHAR(25))
BEGIN
    SELECT o.nombre as nombre, o.apellidos as apellidos, o.dni as dni FROM ocupantes o, Viviendas v, Municipios m, Barrios b
    WHERE v.idBarrios = b.idBarrios AND o.idViviendas = v.idViviendas AND b.idMunicipio = m.idMunicipio
    AND m.nombre = muni
    UNION
    SELECT p.nombre as nombre, p.apellidos as apellidos, p.dni as dni FROM propietarios p, Viviendas v, Municipios m, Barrios b
    WHERE v.idBarrios = b.idBarrios AND p.dni = v.dni AND b.idMunicipio = m.idMunicipio
    AND m.nombre = muni
    ORDER BY nombre;

END//
```

Resultados

1.

```
call numImpuestosDeuda('94328497T' , @total);

SELECT @total;
```

@total
0

2.

```
call listarCasas('Judería');
```

nºCatastro	calle	num	piso	m2	precioTasacion	dni	nombre	apellidos
7445852 HG2591T 0202 VN	Avda. Paco Casado	3	1 C	350.000	330000.000	59804933B	Juanjo	Casillas Sevilla

3.

```
select * from impuestos;
call subirImpuestosM2(20, 60);
select * from impuestos;
```

idImpuesto	fechaInico	fechaVencimiento	importe	fechaActualPago	dni	idViviendas
1	2018-10-17	2018-12-26	320.000	2018-11-02	73793158G	1
2	2018-08-11	2018-11-22	582.250	2018-10-15	73793158G	2
3	2018-09-19	2018-12-25	976.320	2018-11-12	73793158G	3
4	2018-08-25	2018-10-21	220.650	2018-09-05	20039032Y	4
5	2018-11-29	2019-01-17	221.900	2018-12-26	41679701K	5
6	2018-07-02	2018-09-12	382.150	2018-08-29	71297004V	6
7	2018-02-15	2018-05-13	273.020	2018-02-28	59804933B	7
8	2018-04-18	2018-06-26	446.332	2018-05-08	20039032Y	8
9	2018-01-30	2018-03-11	501.300	NULL	23310054Z	9
10	2018-10-22	2019-02-02	325.900	2018-02-01	73793158G	10
11	2018-12-15	2019-03-29	577.500	NULL	94328497T	12
15	2019-06-27	2019-08-27	520.990	NULL	59804933B	7
NULL	NULL	NULL	NULL	NULL	NULL	NULL

idImpuesto	fechaInico	fechaVencimiento	importe	fechaActualPago	dni	idViviendas
1	2018-10-17	2018-12-26	340.000	2018-11-02	73793158G	1
2	2018-08-11	2018-11-22	602.250	2018-10-15	73793158G	2
3	2018-09-19	2018-12-25	996.320	2018-11-12	73793158G	3
4	2018-08-25	2018-10-21	240.650	2018-09-05	20039032Y	4
5	2018-11-29	2019-01-17	241.900	2018-12-26	41679701K	5
6	2018-07-02	2018-09-12	402.150	2018-08-29	71297004V	6
7	2018-02-15	2018-05-13	293.020	2018-02-28	59804933B	7
8	2018-04-18	2018-06-26	466.332	2018-05-08	20039032Y	8
9	2018-01-30	2018-03-11	521.300	NULL	23310054Z	9
10	2018-10-22	2019-02-02	345.900	2018-02-01	73793158G	10
11	2018-12-15	2019-03-29	597.500	NULL	94328497T	12
15	2019-06-27	2019-08-27	540.990	NULL	59804933B	7
NULL	NULL	NULL	NULL	NULL	NULL	NULL

4.

```
call encuentraViviendas('Pedro','Ramiro López');
```

idViviendas	nºCatastro	calle	num	piso	m2	precioTasacion	idBarrios	dni	dni	nombre	apellidos	calle	num	piso
1	9872023 VH5797S 0001 WX	Calle Luna	7	1 A	200.000	200000.000	7500	73793158G	73793158G	Pedro	Ramiro López	Avenida Martin Lopez	5	3 F
2	9666525 RR5168T 8541 GG	Calle Lorenzo Gonzalez	2	2 A	60.000	120500.000	7501	73793158G	73793158G	Pedro	Ramiro López	Avenida Martin Lopez	5	3 F
3	9112515 SC5184F 9952 FR	Avda. Camilo Sexto	5	NULL	400.000	450000.000	7503	73793158G	73793158G	Pedro	Ramiro López	Avenida Martin Lopez	5	3 F
10	2255255 CC8858Y 0110 SA	Avda. Zapataria	11	3 B	162.000	126850.000	7509	73793158G	73793158G	Pedro	Ramiro López	Avenida Martin Lopez	5	3 F

5.

```
select * from viviendas;
select * from municipios;
select * from barrios;
call incPrecioViviendasMunic('Cuellar', 5);
```

idViviendas	nºCatastro	calle	num	piso	m2	precioTasacion	idBarrios	dni
1	9872023 VH5797S 0001 WX	Calle Luna	7	1 A	210.000	210000.000	7500	73793158G
2	9666525 RR5168T 8541 GG	Calle Lorenzo Gonzalez	2	2 A	60.000	126525.000	7501	73793158G
3	9112515 SC5184F 9952 FR	Avda. Camilo Sexto	5	NULL	400.000	450000.000	7503	73793158G
4	9001259 ZE7812S 2288 YY	Calle El Caldero	12	2 C	90.000	250000.000	7502	20039032Y
5	9110254 WS448O 3354 QW	Avda. Don Calero Martinez	10	2 B	95.000	220500.000	7505	41679701K
6	7445852 LP8524O 1119 ZX	Calle Sésamo Bravo	11	2 A	83.000	150000.000	7504	71297004V
7	7445852 HG2591T 0202 VN	Avda. Paco Casado	3	1 C	350.000	330000.000	7507	59804933B
8	8956211 VB6518F 0025 VV	Calle El Portal	15	NULL	523.000	452500.000	7509	20039032Y
9	1888254 CF4511R 9955 NB	Calle El Jorobado Danés	10	2 B	122.000	192300.000	7510	23310054Z
10	2255255 CC8858Y 0110 SA	Avda. Zapataria	11	3 B	162.000	126850.000	7509	73793158G
11	2255877 ZZ9523J 7751 MN	Avda. Punta Cuntera	5	5 A	63.000	115300.000	7505	44892211Y
12	9522624 YP4525P 0005 KK	Avda. La Sevillana Marena	2	1 D	80.000	175300.000	7506	94328497T
13	1388754 CX8466H 0666 PO	Avda. Palencia	1	1 A	107.000	107640.000	7510	64105041Q
14	2688754 CY7466H 0666 IO	Avda. Ejemplo	1	1 A	200.000	190000.000	7506	64105041Q
NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

idViviendas	nºCatastro	calle	num	piso	m2	precioTasacion	idBarrios	dni
1	9872023 VH5797S 0001 WX	Calle Luna	7	1 A	210.000	220500.000	7500	73793158G
2	9666525 RR5168T 8541 GG	Calle Lorenzo Gonzalez	2	2 A	60.000	132851.250	7501	73793158G
3	9112515 SC5184F 9952 FR	Avda. Camilo Sexto	5	NULL	400.000	450000.000	7503	73793158G
4	9001259 ZE7812S 2288 YY	Calle El Caldero	12	2 C	90.000	262500.000	7502	20039032Y
5	9110254 WS448O 3354 QW	Avda. Don Calero Martinez	10	2 B	95.000	220500.000	7505	41679701K
6	7445852 LP8524O 1119 ZX	Calle Sésamo Bravo	11	2 A	83.000	150000.000	7504	71297004V
7	7445852 HG2591T 0202 VN	Avda. Paco Casado	3	1 C	350.000	330000.000	7507	59804933B
8	8956211 VB6518F 0025 VV	Calle El Portal	15	NULL	523.000	452500.000	7509	20039032Y
9	1888254 CF4511R 9955 NB	Calle El Jorobado Danés	10	2 B	122.000	192300.000	7510	23310054Z
10	2255255 CC8858Y 0110 SA	Avda. Zapataria	11	3 B	162.000	126850.000	7509	73793158G
11	2255877 ZZ9523J 7751 MN	Avda. Punta Cuntera	5	5 A	63.000	115300.000	7505	44892211Y
12	9522624 YP4525P 0005 KK	Avda. La Sevillana Marena	2	1 D	80.000	175300.000	7506	94328497T
13	1388754 CX8466H 0666 PO	Avda. Palencia	1	1 A	107.000	107640.000	7510	64105041Q
14	2688754 CY7466H 0666 IO	Avda. Ejemplo	1	1 A	200.000	190000.000	7506	64105041Q
NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

Funciones

Enunciado en formato textual

1. Estima el valor económico de un barrio.
2. Calcular la recaudación total de un año
3. Calcular número de ocupantes que residen en un barrio
4. Buscar el dni del ocupante más mayor
5. El área media de los municipios de una provincia

Código SQL asociado

1.

```
-- 1. Estima el valor economico de un barrio
delimiter //
CREATE FUNCTION valorBarrio(nombreBarrio VARCHAR(45)) RETURNS decimal(10,3)
begin
    DECLARE precioTotal decimal(10,3);
    SELECT SUM(precioTasacion) into precioTotal
    FROM viviendas v, barrios b
    WHERE v.idBarrios = b.idBarrios AND b.nombre = nombreBarrio;
    return precioTotal;
end//
```

2.

```
-- 2. Calcular la recaudacion total de un año
delimiter //
CREATE FUNCTION recaudacionAnual( año INTEGER ) RETURNS DECIMAL(15, 4)
begin
    DECLARE recTotal decimal(15,3);
    SELECT SUM(importe) INTO recTotal FROM Impuestos WHERE (SELECT year(fechaActualPago)) = año;
    return recTotal;
end //
```

3.

```
CREATE FUNCTION poblacionBarrio( nomBarrio VARCHAR(25) ) RETURNS INTEGER
begin
    DECLARE recuento INTEGER;
    SELECT count(*) INTO recuento FROM Barrios b, Ocupantes o, Viviendas v WHERE b.idBarrios = v.idBarrios
    AND v.idViviendas = o.idViviendas AND b.nombre = nomBarrio;
    return recuento;
end //
```

4.

```
-- 4. Buscar el dni del ocupante más mayor
delimiter //
CREATE FUNCTION dniOcupanteMasMayor() RETURNS char(9)
begin
    DECLARE dniOc char(9);
    SELECT dni INTO dniOc FROM Ocupantes WHERE fNac = (SELECT min(fNAc) FROM Ocupantes);
    return dniOc;
end //
```

5.

```
-- 5. El área media de los municipios de una provincia
delimiter //
CREATE FUNCTION areaMediaMunicipios(prov varchar(45)) RETURNS decimal(10,2)
begin
    DECLARE areaMedia decimal(10,2);
    SELECT avg(area) INTO areaMedia
    FROM Municipios m, Provincias p
    WHERE p.nombre = prov AND m.codigoProvincia = p.codigoProvincia;
    return areaMedia;
end //
select areaMediaMunicipios('Valladolid');
```

Resultados

1.

valorBarrio('Pajarillos')
126850.000

2.

	recaudacionAnual(2018)
▶	3568.5220

3.

poblacionBarrio('Pajarillos')
1

4.

dniOcupanteMasMayor()
64105041Q

5.

areaMediaMunicipios('Valladolid')
107.24

Seguridad de la Base de Datos

Amenazas y contramedidas a aplicar

Los datos son uno de los activos más valiosos para las empresas a día de hoy este es el motivo por lo que la seguridad de estos ha saltado a un primer plano, las empresas cada vez almacenan más datos y de carácter más sensible en los SGBD, la pérdida o corrupción de estos datos podría suponer grandes pérdidas tanto de dinero como de reputación para una empresa.

Las amenazas a las que se somete un SGBD son innumerables, estas amenazas se clasifican conforme las áreas de incidencia que afecten, estas áreas son: robo y fraude, pérdida de confidencialidad, pérdida de la privacidad, pérdida de la integridad y pérdida de disponibilidad.

Una amenaza implica cualquier suceso que pueda afectar negativamente a un sistema y como hemos dicho antes son muchísimas las que un SGBD se puede encontrar, pero no todas las amenazas conllevan el mismo riesgo para el sistema, es por esto que las empresas se centran en buscar solución para las más serias o que más puedan afectar a su organización.

A continuación, citaremos algunas de las amenazas más importantes que pueden afectar a nuestro sistema:

Una de las áreas más afectadas por amenazas de seguridad es el apartado de usuarios que acceden a las bases de datos, aquí pueden surgir todo tipo de amenazas, robo de credenciales para un uso posterior, abuso de privilegios por parte de una cuenta, permisos excesivos, etc. Esta área es extremadamente sensible ya que cualquiera de estas amenazas citadas afecta a casi todas las áreas de incidencia, la pérdida de confidencialidad, privacidad y disponibilidad, así como el robo y fraude son consecuencias inevitables de estas vulnerabilidades ya que afectan al acceso de los datos y dependiendo de los permisos que se hayan visto comprometidos, nos podemos enfrentar a modificaciones, borrado o corrupción de estos.

También son usuales problemas derivados de riesgos físicos a los que se puede enfrentar el hardware, incendios, desastres naturales, daños físicos a los equipos… Todos estos escenarios son plausibles y ponen en gran peligro constante la disponibilidad e integridad de los datos.

Como en todo sistema informático, existe la posibilidad de que los SGBD se vean comprometidos ya sea por puertas traseras, hackeos o escuchas. No solo esto, si no que los sistemas o redes sobre los que están desplegados estos SGBD pueden verse comprometidos, pero esto no nos ocupa en este punto ya que solo prestaremos atención al terreno sobre el que administra la seguridad el SGBD.

Ahora que hemos visto las Amenazas más importantes que acechan a un SGBD, vamos a ver que mecanismos de defensa implementan estos sistemas para protegerse de dichas amenazas.

Comenzamos por el control de autorización, este proceso tendrá como resultado la concesión de un privilegio de acceso a la “persona” autorizada, este proceso conlleva una autenticación mediante unas credenciales. Por lo general, la creación de las cuentas de usuario individuales es tarea del administrador de la base de datos pero

La contramedida anterior va de la mano con la siguiente a tratar, el control de acceso, este se basa en la concesión y revocación de privilegios. Este es un área muy importante ya que, si se utiliza debidamente, resulta en un buen mecanismo de seguridad, pero si no se restringen debidamente los permisos, este mecanismo puede convertirse en una vía libre para vulnerabilidades. Los privilegios deberían ser concedidos de manera que solo se hiciese efectiva la concesión de uno si el usuario que lo recibe no puede hacer su trabajo sin él. Hay distintas formas de implementar este control de acceso, el estándar SQL sigue un control de acceso discrecional o DAC en el que los permisos fluyen desde el propietario de un objeto a quien se crea oportuno mediante las palabras reservadas GRANT y REVOKE. También se puede seguir un control de acceso obligatorio o MAC basado en políticas de nivel de sistema y clases de seguridad que determinan si cierto usuario es apto o no para el acceso o modificación de determinados datos. Este modelo impone dos restricciones, Read-down que restringe a tener una clase de seguridad mayor que el objeto a leer para poderse acceder a este y Write-up que restringe la escritura de un objeto si el sujeto no tiene la misma clase de seguridad o menor.

Las vistas son otra de las contramedidas integradas por el SGBD para evitar los robos de información y la pérdida de privacidad y confidencialidad. Las vistas son tablas virtuales creadas como respuesta a una solicitud. Estas vistas son creadas con el fin de ocultar partes de

la información a según qué usuarios de forma transparente, esto es, que el usuario no perciba que se está produciendo dicha ocultación. Esto es posible porque a un usuario se le pueden otorgar permisos sobre una vista y no sobre las tablas base, de esta forma se puede restringir aun más el acceso a las tablas de datos. Estas vistas son creadas por los administradores y se basan en una consulta select que definirá los contenidos de dicha vista.

Para enfrentarse a las pérdidas de integridad y maximizar la disponibilidad de los datos los SGBD integran un sistema de copia de seguridad y recuperación, esta funcionalidad hace posible la recuperación del sistema tras un fallo. Este servicio implica la creación de copias de seguridad de forma periódica en intervalos que vayan en función de las necesidades de la organización que lo implemente. Además de la copia en sí, también se guarda un archivo de las operaciones que se han ido realizando desde la última copia, gracias a estos dos elementos se puede llegar fácilmente hasta una recuperación prácticamente integral (hasta el último estado coherente).

De la mano de las copias de seguridad se encuentra la redundancia, también centrada en maximizar la disponibilidad de los datos y preservar la integridad de estos. Estás técnicas son fundamentales ya que los errores del hardware son prácticamente inevitables y es importante asegurar el servicio incluso cuando estos fallan. Una de las soluciones más usadas de este tipo es la tecnología RAID, su forma de operar se basa en la redundancia de discos físicos de forma que la información se pueda almacenar replicada. Hay diversas formas de distribuir estos datos y sus respectivos metadatos, las variantes más avanzadas de esta tecnología implementan un esquema de paridad para asegurar la integridad de los datos además de la redundancia.

No tanto como para recuperar la integridad de los datos, si no para mantenerla el SGBD cuenta con controles y restricciones de integridad varios. El primero es el de datos requeridos o el exigir que una columna tenga datos válidos no nulos o sí (NULL – NOT NULL). El segundo control consiste en restricciones de dominio para ciertas columnas (CHECK ()). El tercero o Integridad de entidades, el cual dictamina que la columna de la clave principal de una tabla no puede contener valores varios o nulos. El cuarto y último exige la integridad referencial, es decir, que los valores de las claves foráneas deben coincidir con sus valores en las tablas donde son clave principal.

La persecución por preservar la confidencialidad de los datos nos lleva a la siguiente contramedida, el cifrado. El cifrado se encarga, mediante un algoritmo de encriptación, de que los datos no sean legibles por ningún programa o sujeto que no tenga la clave. MySQL en si mismo emplea el cifrado para las contraseñas de los usuarios que existen en la base de datos y el algoritmo de cifrado usado por defecto es un hash SHA-2. Se pueden utilizar tanto algoritmos de cifrado simétrico como asimétrico dependiendo del nivel de seguridad deseado y los detalles logísticos de la organización. Si se aplica cifrado sobre los valores de una tabla hemos de tener en cuenta que el rendimiento de la base de datos en cada consulta se verá afectado ya que el proceso de encriptado y desencriptado ha de ser tenido en cuenta y ralentizará las consultas.

Amenazas y contramedidas en entornos web

Si la seguridad a nivel de base de datos ya es complicada cuando bajamos a nivel de red se hace aún mas difícil. Las comunicaciones en internet se realizan sobre un protocolo TCP/IP, este protocolo no fue ideado para ser seguro y si no se toman las medidas necesarias los datos pueden ser fácilmente interceptados por un agente externo. Para lograr una comunicación segura en internet de manera que la transmisión y recepción de datos no se vea

comprometida, esta comunicación debe seguir 5 principios: confidencialidad, integridad, autenticidad, no simulación y no repudio. Aun siguiendo todos estos pasos, un sistema puede verse atacado, existen ataques de todo tipo, inyección de código SQL, cross-site scripting (XSS), usurpación de credenciales, etc…

Por ello los mecanismos de seguridad de los que hemos hablado en el punto anterior no son suficientes, para proteger al sistema contra las amenazas de la web y así hacerlo accesible desde esta, hay que implementar medidas adicionales.

Es muy común en las organizaciones que se instalen servidores proxy, esta medida mejora el rendimiento de la aplicación a la vez que la seguridad. El servidor almacena las consultas recientes esto hace que las consultas sean respondidas más rápidamente y se ahorren accesos a disco los cuales son muy importantes en sistemas que manejen un volumen muy grande de peticiones. En el apartado de seguridad el servidor proxy contribuye en cuanto a que filtra las peticiones que tienen como destino el SGBD, de esta forma se restringen peticiones indebidas y se pueden implantar restricciones de conexión para evitar ataques al sistema.

La siguiente contramedida tiene que ver en ambientes empresariales en los que existan redes internas, un requisito fundamental en las bases de datos es que se intente en la medida de lo posible mantener el SGBD desconectado de redes de esta índole ya que si dicha red se viese comprometida pondría en peligro la seguridad de toda la base de datos. Es por esto que se implementan cortafuegos para restringir accesos no autorizados a los datos desde o hacia redes privadas, estos pueden implementarse como software o como hardware. Por lo general se implementan mediante diferentes técnicas, la más usual es el filtrado de paquetes, pero también puede por ejemplo utilizarse un servidor proxy como los mencionados en el punto anterior, a modo de cortafuegos si se sitúa entre un SGBD y la red privada.

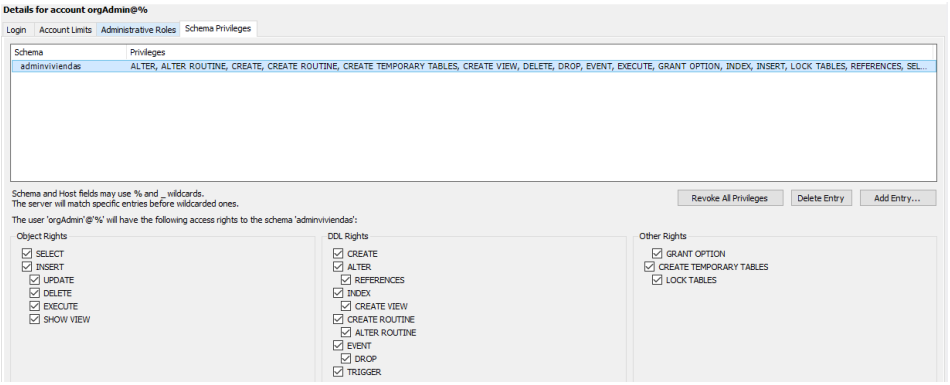
Estos puntos son sumamente importantes para la seguridad del SGBD, pero sin duda el punto más importante para poder hacer nuestra BD disponible en la red es establecer un canal seguro para la comunicación con esta. Aquí es donde entra en juego la capa de sockets seguros o SSL (Secure Sockets Layer), esta tecnología nos permite cifrar los datos que se transmiten en una conexión mediante algoritmos de cifrado. Cuando una conexión HTTP se realiza sobre SSL se dice que se está siguiendo el protocolo HTTPS o HTTP seguro. Estos protocolos se encargan de hacer posible que el cliente y servidor se autentiquen entre si y validen la información transmitida. Dependiendo de las necesidades logísticas y de confidencialidad de la organización que se esté tratando se utilizaran unas técnicas u otras de autenticación y cifrado, se pueden emplear tanto firmas digitales como certificados expedidos por una Autoridad de certificación en la que ambas partes de la comunicación confían.

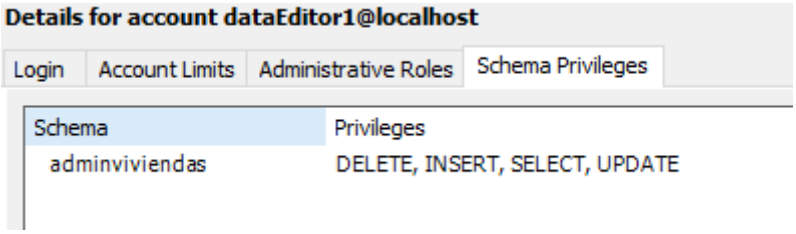
Por último, es importante volver a destacar en este punto la importancia del cifrado, en este caso el hasheo de las contraseñas de los usuarios. Los usuarios están almacenados en una tabla del servidor y van a ser un claro objetivo de hackers es por esto que debemos protegerlas. Pongamos un supuesto, nuestro sistema se ve comprometido por un ataque de inyección de código SQL, el atacante ha conseguido ejecutar una consulta SELECT en nuestro sistema y ha conseguido visualizar la tabla usuarios, si nuestras contraseñas no estuviesen cifradas podríamos enfrentarnos a un problema mucho mayor.

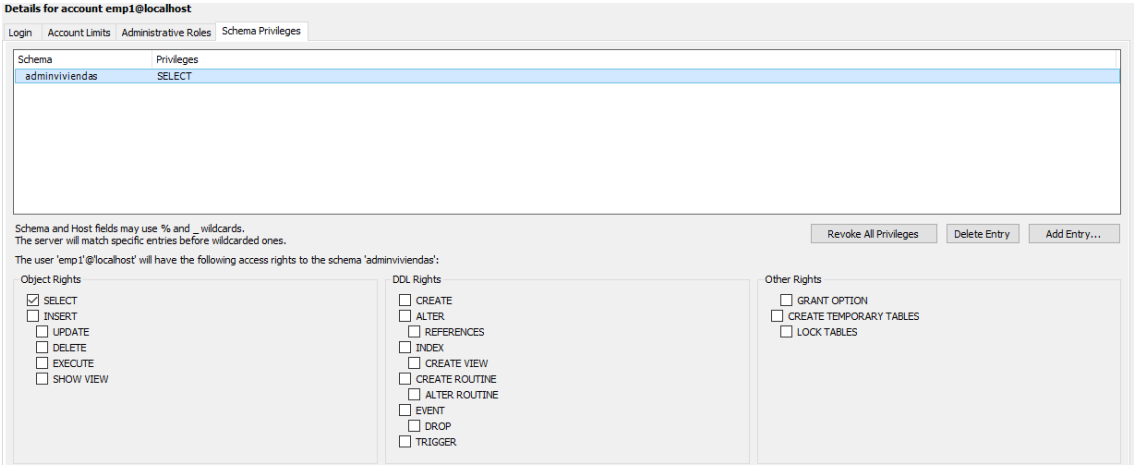
Administración de la Base de Datos

Usuarios creados y su asignación de permisos

```
33    -- Usuario destinado a ser usado por una aplicacion web
34
35 •  CREATE USER 'webAppUser'@'localhost'
36    IDENTIFIED BY '1111'
37    REQUIRE SSL;
38
39 •  GRANT INSERT, SELECT, UPDATE
40    ON adminviviendas.*
41    TO 'webAppUser'@'localhost';
42
```







```
25 •  CREATE USER 'dataEditor1'@'localhost'
26    IDENTIFIED BY '1111'
27    PASSWORD EXPIRE INTERVAL 60 DAY;
28
29 •  GRANT INSERT, UPDATE, DELETE, SELECT
30    ON adminviviendas.*
31    TO 'dataEditor1'@'localhost';
```

```
4 •  CREATE USER orgAdmin
5    IDENTIFIED BY '12341234'
6    PASSWORD EXPIRE INTERVAL 365 DAY;
7
8 •  GRANT ALL PRIVILEGES ON adminviviendas.* TO orgAdmin
9    WITH GRANT OPTION;
```

```
16 •  CREATE USER 'emp2'@'localhost'
17    IDENTIFIED BY '1234'
18    PASSWORD EXPIRE INTERVAL 60 DAY;
19
20 •  GRANT SELECT ON adminviviendas.*
21    TO 'emp1'@'localhost';                n tiene todos los permisos
2     -- sobre todas las tablas del esquema Admin Viviendas
3
4 •  CREATE USER orgAdmin
5    IDENTIFIED BY '12341234'
6    PASSWORD EXPIRE INTERVAL 365 DAY;
7
8 •  GRANT ALL PRIVILEGES ON adminviviendas.* TO orgAdmin
9    WITH GRANT OPTION;
```

Details for account webAppUser@localhost

Login | Account Limits | Administrative Roles | Schema Privileges

Schema	Privileges
adminviviendas	INSERT, SELECT, UPDATE

Referencias

APUNTES ASIGNATURA BASE DE DATOS II:

1- TEMA 3 – PROCESAMIENTO DE CONSULTAS, OPTIMIZACIÓN
2- TEMA 4 – PROGRAMACIÓN DE LA BASE DE DATOS
3- TEMA 5 – LA ADMINISTRACIÓN Y LA SEGURIDAD EN LAS BASES DE DATOS