



29-6-2023

# MySQL a SQL

Seminario de Actualización

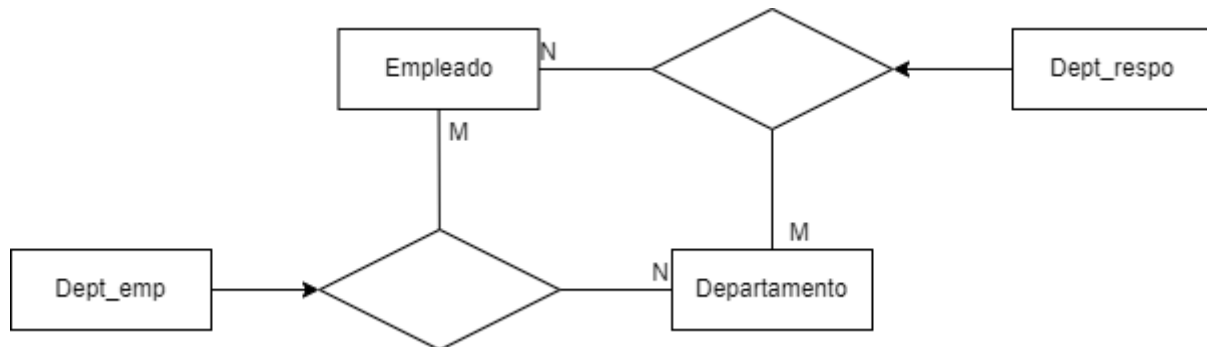


Alegre Julieta y Kempfen Dante  
INSTITUTO



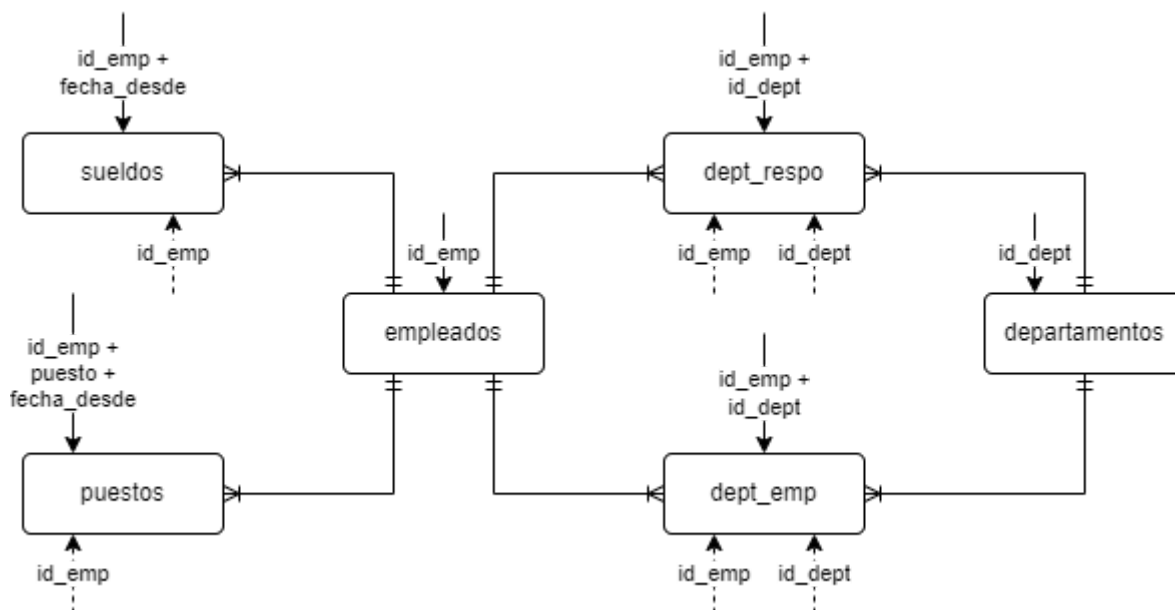
Una empresa cuenta con un sistema para el manejo del personal, que almacena información sobre los empleados, los departamentos en los que trabajan (incluyendo el historial). Cada departamento tiene asignados responsables en diferentes períodos. De los empleados se almacena la fecha de nacimiento, nombre, apellido, género, fecha de incorporación a la empresa, sueldo (incluyendo el historial) y los distintos puestos por los que fue pasando a lo largo del tiempo.

Cuando se realizó la implementación quedó la siguiente documentación:



Departamento = @id\_dept + nombre\_dept  
Dept\_emp = @Empleado\_ref\_M + @Departamento\_ref\_N + fecha desde + fecha hasta  
Dept\_respo = @Empleado\_ref\_N + @Departamento\_ref\_M + fecha desde + fecha hasta  
Empleado = @emp\_id + fecha nacimiento + nombre + apellido + genero + fecha alta + {sueldo + fecha desde + fecha hasta} + {puesto + fecha desde + fecha hasta}

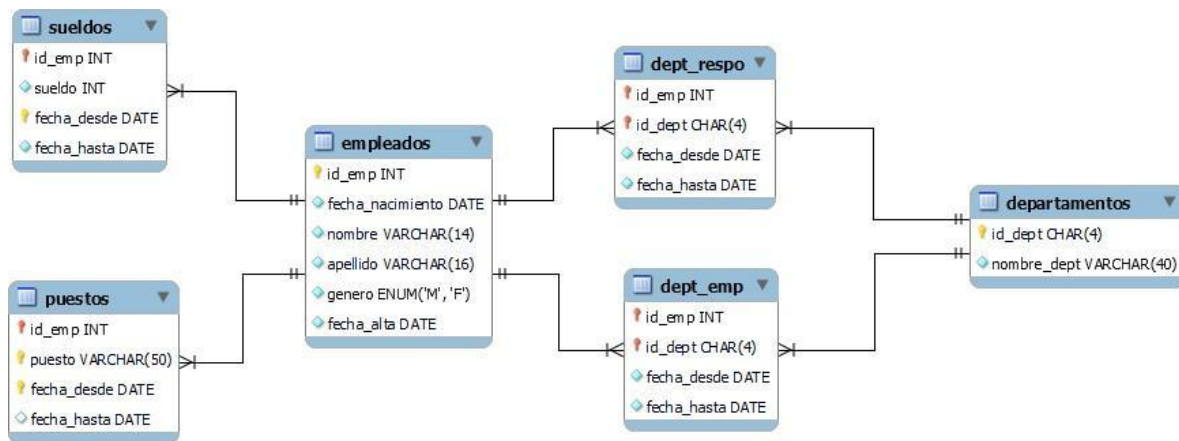
Basado en el DER se confeccionó el siguiente MER, normalizando los atributos repetitivos:



departamento = @id\_dept + nombre\_dept  
dept\_emp = @id\_emp + @id\_dept + fecha\_desde + fecha\_hasta  
dept\_respo = @id\_emp + @id\_dept + fecha\_desde + fecha\_hasta  
empleado = @emp\_id + fecha\_nacimiento + nombre + apellido + genero + fecha\_alta  
puestos = @id\_emp + @puesto + @fecha\_desde + fecha\_hasta  
sueldos = @id\_emp + sueldo + @fecha\_desde + fecha\_hasta



La implementación se realizó en MySQL, y luego de implementar se obtuvo el siguiente diagrama:



Se conserva el script que se utilizó para crear la base:

```
CREATE DATABASE Empleados;
```

```
USE Empleados;
```

```
CREATE TABLE empleados (
    id_emp          INT          NOT NULL,
    fecha_nacimiento DATE        NOT
    NULL, nombre     VARCHAR(14) NOT
    NULL,
    apellido        VARCHAR(16)  NOT NULL,
    genero          ENUM ('M', 'F') NOT NULL,
    fecha_alta      DATE          NOT NULL,
    CONSTRAINT PK_empleados PRIMARY KEY (id_emp)
);
```

```
CREATE TABLE departamentos (
    id_dept        CHAR(4)        NOT NULL,
    nombre_dept    VARCHAR(40)    NOT NULL UNIQUE,
    CONSTRAINT PK_departamentos PRIMARY KEY (id_dept)
);
```

```
CREATE TABLE dept_respo (
    id_emp          INT          NOT NULL,
    id_dept          CHAR(4)      NOT NULL,
    fecha_desde      DATE          NOT NULL,
    fecha_hasta      DATE          NOT NULL,
    CONSTRAINT FK_dept_respo_empleados FOREIGN KEY (id_emp) REFERENCES empleados (id_emp),
    CONSTRAINT FK_dept_respo_departamentos FOREIGN KEY (id_dept) REFERENCES departamentos
(id_dept),
    CONSTRAINT PK_dept_respo PRIMARY KEY (id_emp,id_dept)
);
```



```
CREATE TABLE dept_emp (  
    id_emp          INT          NOT NULL,  
    id_dept         CHAR(4)      NOT NULL,  
    fecha_desde     DATE         NOT NULL,  
    fecha_hasta     DATE         NOT NULL,  
    CONSTRAINT FK_dept_emp_empleados FOREIGN KEY (id_emp) REFERENCES empleados (id_emp),  
    CONSTRAINT FK_dept_emp_departamentos FOREIGN KEY (id_dept) REFERENCES departamentos  
(id_dept),  
    CONSTRAINT PK_dept_emp PRIMARY KEY (id_emp, id_dept)  
);
```

```
CREATE TABLE puestos (  
    id_emp          INT          NOT NULL,  
    puesto          VARCHAR(50)  NOT NULL,  
    fecha_desde     DATE         NOT NULL,  
    fecha_hasta     DATE,  
    CONSTRAINT FK_puestos_empleados FOREIGN KEY (id_emp) REFERENCES empleados (id_emp),  
    CONSTRAINT PK_puestos PRIMARY KEY (id_emp, puesto, fecha_desde)  
);
```

```
CREATE TABLE sueldos (  
    id_emp          INT          NOT NULL,  
    sueldo          INT          NOT NULL,  
    fecha_desde     DATE         NOT NULL,  
    fecha_hasta     DATE         NOT NULL,  
    CONSTRAINT FK_sueldos_empleados FOREIGN KEY (id_emp) REFERENCES empleados (id_emp),  
    CONSTRAINT PK_sueldos PRIMARY KEY (id_emp, fecha_desde)  
);
```

Y se conservan los scripts que se utilizaron para realizar la carga en la base MySQL.

Están disponibles en <https://1drv.ms/u/s!AlKfkXkZ-YCiouMeZrRNKn68b8zNsA?e=P8QXNc>



La empresa desea migrar la base a **SQL Server**. Para esto deberán:

- 1) Crear la base de datos y sus correspondientes tablas. En la medida de lo posible, respetar la misma estructura de MySQL, en los casos en que sea necesario cambiar algo, buscar que el impacto sea mínimo, y que permita utilizar los datos para la carga.
- 2) Realizar la carga de los archivos de datos, adaptándolos en lo que sea necesario para poder utilizarlos sin problemas en SQL Server. Los archivos están numerados del 01 al 08, y deben ser cargados en ese orden para mantener la integridad referencial. Detallar los cambios realizados sobre los archivos.
- 3) Junto con los datos, hay un archivo de control, el 09, que realiza consultas sobre la base de datos para controlar que los datos hayan sido cargados correctamente. Controla cantidad de registros y los datos mediante CRC. Deberán modificar este script para que funcione correctamente en SQL Server. El script utiliza la función MD5 que devuelve el texto correspondiente al hash calculado con MD5. Deberán programar una función que devuelva el mismo resultado (se puede obtener combinando funciones existentes en SQL Server).

Luego de la ejecución del script, deberán obtener una salida similar a la siguiente:

	tabla	registros_esperados	crc_esperado
1	departamentos	9	26eb605e3ec58718f8d588f005b3d2aa
2	dept_emp	331603	ccf6fe516f990bdaa49713fc478701b7
3	dept_respo	24	8720e2f0853ac9096b689c14664f847e
4	empleados	300024	4ec56ab5ba37218d187cf6ab09ce1aa1
5	puestos	443308	bfa016c472df68e70a03facafa1bc0a8
6	sueños	2844047	fd220654e95aea1b169624ffe3fca934

	tabla	registros_encontrados	crc_encontrado
1	departamentos	9	26eb605e3ec58718f8d588f005b3d2aa
2	dept_emp	331603	ccf6fe516f990bdaa49713fc478701b7
3	dept_respo	24	8720e2f0853ac9096b689c14664f847e
4	empleados	300024	4ec56ab5ba37218d187cf6ab09ce1aa1
5	puestos	443308	bfa016c472df68e70a03facafa1bc0a8
6	sueños	2844047	fd220654e95aea1b169624ffe3fca934

	tabla	coinciden_registros	coinciden_crc
1	departamentos	OK	OK
2	dept_emp	OK	OK
3	dept_respo	OK	OK
4	empleados	OK	OK
5	puestos	OK	OK
6	sueños	OK	OK

	Resumen	Resultado
1	UUID	C4443C83-FDE7-423C-B8CD-56460261170D
2	CRC	OK
3	Cantidad	OK
4	Tiempo	66300

Donde el valor UUID corresponde a un identificador único de la base de datos (RFC4122). En MySQL este valor está en @@server\_uuid, tendrán que buscar el equivalente para obtener esta información en SQL Server. Incluir en el trabajo el código de la función MD5 y la salida obtenida.



- 4) La información de los puestos está cargada en inglés y se desea hacer la traducción al español. Al revisar la información de la tabla, encontramos que la misma información está cargada muchas veces:

puesto	Cant
Assistant Engineer	15128
Engineer	115003
Manager	24
Senior Engineer	97750
Senior Staff	92853
Staff	107391
Technique Leader	15159

(7 rows affected)

Así que se desea modificar la estructura de la base de datos, buscando normalizar y minimizar esta redundancia de información, para lo cual se pide:

- Crear una nueva tabla ***puesto\_descr***, que tenga un identificador de tipo *smalint*, la descripción del puesto tal cual está en la tabla puestos y un tercer atributo con la descripción del puesto en español
- Reemplazar en la tabla puestos la descripción por el identificador

correspondiente. Detallar todos los pasos realizados para lograr esto.



## 1- Pasar Tablas a SQL

```
CREATE DATABASE Empleados;
go
USE Empleados;

CREATE TABLE empleados (
    id_emp INT NOT NULL,
    fecha_nacimiento DATE NOT NULL,
    nombre VARCHAR(14) NOT NULL,
    apellido VARCHAR(16) NOT NULL,
    genero CHAR(1) NOT NULL,
    fecha_alta DATE NOT NULL,
    CONSTRAINT PK_empleados PRIMARY KEY (id_emp)
);

CREATE TABLE departamentos (
    id_dept CHAR(4) NOT NULL,
    nombre_dept VARCHAR(40) NOT NULL UNIQUE,
    CONSTRAINT PK_departamentos PRIMARY KEY (id_dept)
);

CREATE TABLE dept_respo (
    id_emp INT NOT NULL,
    id_dept CHAR(4) NOT NULL,
    fecha_desde DATE NOT NULL,
    fecha_hasta DATE NOT NULL,
    CONSTRAINT FK_dept_respo_empleados FOREIGN KEY (id_emp) REFERENCES empleados
(id_emp),
    CONSTRAINT FK_dept_respo_departamentos FOREIGN KEY (id_dept) REFERENCES departamentos
(id_dept),
    CONSTRAINT PK_dept_respo PRIMARY KEY (id_emp,id_dept)
);

CREATE TABLE dept_emp (
    id_emp INT NOT NULL,
    id_dept CHAR(4) NOT NULL,
    fecha_desde DATE NOT NULL,
    fecha_hasta DATE NOT NULL,
    CONSTRAINT FK_dept_emp_empleados FOREIGN KEY (id_emp) REFERENCES empleados (id_emp),
    CONSTRAINT FK_dept_emp_departamentos FOREIGN KEY (id_dept) REFERENCES
departamentos (id_dept),
    CONSTRAINT PK_dept_emp PRIMARY KEY (id_emp,id_dept)
);

CREATE TABLE puestos (
    id_emp INT NOT NULL,
    puesto VARCHAR(50) NOT NULL,
    fecha_desde DATE NOT NULL,
    fecha_hasta DATE,
    CONSTRAINT FK_puestos_empleados FOREIGN KEY (id_emp) REFERENCES empleados (id_emp),
    CONSTRAINT PK_puestos PRIMARY KEY (id_emp,puesto, fecha_desde)
);

CREATE TABLE sueldos (
    id_emp INT NOT NULL,
    sueldo INT NOT NULL,
    fecha_desde DATE NOT NULL,
    fecha_hasta DATE NOT NULL,
    CONSTRAINT FK_sueldos_empleados FOREIGN KEY (id_emp) REFERENCES empleados (id_emp),
    CONSTRAINT PK_sueldos PRIMARY KEY (id_emp, fecha_desde)
);
```



## 2- Pasar tablas MYSQL a SQL

### 2.a- Tabla “departamentos”:

```
USE Empleados;
INSERT INTO departamentos VALUES
('d001','Marketing'),
('d002','Finanzas'),
('d003','Recursos Humanos'),
('d004','Produccion'),
('d005','Desarrollo'),
('d006','Gestion de Calidad'),
('d007','Ventas'),
('d008','Investigacion'),
('d009','Atencion a Clientes');
```

### 2.b- Tabla “empleados”:

```
BULK INSERT empleados
FROM 'C:\Users\User\OneDrive\Documentos\sql\02.empleados.csv'
WITH (
    FIELDTERMINATOR = ',', -- Reemplaza ',' si el archivo utiliza otro delimitador
    ROWTERMINATOR = '0x0a'
);
Go
```



### 2.c- Tabla “dept\_emp”:

```
BULK INSERT dept_emp
FROM 'C:\Users\User\OneDrive\Documentos\sql\03.dept_emp.csv'
WITH (
    FIELDTERMINATOR = ',', -- Reemplaza ',' si el archivo utiliza otro delimitador
    ROWTERMINATOR = '0x0a'
);
Go
```





2.d- Tabla “dept\_respo”:

```
USE Empleados;
INSERT INTO dept_respo VALUES
(110022, 'd001', '1985-01-01', '1991-10-01'),
(110039, 'd001', '1991-10-01', '9999-01-01'),
(110085, 'd002', '1985-01-01', '1989-12-17'),
(110114, 'd002', '1989-12-17', '9999-01-01'),
(110183, 'd003', '1985-01-01', '1992-03-21'),
(110228, 'd003', '1992-03-21', '9999-01-01'),
(110303, 'd004', '1985-01-01', '1988-09-09'),
(110344, 'd004', '1988-09-09', '1992-08-02'),
(110386, 'd004', '1992-08-02', '1996-08-30'),
(110420, 'd004', '1996-08-30', '9999-01-01'),
(110511, 'd005', '1985-01-01', '1992-04-25'),
(110567, 'd005', '1992-04-25', '9999-01-01'),
(110725, 'd006', '1985-01-01', '1989-05-06'),
(110765, 'd006', '1989-05-06', '1991-09-12'),
(110800, 'd006', '1991-09-12', '1994-06-28'),
(110854, 'd006', '1994-06-28', '9999-01-01'),
(111035, 'd007', '1985-01-01', '1991-03-07'),
(111133, 'd007', '1991-03-07', '9999-01-01'),
(111400, 'd008', '1985-01-01', '1991-04-08'),
(111534, 'd008', '1991-04-08', '9999-01-01'),
(111692, 'd009', '1985-01-01', '1988-10-17'),
(111784, 'd009', '1988-10-17', '1992-09-08'),
(111877, 'd009', '1992-09-08', '1996-01-03'),
(111939, 'd009', '1996-01-03', '9999-01-01');
```

2.e- Tabla “puestos”:

```
BULK INSERT puestos
FROM 'C:\Users\User\OneDrive\Documentos\sql\5_puesto.csv'
WITH (

    FIELDTERMINATOR = ',', -- Reemplaza ',' si el archivo utiliza otro delimitador
    ROWTERMINATOR = '\n' -- Reemplaza '\n' si el archivo utiliza otro carácter de
terminación de línea
);
```

2.e- Tabla “sueldos”:

```
BULK INSERT sueldos
FROM 'C:\Users\User\OneDrive\Documentos\sql\06.sueldos.csv'
WITH (

    FIELDTERMINATOR = ',', -- Reemplaza ',' si el archivo utiliza otro delimitador
    ROWTERMINATOR = '0x0a' -- Reemplaza '\n' si el archivo utiliza otro carácter de
terminación de línea
);
Go
```



2.f- Tabla “sueldos”:

```
BULK INSERT sueldos
FROM 'C:\Users\User\OneDrive\Documentos\sql\06.sueldos.csv'
WITH (
    FIELDTERMINATOR = ',', -- Reemplaza ',' si el archivo utiliza otro delimitador
    ROWTERMINATOR = '0x0a' -- Reemplaza '\n' si el archivo utiliza otro carácter de
    terminación de línea
);
Go
```

2.g- Tabla “sueldos”:

```
BULK INSERT sueldos
FROM 'C:\Users\User\OneDrive\Documentos\sql\07_sueldos.csv'
WITH (
    FIELDTERMINATOR = ',', -- Reemplaza ',' si el archivo utiliza otro delimitador
    ROWTERMINATOR = '\n' -- Reemplaza '\n' si el archivo utiliza otro carácter de
    terminación de línea
);
Go
```

2.h- Tabla “sueldos”:

```
BULK INSERT sueldos
FROM 'C:\Users\User\OneDrive\Documentos\sql\08.sueldos.csv'
WITH (
    FIELDTERMINATOR = ',', -- Reemplaza ',' si el archivo utiliza otro delimitador
    ROWTERMINATOR = '0x0a'
);
go
```



- 3- Realizar consultas sobre la base de datos para controlar que los datos hayan sido cargados correctamente.

```
use Empleados
go
DECLARE @tiempoini DATETIME2(6);
SET @tiempoini = SYSDATETIME();
SELECT @tiempoini;

DROP TABLE IF EXISTS valores_esperados, valores_encontrados;
CREATE TABLE valores_esperados (
    tabla VARCHAR(30) NOT NULL PRIMARY KEY,
    regs INT NOT NULL,
    crc_md5 VARCHAR(100) NOT NULL
);

SELECT * INTO valores_encontrados FROM valores_esperados WHERE 1 = 0;

INSERT INTO valores_esperados VALUES
('empleados', 300024, '4ec56ab5ba37218d187cf6ab09ce1aa1'),
('departamentos', 9, '26eb605e3ec58718f8d588f005b3d2aa'),
('dept_respo', 24, '8720e2f0853ac9096b689c14664f847e'),
('dept_emp', 331603, 'ccf6fe516f990bdaa49713fc478701b7'),
('puestos', 443308, 'bfa016c472df68e70a03facafa1bc0a8'),
('sueldos', 2844047, 'fd220654e95aea1b169624ffe3fca934');

SELECT tabla, regs AS registros_esperados, crc_md5 AS crc_esperado FROM
valores_esperados;
DECLARE @crc VARCHAR(100)
SET @crc = '';

DROP TABLE IF EXISTS tchecksum;
CREATE TABLE tchecksum (chk char(100));

SELECT @crc = LOWER(CONVERT(VARCHAR(100), HASHBYTES('MD5', CONCAT_WS('#', @crc,
id_dept, nombre_dept)), 2))
FROM departamentos
ORDER BY id_dept;

INSERT INTO tchecksum values (@crc)
INSERT INTO valores_encontrados VALUES ('departamentos', (SELECT COUNT(*) FROM
departamentos), @crc);

SET @crc = '';

SELECT @crc = LOWER(CONVERT(VARCHAR(100), HASHBYTES('MD5', CONCAT_WS('#', @crc,
id_dept, id_emp, fecha_desde, fecha_hasta)), 2))
FROM dept_emp ORDER BY id_dept, id_emp;
INSERT INTO tchecksum values (@crc)
INSERT INTO valores_encontrados VALUES ('dept_emp', (SELECT COUNT(*) FROM
dept_emp), @crc);
```



```
SET @crc = '';

SELECT @crc = LOWER(CONVERT(VARCHAR(100), HASHBYTES('MD5', CONCAT_WS('#', @crc,
id_emp, fecha_nacimiento, nombre, apellido, genero, fecha_alta)), 2))
FROM empleados;

INSERT INTO tchecksum values (@crc)
INSERT INTO valores_encontrados VALUES ('empleados', (SELECT COUNT(*) FROM
empleados),@crc);

SET @crc = '';

SELECT @crc = LOWER(CONVERT(VARCHAR(100), HASHBYTES('MD5', CONCAT_WS('#', @crc,
id_dept,id_emp, fecha_desde,fecha_hasta)) , 2))
FROM dept_respo ORDER BY id_dept,id_emp,fecha_desde;

INSERT INTO tchecksum values (@crc)
INSERT INTO valores_encontrados VALUES ('dept_respo', (SELECT COUNT(*) FROM
dept_respo), @crc);

SET @crc = '';

SELECT @crc = LOWER(CONVERT(VARCHAR(100), HASHBYTES('MD5', CONCAT_WS('#',
@crc,id_emp, puesto, fecha_desde,fecha_hasta)), 2))
FROM puestos ORDER BY id_emp,puesto,fecha_desde;
INSERT INTO tchecksum values (@crc)
INSERT INTO valores_encontrados VALUES ('puestos', (SELECT COUNT(*) FROM
puestos), @crc);

SET @crc = '';

SELECT @crc = LOWER(CONVERT(VARCHAR(100), HASHBYTES('MD5', CONCAT_WS('#',
@crc,id_emp, sueldo, fecha_desde,fecha_hasta)) , 2))
FROM sueldos ORDER BY id_emp,fecha_desde,fecha_hasta;
INSERT INTO tchecksum values (@crc)
INSERT INTO valores_encontrados VALUES ('sueldos', (SELECT COUNT(*) FROM
sueldos), @crc);

DROP TABLE IF EXISTS tchecksum;

SELECT tabla, regs AS 'registros_encontrados', crc_md5 AS crc_encontrado FROM
valores_encontrados;

SELECT e.tabla,
CASE WHEN e.regs = f.regs THEN 'OK' ELSE 'No OK' END AS coinciden_registros,
CASE WHEN e.crc_md5 = f.crc_md5 THEN 'OK' ELSE 'No OK' END AS coinciden_crc
FROM valores_esperados e
INNER JOIN valores_encontrados f ON e.tabla = f.tabla;

declare @crc_fail int
declare @count_fail int
SET @crc_fail=(SELECT COUNT(*) FROM valores_esperados e INNER JOIN
valores_encontrados f ON (e.tabla=f.tabla) WHERE f.crc_md5 != e.crc_md5);
SET @count_fail=(SELECT COUNT(*) FROM valores_esperados e INNER JOIN
valores_encontrados f ON (e.tabla=f.tabla) WHERE f.regs != e.regs);

DROP TABLE valores_esperados,valores_encontrados;
```



```
SELECT 'UUID' AS Resumen, convert(varchar(36),NEWID ( )) AS Resultado
UNION ALL
SELECT 'CRC', CASE WHEN @crc_fail = 0 THEN 'OK' ELSE 'Error' END
UNION ALL
SELECT 'Cantidad', CASE WHEN @crc_fail = 0 THEN 'OK' ELSE 'Error' END
UNION ALL
SELECT 'Tiempo', convert(varchar(10),DATEDIFF(MICROSECOND, @tiempoini,
SYSDATETIME()) / 1000)
```

### Resultado:

Results		Messages	
		(No column name)	
1	2023-06-29 01:22:14.202192		
	tabla	registros_esperados	crc_esperado
1	departamentos	9	26eb605e3ec58718f8d588f005b3d2aa
2	dept_emp	331603	ccf6fe516f990bdaa49713fc478701b7
3	dept_respo	24	8720e2f0853ac9096b689c14664f847e
4	empleados	300024	4ec56ab5ba37218d187cf6ab09ce1aa1
5	puestos	443308	bfa016c472df68e70a03facafa1bc0a8
6	sueños	2844047	fd220654e95aea1b169624ffe3fca934
	tabla	registros_encontrados	crc_encontrado
1	departamentos	9	26eb605e3ec58718f8d588f005b3d2aa
2	dept_emp	331603	ccf6fe516f990bdaa49713fc478701b7
3	empleados	300024	4ec56ab5ba37218d187cf6ab09ce1aa1
4	dept_respo	24	8720e2f0853ac9096b689c14664f847e
5	puestos	443308	bfa016c472df68e70a03facafa1bc0a8
6	sueños	2844047	fd220654e95aea1b169624ffe3fca934
	tabla	coinciden_registros	coinciden_crc
1	departamentos	OK	OK
2	dept_emp	OK	OK
3	empleados	OK	OK
4	dept_respo	OK	OK
5	puestos	OK	OK
6	sueños	OK	OK
	Resumen	Resultado	
1	UUID	4040D4A2-A4A4-4099-8446-D78FA20A5C09	
2	CRC	OK	
3	Cantidad	OK	



4-

4-1: Agrupar puestos

```
SELECT puesto,count(puesto) AS Cantidad FROM puestos
```

```
GROUP BY puesto
```

puesto	Cantidad
Assistant Engineer	15128
Engineer	115003
Senior Engineer	97750
Manager	24
Staff	107391
Technique Leader	15159
Senior Staff	92853

4.1

```
CREATE TABLE puesto_descr (
```

```
    id_puesto SMALLINT PRIMARY KEY,
```

```
    descripcion_ingles VARCHAR(50),
```

```
    descripcion_espanol VARCHAR(50)
```

```
);
```

```
GO
```

```
insert into puesto_descr values (1,'Assistant Engineer', 'Ingeniero Asistente'),
```

```
(2,'Engineer', 'Ingeniero'),(3,'Manager', 'Gerente'), (4,'Staff', 'Personal'), (5,'Technique
```

```
Leader','Líder Técnico'),(6,'Senior Staff', 'Personal Superior'),(7,'Senior Engineer', 'Ingeniero
```

```
Superior');
```



```
1 SELECT * FROM puesto_descr
```

#	id_puesto	descripcion_ingles	descripcion_espanol
1		Assistant Engineer	Ingeniero Asistente
2		Engineer	Ingeniero
3		Manager	Gerente
4		Staff	Personal
5		Technique Leader	Líder Técnico
6		Senior Staff	Personal Superior
7		Senior Engineer	Ingeniero Superior

4.3 : Actualizar la estructura de la tabla "puestos":

```
ALTER TABLE puestos ADD id_puesto smallint;
```

4.4

```
SET id_puesto = pd.id
```

```
FROM puestos p
```

```
JOIN puesto_descr pd ON p.descripcion = pd.descripcion_español;
```

Tambien se puede

```
UPDATE puesto_descr pd
```

```
SET descripcion = descripcion_español
```

```
FROM puesto p
```

```
WHERE p.id.emp = pd.id_puesto
```



Enlace: Tablas

[https://drive.google.com/drive/folders/1ZCdZJWVFMUufGXpKQS8IKDXRGLIQ63\\_x?usp=sharing](https://drive.google.com/drive/folders/1ZCdZJWVFMUufGXpKQS8IKDXRGLIQ63_x?usp=sharing)

```

CREATE OR ALTER FUNCTION [dbo].[MD5]
(
    -- Add the parameters for the function here
    @pcadena varchar(max)
)
RETURNS varchar(32)
AS
BEGIN
    -- Declare the return variable here
    DECLARE @Result varchar(32)

    -- Add the T-SQL statements to compute the return value here
    SELECT @Result = LOWER(CONVERT(VARCHAR(32), HASHBYTES('MD5', @pcadena), 2))

    -- Return the result of the function
    RETURN @Result

END

```

```

SET @crc= '';
| SELECT @crc = dbo.MD5(CONCAT_WS('#',@crc,id_emp,fecha_nacimiento,nombre,apellido,genero,fecha_alta))
  FROM empleados ORDER BY id_emp;
INSERT INTO valores_encontrados VALUES ('empleados', (SELECT COUNT(*) FROM empleados),@crc);

```