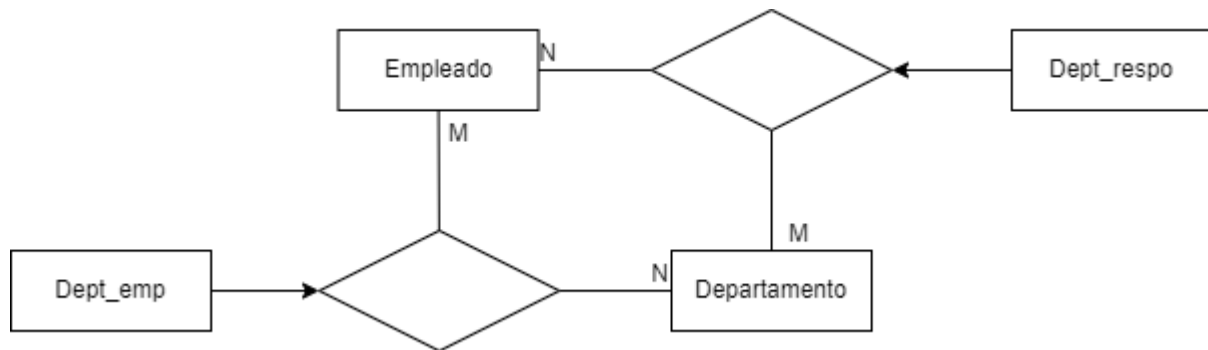




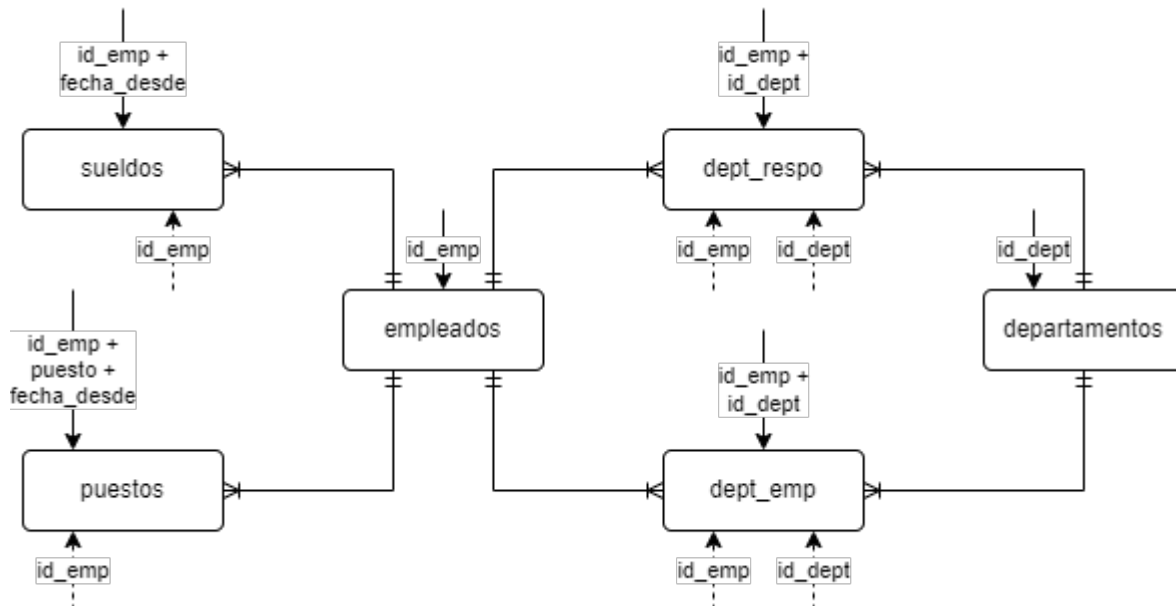
Una empresa cuenta con un sistema para el manejo del personal, que almacena información sobre los empleados, los departamentos en los que trabajan (incluyendo el historial). Cada departamento tiene asignados responsables en diferentes períodos. De los empleados se almacena la fecha de nacimiento, nombre, 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}

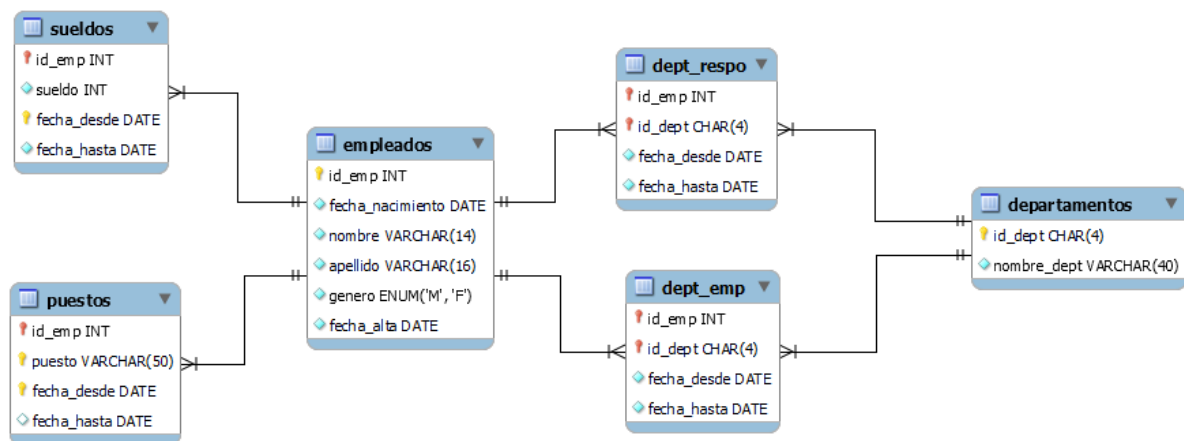
Basado en el DER se confeccionó el siguiente MER, normalizando los atributos repetitivos:



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!AIKfkXkZ-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.



Consultas

- 5) Realizar las consultas asignadas. En las consultas g, h e i incluir solamente las sentencias SQL, en el resto de las consultas, incluir las sentencias SQL y una captura del resultado obtenido. Como parámetros para realizar los filtros en las consultas, cada miembro del equipo tendrá asignado un `id_emp`, que se conformará utilizando su número de DNI tomando el primer dígito y los cuatro últimos. Por ejemplo, si mi número de DNI es **12.345.678**, el `id_emp` asignado es **15678**.

Del mismo modo, se asignará un `id_dept`, tomando el último dígito del DNI. Considerando el ejemplo anterior, el `id_dept` asignado sería **d008**. Como los `id_dept` van del 1 al 9, aquellos cuyo DNI termine en cero, deberán usar **d009**.

Las consultas deberán contener los parámetros asignados, pero deben ser realizadas de tal modo que si se modifica el `id_emp` o el `id_dept`, las mismas funcionen correctamente.

Asimismo, deberían funcionar en caso de que se realicen cambios en los datos existentes en la base.

- Devolver los datos (`id`, fecha de nacimiento, nombre y apellido concatenados, género y fecha de alta) del empleado asignado si alguna vez trabajó en el departamento asignado.
- Devolver los datos (`id`, fecha de nacimiento, nombre y apellido concatenados, género y fecha de alta) del empleado asignado si no trabaja actualmente en el departamento asignado.
- Devolver los datos (`id`, nombre y apellido concatenados) del responsable actual del departamento asignado.
- Devolver los datos del empleado asignado, incluyendo el nombre del departamento en el que trabaja actualmente (realizar la consulta considerando la posibilidad de que el empleado ya no trabaje en la compañía, y en ese caso se debería mostrar el último departamento al que estuvo asignado).
- Agregar a la consulta anterior, el apellido del responsable del departamento.
- Considerando el menor y el mayor sueldo, ¿qué porcentaje de aumento recibió el empleado asignado? El formato de la respuesta deberá contar con dos decimales y el símbolo de porcentaje, por ejemplo 20.50%
- Obtener la lista de empleados (`emp_id`, `fecha_nacimiento`, nombre, apellido y género) que trabajan actualmente en el departamento asignado, cuyo sueldo actual es mayor a 120.000. El puesto que ocupa debe ser Ingeniero (el nombre del puesto debe contener "Engineer") o tener categoría Senior (el nombre del puesto contiene "Senior").
- Agregar un nuevo departamento, con el código **d010** y el nombre "Inteligencia Artificial".
- Con vigencia desde el 01/01/2023, asigne los empleados obtenidos en la consulta g al nuevo departamento **d010** y otórgueles un 15% de aumento de sueldo. Realice todos los comandos DML necesarios para que estos cambios queden reflejados correctamente en la base de datos.
- Liste la cantidad de empleados de empleados y sueldo promedio de cada departamento, considerado solamente a los empleados que trabajan actualmente.