

EXAMEN FINAL DE LA 1ª EVALUACIÓN:

1. Dado el siguiente enunciado diseña el modelo E/R correspondiente. (3 Puntos):

Tenemos que diseñar una base de datos para gestionar las reservas de hoteles y vuelos que realizan los clientes de una agencia de viajes:

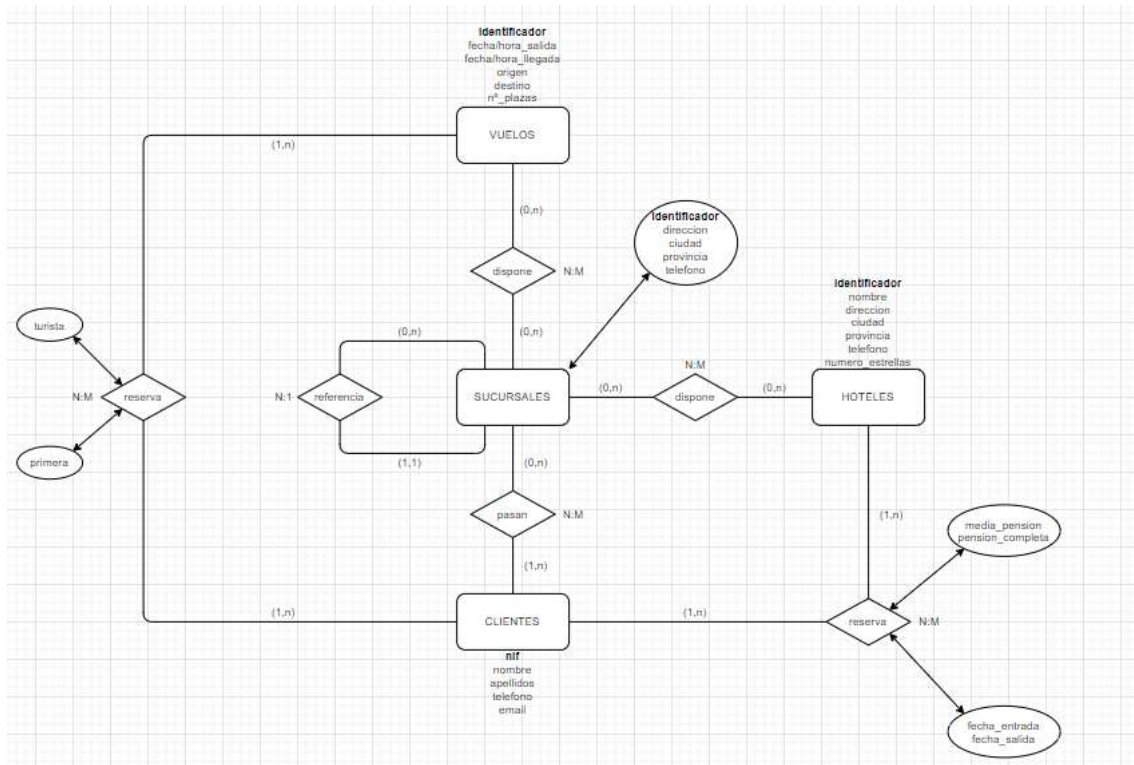
La agencia de viajes tiene oficinas **sucursales**. Cada sucursal está definida por el identificador de sucursal, dirección, ciudad, provincia y teléfono. Una sucursal tiene siempre otra oficina de referencia con la que se comunica normalmente.

Por su parte, los clientes normalmente se pasan por las sucursales para contratar vuelos y estancias en hoteles, sabiendo que:

- Cada **vuelo** está definido por un identificador, fecha/hora de salida, fecha/hora de llegada, origen, destino y número de plazas totales.
- Cada **hotel** está definido por el identificador del hotel, nombre, dirección, ciudad, provincia, teléfono y número de estrellas.
- La información que se desea almacenar para cada **cliente** es un nif, nombre, apellidos, teléfono y email.

También habrá que tener en cuenta la siguiente información:

- A la agencia de viajes le interesa conocer a través de qué sucursal ha contratado cada cliente los servicios de vuelo y alojamiento.
- A la hora de reservar un vuelo el cliente puede elegir cualquiera de los vuelos que ofrece la agencia y en qué clase (turista o primera) desea viajar.
- El cliente se puede hospedar en cualquiera de los hoteles que ofrece la agencia, y elegir el régimen de hospedaje (media pensión o pensión completa). Siendo significativa la fecha de entrada y de salida.



En el caso de “turista”, “primera”, “media_pension”, “pension_completa”; se tratan de campos de tipo BOOLEAN.

2. Dado el siguiente el modelo E/R anterior, pásalo al modelo relacional. (1,5 Puntos):

VUELOS (**identificador**, fecha/hora_salida, fecha/hora_llegada, origen, destino, nº_plazas).

HOTELES (**identificador**, nombre, dirección, ciudad, provincia, teléfono, numero_estrellas).

SUCURSALES (**identificador**, dirección, ciudad, provincia, teléfono, **identificador_SUCURSALES**).

CLIENTES (**nif**, nombre, apellidos, teléfono, email).

RESERVA_HOTEL (**nif_CLIENTES**, **identificador_HOTELES**, media_pension, pension_completa, fecha_entrada, fecha_salida).

RESERVA_VUELOS (**nif_CLIENTES**, **identificador_VUELOS**, turista, primera).

DISPOSICION_VUELOS (**identificador_SUCURSALES**, **identificador_VUELOS**).

DISPOSICION_HOTELES (**identificador_SUCURSALES**, **identificador_HOTELES**).

CONTRATO_SUCURSALES (**identificador_SUCURSALES**, **nif_CLIENTES**).

3. Ejercicio de Normalización. (1,5 Puntos):

Teniendo en cuenta que:

- Una editorial tiene una sede y una persona de contacto (aunque no se encuentran registradas actualmente, pero se registrará en un futuro)
- Se quiere también añadir en un futuro el año de edición del libro.

Realiza el modelo relacional de modo que el diseño se encuentre al menos en 3ª forma normal.

LIBROS (Cod_libro, ISBN, Titulo, Autor, año_edicion).

AUTORES (Autor, Pais_autor, ISBN).

EDITORIALES (Editorial, ISBN).

PRESTAMO (ISBN, Tele, Fecha_prestamo, Fecha_devolucion).

USUARIO (Apellido1, Apellido2, Nombre, Tele).

PERSONA_CONTACTO (nombre, apellidos, teléfono, nif).

4. Dada la base de datos empleado (script2.sql), realiza las siguientes consultas (4 puntos):

1. Lista el código de los departamentos de los empleados que aparecen en la tabla empleado, eliminando los códigos que aparecen repetidos:

```
SELECT DISTINCT(codigo)
FROM empleado;
```



empleado (13r x 1c)

codigo
12
13
1
6
11
2
7
9
3
8
4
5
10

2. Lista el nombre y apellidos de los empleados en una única columna:

```
SELECT CONCAT(nombre, apellido1, apellido2)
FROM empleado
```

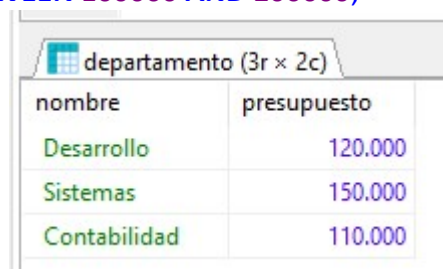


empleado (13r x 1c)

CONCAT(nombre, apellido1, apellido2)
AarónRiveroGómez
AdelaSalasDíaz
AdolfoRubioFlores
(NULL)
MarcosLoyolaMéndez
MaríaSantanaMoreno
(NULL)
PepeRuizSantana
JuanGómezLópez
DiegoFloresSalas
MartaHerreraGil
IreneSalasFlores
Juan AntonioSáezGuerrero

3. Devuelve una lista con el nombre de los departamentos y el presupuesto, de aquellos que tienen un presupuesto entre 100000 y 200000 euros:

```
SELECT nombre, presupuesto
FROM departamento
WHERE presupuesto BETWEEN 100000 AND 200000;
```



departamento (3r x 2c)

nombre	presupuesto
Desarrollo	120.000
Sistemas	150.000
Contabilidad	110.000

4. Devuelve un listado con los empleados y los datos de los departamentos donde trabaja cada uno. Ordena el resultado, en primer lugar, por el nombre del departamento (en orden alfabético) y en segundo lugar por los apellidos y el nombre de los empleados:

```
SELECT empleado.*, departamento.*
FROM empleado
INNER JOIN departamento ON empleado.codigo_departamento=departamento.codigo
ORDER BY departamento.nombre ASC, empleado.apellido1, empleado.apellido2,
empleado.nombre;
```

Resultado #1 (11r x 10c)									
codigo	nif	nombre	apellido1	apellido2	codigo_departamento	codigo	nombre	presupuesto	gastos
4	77705545E	Adrián	Suárez	(NULL)	4	4	Contabilidad	110.000	3.000
11	67389283A	Marta	Herrera	Gil	1	1	Desarrollo	120.000	6.000
1	32481596F	Aarón	Rivero	Gómez	1	1	Desarrollo	120.000	6.000
6	38382980M	María	Santana	Moreno	1	1	Desarrollo	120.000	6.000
10	46384486H	Diego	Flores	Salas	5	5	I+D	375.000	380.000
5	17087203C	Marcos	Loyola	Méndez	5	5	I+D	375.000	380.000
3	R6970642B	Adolfo	Rubio	Flores	3	3	Recursos Humanos	280.000	25.000
8	71651431Z	Pepe	Ruiz	Santana	3	3	Recursos Humanos	280.000	25.000
9	56399183D	Juan	Gómez	López	2	2	Sistemas	150.000	21.000
7	80576669X	Pilar	Ruiz	(NULL)	2	2	Sistemas	150.000	21.000
2	Y5575632D	Adela	Salas	Díaz	2	2	Sistemas	150.000	21.000

5. Devuelve un listado con los datos de los empleados que trabajan en el departamento de Sistemas, Contabilidad o I+D. Ordena el resultado alfabéticamente:

```
SELECT empleado.*, departamento.nombre
FROM empleado
INNER JOIN departamento ON empleado.codigo_departamento=departamento.codigo
WHERE departamento.nombre IN ('Sistemas', 'Contabilidad', 'I+D')
ORDER BY empleado.nombre;
```

Resultado #1 (6r x 7c)						
codigo	nif	nombre	apellido1	apellido2	codigo_departamento	nombre
2	Y5575632D	Adela	Salas	Díaz	2	Sistemas
4	77705545E	Adrián	Suárez	(NULL)	4	Contabilidad
10	46384486H	Diego	Flores	Salas	5	I+D
9	56399183D	Juan	Gómez	López	2	Sistemas
5	17087203C	Marcos	Loyola	Méndez	5	I+D
7	80576669X	Pilar	Ruiz	(NULL)	2	Sistemas

6. Calcula el número de empleados que trabajan en cada uno de los departamentos que tienen un presupuesto mayor a 200000 euros. Detrás del número de empleados tiene que aparecer la palabra “empleados”:

```
SELECT departamento.nombre, CONCAT(COUNT(empleado.nombre),' empleados')
FROM empleado
INNER JOIN departamento ON empleado.codigo_departamento=departamento.codigo
WHERE departamento.presupuesto>200000
GROUP BY codigo_departamento;
```

departamento (2r x 2c)	
nombre	CONCAT(COUNT(empleado.nombre),' empleados')
Recursos Humanos	2 empleados
I+D	2 empleados

7. Devuelve el nombre del departamento con menor presupuesto y la cantidad 3 que tiene asignada (sin usar limit):

```
SELECT nombre, MIN(presupuesto)
FROM departamento;
```

departamento (1r x 2c)	
nombre	MIN(presupuesto)
Desarrollo	0

8. Listado de los empleados que tienen el segundo apellido con más letras que el 2º apellido de Aaron:

```
SELECT nombre
FROM empleado
WHERE nombre>(
SELECT nombre
FROM empleado
WHERE nombre='Aaron');
```

empleado (12r x 1c)	
nombre	
Adela	
Adolfo	
Adrián	
Marcos	
María	
Pilar	
Pepe	
Juan	
Diego	
Marta	
Irene	
Juan Antonio	

9. Lista el nombre y apellidos de los empleados en una única columna, convirtiendo todos los caracteres en mayúscula. Prueba con la función upper():

```
SELECT UPPER(CONCAT(nombre, apellido1, apellido2))
FROM empleado;
```

empleado (13r x 1c)
UPPER(CONCAT(nombre, apellido1, apellido2))
AARÓN RIVERO GÓMEZ
ADELA SALAS DÍAZ
ADOLFO RUBIO FLORES
(NULL)
MARCOS LOYOLA MÉNDEZ
MARÍA SANTANA MORENO
(NULL)
PEPE RUIZ SANTANA
JUAN GÓMEZ LÓPEZ
DIEGO FLORES SALAS
MARTA HERRERA GIL
IRENE SALAS FLORES
JUAN ANTONIO SÁEZ GUERRERO

10. Devuelve una lista con el nombre de los empleados que tienen los departamentos que no tienen un presupuesto entre 100000 y 200000 euros:

```
SELECT empleado.nombre, departamento.presupuesto
FROM empleado
INNER JOIN departamento ON empleado.codigo_departamento=departamento.codigo
WHERE departamento.presupuesto NOT BETWEEN 100000 AND 200000;
```

Resultado #1 (4r x 2c)	
nombre	presupuesto
Adolfo	280.000
Pepe	280.000
Marcos	375.000
Diego	375.000