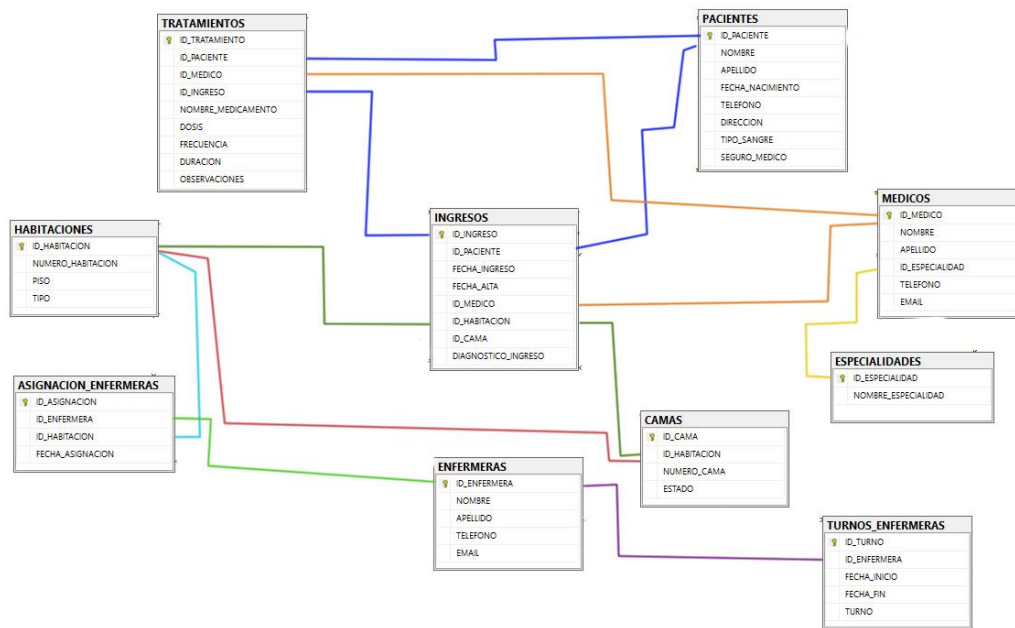


Esquema relacional de la base de datos ELHUBU



Importante: aunque en el esquema relacional de la imagen no aparece reflejado, tanto la tabla médicos como la de enfermeras tienen dos campos llamados apellido1 y apellido2 para guardar los apellidos de los empleados.

Ejercicio transacciones:

1.- En la tabla habitaciones, tenemos identificadores para 122 habitaciones. Pero en la tabla camas, solo aparecen reflejadas 100 de ellas. Crear una transacción con las sentencias de inserción necesarias para dar de alta el resto de camas (en todas ellas el estado será DISPONIBLE), teniendo en cuenta que algunas de esas habitaciones tienen una sola cama y otras dos (revisar las tablas de habitaciones y camas para comprender el sistema de asignación)

Ejercicios procedimientos:

1.- Con el fin de emplear la información que maneja el centro adecuadamente, el hospital ha decidido utilizar un dominio de correo propio y generar cuentas de correo para todos los médicos. Las instrucciones que nos dan, son que debemos generar una dirección de correo para cada médico, que esté compuesta por las dos primeras letras de su nombre, y las dos primeras de cada uno de sus apellidos. Seguido del sufijo @elhubu.es. Por ejemplo, Raquel Hernando Bernabé sería: rahebe@elhubu.es

Diseñar un procedimiento almacenado, que reciba como parámetro el id del médico, genere su cuenta de correo, e inserte esa información en la tabla de médicos, en la línea correspondiente.

2.- El hospital ha recibido quejas de los médicos por sus nuevas direcciones de correo electrónico. Con el fin de no tener el mismo problema con el personal de enfermería nos piden que, las direcciones de correo de éstos, estén compuestas por las tres primeras letras de su primer apellido, las dos primeras de su segundo apellido y las dos primeras de su nombre. Seguidas del sufijo @elhubu.es. Por ejemplo, Mario García Domínguez sería: gardoma@elhubu.es

Necesitamos diseñar un procedimiento almacenado que reciba como parámetro el id de enfermera y genere su cuenta de correo, insertando el dato en el campo correspondiente.

3.-El hospital quiere poder asignar la atención de habitaciones a enfermeras de una forma automatizada. De manera que, indicando el id de la enfermera, el número de la habitación, y la fecha, se pueda asignar una nueva enfermera a una habitación.

Diseñar un procedimiento almacenado que reciba esos tres datos y modifique la tabla de ASIGNACION_ENFEREMERAS convenientemente.

Nota: debéis tener en cuenta que en la tabla de asignación de enfermeras no figura el número de la habitación, sino su identificador. Por lo que, tendréis que averiguar cuál es el identificador correspondiente al número de habitación que os han proporcionado, antes de realizar la nueva asignación.

4.-El hospital necesita saber qué enfermeras están trabajando en fechas determinadas y qué turno realizan.

Por ello, necesitamos un procedimiento almacenado que, dada una fecha, nos diga el identificador de la enfermera, su nombre, la fecha de inicio y fin de su turno, los números de las habitaciones que atiende y si está de tarde, mañana o noche.

Nota: si lo veis conveniente, podéis diseñar una vista con los datos de las tablas que contienen esa información y después emplear la vista para consultarla en el procedimiento almacenado.

Función concat

Esta función devuelve una cadena resultante de la concatenación, o la combinación, de dos o más valores de cadena de una manera integral.

Sintaxis

```
CONCAT ( argument1 , argument2 [ , argumentN ] ... )
```

Ejemplo

```
SELECT CONCAT ('Happy ', 'Birthday ', 11, '/', '25') AS fecha;
```

Resultado

```
fecha
-----
Happy Birthday 11/25
```

Función substring

Devuelve parte de una expresión de caracteres, binaria, de texto o de imagen en SQL Server.

Sintaxis

```
SUBSTRING ( expression, start, length )
```

Ejemplo

```
SELECT SUBSTRING('abcdef', 2, 3) AS CADENA;
```

Resultado

```
CADENA
```

```
-----
```

```
bcd
```

Función datediff

Esta función devuelve el recuento (como un valor entero con firma) de los límites *startdate* que se han cruzado entre los valores *startdate* y *enddate* especificados.

Sintaxis

```
DATEDIFF ( datepart , startdate , enddate )
```

Datepart, especifica las unidades en las que DATEDIFF se notifica la diferencia entre *startdate* y *enddate*. Entre las unidades de *datepart* usadas comúnmente se incluyen month o day.

Ejemplo

```
SELECT DATEDIFF(day, '2005-12-30', '2006-01-01') as diferenciafecha;
```

Resultado

```
Diferenciafecha
```

```
-----
```

```
2
```