

Escuela Superior Politécnica de Chimborazo
Facultad de Informática y Electrónica
Carrera de Software

Base de Datos 1
Paralelo 1

Consultas Combinadas y Conjuntos

1. - Datos Generales.

Tema: Lenguaje de manejo de datos (DML) - Selección

Nombre: [REDACTED]

Código: [REDACTED]

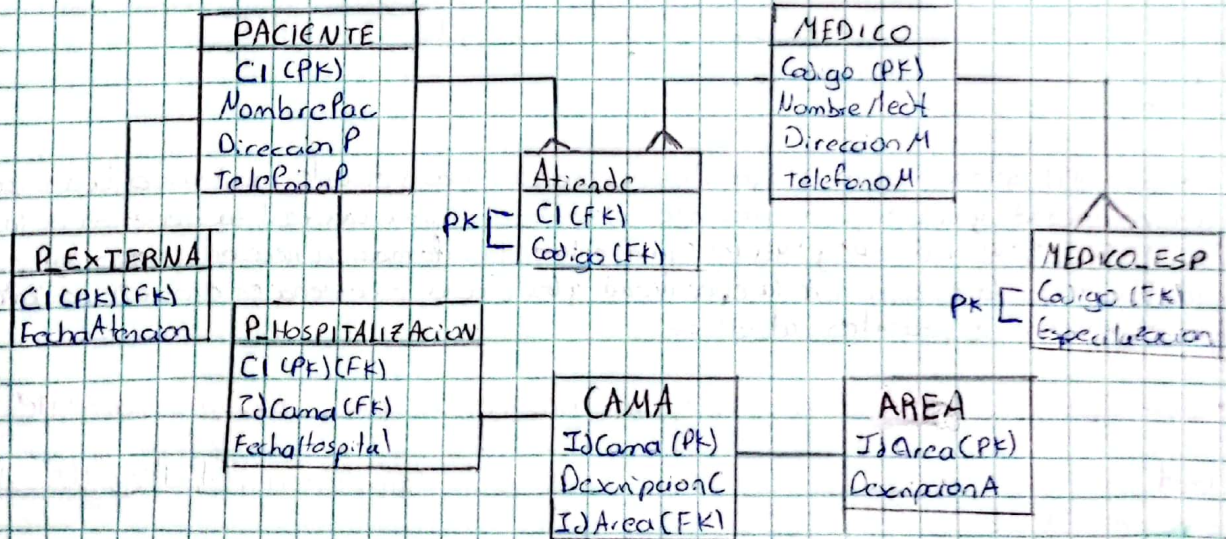
Fecha: 27/05/2024

2. - Objetivo

Manejar datos mediante el Lenguaje Estructuras de Consulta (SQL), sentencias de Selección.

3. - Instrucciones

Considere una base de datos relacional con las siguientes tablas y codifique las sentencias SQL para las consultas solicitadas.



Áreas para hospitalización: ID Área: 1: urgencias / 2: cuidados intensivos / 3: anestesiología / 4: cardiología

Código de Especialización de Médico... Especialización: cirujano cardiovascular / pediatra / endocrinólogo / general - familiar

4.- Actividades por Desarrollar

Con base al esquema anterior, codifique las sentencias SQL para las consultas solicitadas:

- Listado alfabético de los pacientes (NombrePac y CI) que se han hospitalizado y la cama (DescripcionC) en los que se ubicaron.

```
Select P.NombrePac as "Numero Paciente", P.CI as Cedula, C.DescripcionC as DescripcionC  
From PACIENTE as P  
Inner Join P.HOSPITALIZACION as PH on P.CI = PH.CI  
Inner Join CAMA as C on C.IdCama = PH.IdCama  
Group by P.NombrePac, P.CI, C.DescripcionC  
Order by P.NombrePac
```

El propósito general del código SQL es generar un informe detallado de pacientes hospitalizados, mostrando su nombre, cédula y la descripción de la cama asignada. Combina datos de las tablas 'PACIENTES', 'P-HOSPITALIZACION' y 'CAMA', agrupando los resultados para evitar duplicados. Finalmente ordena los datos por nombre del paciente en orden ascendente para una vista organizada.

- Conocer las 3 especialidades con mayor número de pacientes

```
Select Top 3 ME.Especializacion as Especialidad, count(A.CI) as "Numero Pacientes"  
From ATIENDE as A  
Inner Join MEDICO_ESP as ME on A.Codigo = ME.Codigo  
Inner Join MEDICO as M on ME.Codigo = M.Codigo  
Group by ME.Especializacion  
Order by "Numero Pacientes" Desc;
```

El código SQL selecciona las 3 especialidades médicas más frecuentes mostrando la descripción de la especialidad y la cantidad de pacientes atendidos por cada una. Realiza una unión de las tablas 'ATIENDE', 'MEDICO-ESP' y 'MEDICO' para obtener la información necesaria. Luego agrupa los resultados por la descripción de la especialidad y los ordena en orden descendente según el número de pacientes atendidos.

- Conocer los médicos que han atendido más de dos veces al mismo paciente

```
Select M.NombreMedt as Medico, P.NombrePac as Paciente, count(A.CI) as Num_Atencion  
From ATIENDE as A  
Inner Join MEDICO as M on M.Codigo = A.Codigo  
Inner Join PACIENTE as P on P.CI = A.CI  
Group by M.NombreMedt, P.NombrePac  
Having count(A.CI) > 2;
```

El código SQL cuenta y muestra las interacciones entre médicos y pacientes donde el número de atención es mayor que 2. Utiliza uniones entre las tablas 'ATIENDE', 'MEDICO' y 'PACIENTE' basadas en códigos de médico y cédulas de pacientes. Luego agrupa los resultados por el nombre del médico y del paciente.

- Conocer los pacientes de consulta externa que NO han sido hospitalizado

```

Select PE.CI, P.Nombre Pac as Paciente, PH.CI as Cedula
From P_EXTERNA as PE
Inner Join PACIENTE as P on P.CI = PE.CI
Left Join P_HOSPITALIZACION as PH on PE.CI = PH.CI
Where PH.CI is null

```

El código SQL selecciona pacientes externos y muestra su nombre, cédula y la cédula asociada a la hospitalización si no están hospitalizados. Utiliza uniones entre las tablas 'P_EXTERNA', 'PACIENTE' y 'P_HOSPITALIZACION' luego filtra los resultados para mostrar solo las visitas externas que no están en la tabla de hospitalización.

- Conocer los días (Fecha Hospital) del año 2023 que tuvieron más de 10 pacientes hospitalizados

```

Select Year(FechaHospital) as Año, day(FechaHospital) as Día,
count(Cday(FechaHospital)) as "Número Pacientes Hospitalizados"
From P_HOSPITALIZACION
Where Year(FechaHospital) = 2023
Group by Year(FechaHospital), day(FechaHospital)
Having count(Cday(FechaHospital)) > 10

```

El código SQL cuenta el número de pacientes atendidos por día en el año 2023 mostrando el año, día y la cantidad de pacientes utiliza la función 'Year()' para extraer el año 'Day()' para obtener pacientes atendidos.

- Conocer las áreas que no tuvieron camas ocupadas durante el mes de diciembre del 2021.

```

Select Count(C.IdCama) as "Cama Disponible", A.DescripcionA as "Área"
From AREA as A
Inner Join CAMA as C on A.IdArea = C.IdArea
Left Join P_HOSPITALIZACION as PH on C.IdCama = PH.IdCama
month(PH.FechaHospital) = 12 and year(PH.FechaHospital) = 2021
Group by A.IdArea, A.DescripcionA
Having Count(CPH.IdCama) = 0;

```

El código SQL cuenta el número de camas disponibles por área mostrando el área y la cantidad de camas libres realiza una unión entre las tablas 'AREA' y 'CAMA' y una unión izquierda con 'P_HOSPITALIZACION' filtrado por diciembre 2021 luego agrupa las áreas sin camas ocupadas.

- Conocer los médicos (Nombre Medt) con más de dos especialidades

```

Select M.Nombre Medt as Medico, count(ME.Especialidad) as "Número Especialidades"
From MEDICO as M
Inner Join MEDICO_ESP as ME on M.Codigo = ME.Codigo
Group by M.Nombre Medt
Having count(ME.Especialización) > 2;

```


El código SQL cuenta el número de especialidades por médico que tenga más de dos especialidades realiza una unión interna entre las tablas 'MEDICO' y 'MEDICO_ESP' basada en el código de médico luego agrupa los resultados por nombre de médico y cuenta las especialidades asociadas a cada uno. Finalmente filtra los resultados para incluir solo los médicos con más de dos especialidades.

- Listado en orden alfabético (Nombre Pac y CI) de todos los pacientes tanto de consulta externa como de hospitalización.

```
Select P.NombrePac as "Nombre Paciente", P.CI as "Cedula"
From P.HOSPITALIZACION as PH
Inner Join PACIENTE as P on P.CI = PH.CI
UNION
Select P.NombrePac, P.CI
From P.EXTERNA as PE
Inner Join PACIENTE as P on PE.CI = P.CI
Order by P.NombrePac
```

El código SQL fusiona pacientes de consulta externa y hospitalización mostrando nombre y cédula utiliza 'Union' para combinar resultados de consultas separadas: una para consulta externa y otra para la hospitalización. Cada consulta interna se une a la tabla de paciente 'PACIENTE' mediante 'INNER JOIN' con la tabla correspondiente 'P.EXTERNA' y 'P.HOSPITALIZACION' usando la cédula de identidad CI.

Recomendaciones

Antes de comenzar a escribir consultas SQL es crucial comprender completamente la estructura de la base de datos con la que estás trabajando esto incluye conocer las relaciones entre las tablas y entender el significado de cada columna.

Aprovecha las funciones como count(), Group by, MAX, MIN, AVG y otras funciones de agregación para realizar cálculos y análisis de datos más avanzados.

Después de escribir cada consulta verifica cuidadosamente los resultados para asegurarte de que sean precisos y cumpla con los requisitos de la actividad se puede comparar con ejercicios hechos en clases.

Conclusion

En conclusión el dominio de SQL es crucial para manejar eficientemente bases de datos a través de esta actividad propuesta se ha explorado desde consultas simples hasta consultas complejas abordando aspectos como la estructura de la base de datos la descomposición de requerimientos y la optimización de consultas.

Estas habilidades son fundamentales para gestionar y analizar los datos de manera efectiva lo que permite a los profesionales tomar decisiones informadas en diversos entornos laborales.