

# 1. Ejercicios Complementarios

Escribir las sentencias SELECT para obtener la siguiente informacion aplicando los conceptos de modularidad:

1. Relacion de prestamos efectuados por los prestatarios de una determinada comunidad.

```
WITH R(CodPrestatario, DocPrestamo) as (  
    SELECT P.CodPrestatario, P.DocPrestamo  
    FROM Prestamo P left outer join Amortizacion A on P.DocPrestamo = A.DocPrestamo  
    GROUP BY P.CodPrestatario, P.DocPrestamo  
    HAVING SUM(P.Importe) = SUM(IsNull(A.Importe, 0))  
)  
exec cr_CompletarPrestamos R;  
go
```

2. Relacion de prestatarios que hasta la fecha hayan efectuado mas de 5 prestamos.

```
WITH  
R (CodComunidad, CodPrestatario) as (  
    SELECT Pio.CodComunidad, Pio.CodPrestatario  
    FROM Prestatario Pio inner join Prestamo Pmo on Pio.CodPrestatario = Pmo.CodPrestatario  
)  
exec cr_CompletarPrestamos R;  
go
```

3. Relacion de prestatarios morosos, es decir, aquellos que aun no han cancelado alguna de sus deudas y ya paso la fecha de vencimiento.

```
WITH  
R (CodPrestatario) as (  
    SELECT Pio.CodPrestatario  
    FROM Prestatario Pio inner join Prestamo Pmo on Pio.CodPrestatario = Pmo.CodPrestatario  
    GROUP BY Pio.CodPrestatario  
    HAVING COUNT(Pmo.DocPrestamo) > 5  
)  
exec cr_CompletarPrestatarios R;  
go
```

4. Relacion de las 5 comunidades que tienen el mayor numero de prestatarios.

```
WITH  
R (CodPrestatario) as (  
    SELECT P.CodPrestatario  
    FROM Prestamo P left outer join Amortizacion A on P.DocPrestamo = A.DocPrestamo  
    GROUP BY P.DocPrestamo, P.CodPrestatario, P.FechaVencimiento, P.Importe  
    HAVING SUM(IsNull(A.Importe, 0)) < P.Importe and GetDate() > P.FechaVencimiento  
)  
exec cr_CompletarPrestatarios R;  
go
```

5. Relacion de comunidades cuyos prestatarios que aun tienen saldos, no hayan efectuado ninguna amortizacion en lo que va del año 2004.

```

WITH
R (CodComunidad, NroPrestatarios) as (
    SELECT TOP(5) C.CodComunidad, COUNT(P.CodPrestatario) as NroPrestatarios
    FROM Comunidad C inner join Prestatario P on C.CodComunidad = P.CodComunidad
    GROUP BY C.CodComunidad
    ORDER BY NroPrestatarios DESC
)
exec cr_CompletarComunidades R
go

```

6. Relacion de comunidades que no tengan prestatarios morosos.

```

WITH
R1 (CodPrestatario) as (
    SELECT P.CodPrestatario, P.DocPrestamo
    FROM Prestamo P left outer join Amortizacion A on P.DocPrestamo = A.DocPrestamo
    WHERE DATEPART(year, A.FechaCancelacion) = 2004
    GROUP BY P.DocPrestamo
    HAVING (SUM(P.Importe) > SUM(IsNull(A.Importe, 0)))
),
R2 (CodComunidad) as (
    SELECT P.CodComunidad
    FROM Prestatario P left outer join R1 on P.CodPrestatario = R1.DocPrestamo
)
exec cr_CompletarComunidades R2;
go

```

7. Relación de comunidades que tengan prestatarios morosos

```

WITH
R1 (CodPrestatario) as (
    SELECT P.CodPrestatario
    FROM Prestamo P left outer join Amortizacion A on P.DocPrestamo = A.DocPrestamo
    GROUP BY P.DocPrestamo, P.CodPrestatario, P.FechaVencimiento
    HAVING SUM(IsNull(A.Importe, 0)) < SUM(P.Importe) and GetDate() > P.FechaVencimiento
),
R2 (CodComunidad) as (
    SELECT P.CodComunidad
    FROM Prestatario P, R1
    WHERE P.CodPrestatario = R1.CodPrestatario
),
R3 (CodComunidad) as (
    SELECT CodComunidad
    FROM R1
    EXCEPT
    SELECT CodComunidad
    FROM R2
)
exec cr_CompletarComunidades R2;
go

```

## 2. Investigacion Formativa

Elaborar un resumen de las funciones internas TRANSACT SQL. Mas informacion en <https://learn.microsoft.com/en-us/sql/t-sql/functions/system-functions-transact-sql>

### 1. Funciones Agregadas

- SUM(), AVG(), COUNT(), MIN(), MAX(): Realizan cálculos en un conjunto de valores y devuelven un solo valor (e.g., suma, promedio, conteo, mínimo, máximo).

### 2. Funciones Analíticas

- ROW\_NUMBER(): Asigna un número de fila único dentro de la partición de un conjunto de resultados.
- RANK(), DENSE\_RANK(): Asignan un rango a cada fila en una partición, con o sin saltos.
- NTILE(): Divide el conjunto de resultados en un número especificado de grupos aproximadamente iguales.

### 3. Funciones de Manipulación de Bits

- BIT\_AND(), BIT\_OR(), BIT\_XOR(): Operan a nivel de bits en enteros para realizar operaciones AND, OR y XOR.

### 4. Funciones de Clasificación

- CUME\_DIST(), PERCENT\_RANK(): Calculan la posición relativa de una fila en una partición.

### 5. Funciones de Conjuntos de Filas

- STRING\_SPLIT(): Divide una cadena en una tabla de filas basada en un delimitador.

### 6. Funciones Escalares

#### a) Funciones de Cadena:

- LEN(), SUBSTRING(), CHARINDEX(): Operan sobre cadenas para obtener longitud, subcadenas, posición de subcadenas, etc.
- REPLACE(), UPPER(), LOWER(): Reemplazan subcadenas y cambian el caso de las letras.

#### b) Funciones de Fecha y Hora:

- GETDATE(), SYSDATETIME(): Devuelven la fecha y hora actuales.
- DATEADD(), DATEDIFF(): Manipulan fechas añadiendo intervalos o calculando diferencias.

#### c) Funciones Matemáticas:

- ABS(), CEILING(), FLOOR(): Realizan operaciones matemáticas básicas.
- ROUND(), SQRT(), POWER(): Redondean números, calculan raíces cuadradas y potencias.

#### d) Funciones de Conversión:

- CAST(), CONVERT(): Convierten datos de un tipo a otro.
- PARSE(), TRY\_PARSE(): Convierten cadenas en tipos de datos, con manejo de errores.

#### e) Funciones de Metadatos:

- OBJECT\_ID(), OBJECT\_NAME(): Recuperan el identificador y nombre de objetos en la base de datos.

#### f) Funciones de Seguridad:

- SUSER\_NAME(), SUSER\_SID(): Devuelven información sobre los inicios de sesión de SQL Server.