



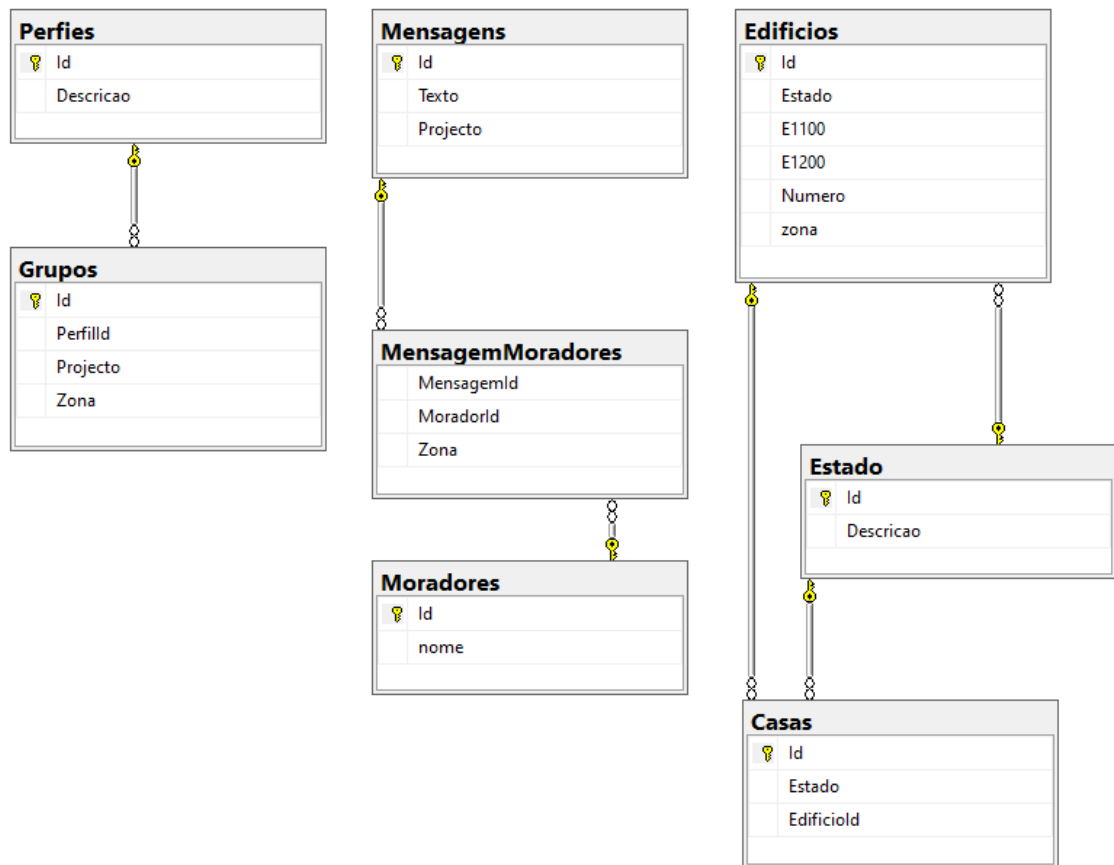
DXS
powered by **agap2IT**

Exercício SQL Server

04/01/2022

Data de entrega da resolução: 06/01/2022

Exemplo de modelo de dados



- 1) Mudarias algo no Modelo de dados? Se sim, enumera as razões e porquê?
- 2) Criar uma *View* que:
 1. Para cada *zona* deve devolver:
 - Número total de *edifícios* da zona
 - Número total de *casas* da zona
 - Número total de mensagens da zona
 - Qual é o *estado* do *edifício* que se repete mais vezes na *zona*.
 2. Apenas deve contabilizar *casas* com o *estado* 3 - *concluídos* e *edifícios* com o estado 3 - *concluídos* ou com o estado 2 - *progresso*.
 3. Só deve contabilizar zonas em que existam mais de 10 *mensagens* para essa *zona*.
- 3) Criar uma *function* que devolve o próximo número do *edifício* a criar. O número do *edifício* começa em E01 e vai até E99, aí deve mudar para F01 que vai até F99, G01 ... G99.

- 4) Criar um *stored procedure* que coloca na coluna *estado* da tabela *edificio* o valor 1 – *Novo*, 2 – *Em Progresso* ou 3 – *Concluído*.

Regras:

1. Se E1200 tiver o valor 1 ou se E1200 for *null* e o *edificio* não tiver *casas* então fica em 1 - *novo*.
2. Se o campo E1200 tiver o valor 1 e se o *edificio* contiver *casas* fica em 2 - *progresso*.
3. Se os campos E1100 e E1200 tiverem o valor 1 fica em 3 - *concluído*.

O *estado* das *casas* que existem no *edificio* deve mudar para o mesmo estado do *edificio*.

- 5) Partindo do pressuposto que a tabela *Mensagens* está ligada à tabela *Grupos* pela propriedade *Projeto*, cria um *stored procedure* que recebe um *perfil* e um *nome de morador* e devolve uma lista de *mensagens*.

Regras:

1. Se receber o *nome de morador* e *perfil* deve devolver *mensagens* para esse *morador* e para esse *perfil*.
2. Se receber apenas o *nome de morador*, deve devolver *mensagens* para esse *morador*.
3. Se não receber o *nome do morador*, deve devolver as *mensagens* para o *perfil* que recebeu.
4. Se o *perfil* for 2 – *Admin* deve devolver todas as *mensagens*.
5. Deve devolver sempre todas as *mensagens* de todas as *zonas* em que todos os *edifícios* estão no *estado* 3 – *concluído*.

SQL

```
/****** Object: Table [dbo].[Casas]    Script Date: 23/12/2021 20:48:26 *****/
CREATE TABLE [dbo].[Casas](
    [Id] [uniqueidentifier] NOT NULL,
    [Estado] [int] NULL,
    [EdificioId] [uniqueidentifier] NULL,
    CONSTRAINT [PK_casas] PRIMARY KEY CLUSTERED
(
    [Id] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF,
ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON, OPTIMIZE_FOR_SEQUENTIAL_KEY = OFF) ON
[PRIMARY]
) ON [PRIMARY]
GO
/****** Object: Table [dbo].[Edificios]    Script Date: 23/12/2021 20:48:26 *****/
CREATE TABLE [dbo].[Edificios](
    [Id] [uniqueidentifier] NOT NULL,
    [Estado] [int] NULL,
    [E1100] [bit] NULL,
    [E1200] [bit] NULL,
    [Numero] [nvarchar](3) NULL,
    [zona] [nvarchar](40) NULL,
    CONSTRAINT [PK_Edificios] PRIMARY KEY CLUSTERED
(
    [Id] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF,
ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON, OPTIMIZE_FOR_SEQUENTIAL_KEY = OFF) ON
[PRIMARY]
) ON [PRIMARY]
GO
/****** Object: Table [dbo].[Estado]    Script Date: 23/12/2021 20:48:26 *****/
CREATE TABLE [dbo].[Estado](
    [Id] [int] NOT NULL,
    [Descricao] [nvarchar](1) NOT NULL,
    CONSTRAINT [PK_Estados] PRIMARY KEY CLUSTERED
(
    [Id] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF,
ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON, OPTIMIZE_FOR_SEQUENTIAL_KEY = OFF) ON
[PRIMARY]
) ON [PRIMARY]
GO
/****** Object: Table [dbo].[Grupos]    Script Date: 23/12/2021 20:48:26 *****/
CREATE TABLE [dbo].[Grupos](
    [Id] [uniqueidentifier] NOT NULL,
    [PerfilId] [int] NOT NULL,
    [Proyecto] [varchar](25) NOT NULL,
    [Zona] [nvarchar](25) NULL,
    CONSTRAINT [PK_Grupos] PRIMARY KEY CLUSTERED
(
    [Id] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF,
ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON, OPTIMIZE_FOR_SEQUENTIAL_KEY = OFF) ON
[PRIMARY]
) ON [PRIMARY]
GO
/****** Object: Table [dbo].[MensajeMoradores]    Script Date: 23/12/2021 20:48:26
*****/
```

```

CREATE TABLE [dbo].[MensagemMoradores](
    [MensagemId] [uniqueidentifier] NOT NULL,
    [MoradorId] [uniqueidentifier] NOT NULL,
    [Zona] [nvarchar](25) NULL
) ON [PRIMARY]
GO
/***** Object: Table [dbo].[Mensagens]    Script Date: 23/12/2021 20:48:26 *****/
CREATE TABLE [dbo].[Mensagens](
    [Id] [uniqueidentifier] NOT NULL,
    [Texto] [varchar](50) NOT NULL,
    [Projecto] [varchar](25) NOT NULL,
    CONSTRAINT [PK_Mensagens] PRIMARY KEY CLUSTERED
(
    [Id] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF,
ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON, OPTIMIZE_FOR_SEQUENTIAL_KEY = OFF) ON
[PRIMARY]
) ON [PRIMARY]
GO
/***** Object: Table [dbo].[Moradores]    Script Date: 23/12/2021 20:48:26 *****/
CREATE TABLE [dbo].[Moradores](
    [Id] [uniqueidentifier] NOT NULL,
    [nome] [varchar](100) NOT NULL,
    CONSTRAINT [PK_Moradores] PRIMARY KEY CLUSTERED
(
    [Id] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF,
ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON, OPTIMIZE_FOR_SEQUENTIAL_KEY = OFF) ON
[PRIMARY]
) ON [PRIMARY]
GO
/***** Object: Table [dbo].[Perfies]      Script Date: 23/12/2021 20:48:26 *****/
CREATE TABLE [dbo].[Perfies](
    [Id] [int] NOT NULL,
    [Descricao] [nvarchar](1) NOT NULL,
    CONSTRAINT [PK_Perfies] PRIMARY KEY CLUSTERED
(
    [Id] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF,
ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON, OPTIMIZE_FOR_SEQUENTIAL_KEY = OFF) ON
[PRIMARY]
) ON [PRIMARY]
GO
ALTER TABLE [dbo].[Casas] WITH CHECK ADD CONSTRAINT [FK_Casas_Edificios] FOREIGN
KEY([EdificioId])
REFERENCES [dbo].[Edificios] ([Id])
GO
ALTER TABLE [dbo].[Casas] CHECK CONSTRAINT [FK_Casas_Edificios]
GO
ALTER TABLE [dbo].[Casas] WITH CHECK ADD CONSTRAINT [FK_Casas_Estado] FOREIGN
KEY([Estado])
REFERENCES [dbo].[Estado] ([Id])
GO
ALTER TABLE [dbo].[Casas] CHECK CONSTRAINT [FK_Casas_Estado]
GO
ALTER TABLE [dbo].[Edificios] WITH CHECK ADD CONSTRAINT [FK_Edificios_Estado]
FOREIGN KEY([Estado])
REFERENCES [dbo].[Estado] ([Id])
GO

```

```
ALTER TABLE [dbo].[Edificios] CHECK CONSTRAINT [FK_Edificios_Estado]
GO
ALTER TABLE [dbo].[Grupos] WITH CHECK ADD CONSTRAINT [FK_Grupos_Perfies] FOREIGN
KEY([PerfilId])
REFERENCES [dbo].[Perfies] ([Id])
GO
ALTER TABLE [dbo].[Grupos] CHECK CONSTRAINT [FK_Grupos_Perfies]
GO
ALTER TABLE [dbo].[MensajeMoradores] WITH CHECK ADD CONSTRAINT
[FK_MensajeMoradores_Mensagens] FOREIGN KEY([MensajeId])
REFERENCES [dbo].[Mensagens] ([Id])
GO
ALTER TABLE [dbo].[MensajeMoradores] CHECK CONSTRAINT
[FK_MensajeMoradores_Mensagens]
GO
ALTER TABLE [dbo].[MensajeMoradores] WITH CHECK ADD CONSTRAINT
[FK_MensajeMoradores_Moradores] FOREIGN KEY([MoradorId])
REFERENCES [dbo].[Moradores] ([Id])
GO
ALTER TABLE [dbo].[MensajeMoradores] CHECK CONSTRAINT
[FK_MensajeMoradores_Moradores]
GO
```