**6.1 Definições de tipos de dados em SQL**

A SQL usa os termos tabela linha e coluna para os termos do modelo relacional formal: relação, tupla e atributo respectivamente.

O principal comando SQL para a definição de dados é o CREATE, que pode ser usada para criar esquemas, tabelas, tipos e domínios, bem como outras construções como views, assertions e triggers.

**6.1.1 Conceitos de esquema e catálogo em SQL**

O conceito de um esquema SQL foi inicialmente incorporado com a SQL2, a fim de agrupar tabelas e outras construções que pertencem à mesma aplicação de banco de dados. Um esquema SQL é identificado por um nome de esquema, e inclui um identificador de autorização para indicar o usuário ou a conta proprietária do esquema, como descritores para cada elemento no esquema.

Esses elementos incluem tablas, tipos, restrições view, dominios e outras construções.

Um esquema é criado por meio da instrução CREATE SCHEMA, que pode incluir todas as definições dos elementos do esquema. Alternativamente, o esquema pode receber um nome e identificador de autorização, e os elementos podem ser definidos mais tarde. Por exemplo, a instrução a seguir cria um esquema chamado EMPRESA, pertencente ao usuário com identificador de autorização ‘Jsilva’, cada instrução termina com um ;



Em geral, todos estão autorizados a criar esquemas e elementos do esquema. O privilégio para criar esquemas, tabelas e outras construções deve ser concedido explicitamente as contas de usuários relevantes pelo ADM do sistema.

O conceito de catálogo também é utilizado. As instalações de BD normalmente possuem um ambiente e um esquema padrão, logo quando um usuário se conecta à instalação desse banco de dados, ele pode se referir diretamente às tabelas e a outras construções dentro desse esquema.

Um catalogo sempre contém um esquema especial, chamado INFORMATION\_SCHEMA, que provê informações sobre todos os esquemas no catálogo e todos os descritores e elementos nesses esquemas. As restrições de integridade como a integridade referencial, podem ser definidas entre as relações somente se existirem nos esquemas dentro do mesmo catálogo. Os esquemas dentro do mesmo catálogo também podem compartilhar certos elementos como definições de domínio.

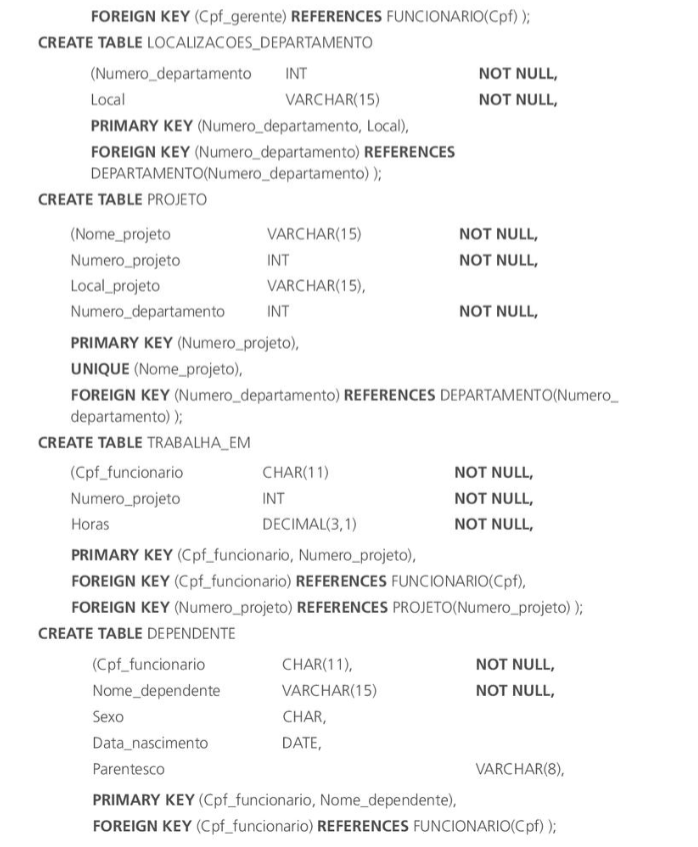
**6.1.2 O comando CREATE TABLE em SQL**

Este comando é usado para especificar uma nova relação, dando-lhe o nome e especificando seus atributos e restrições iniciais. Os atributos são especificados primeiro e cada um deles recebe um nome, um tipo de dado para especificar seu domínio de valores e quaisquer restrições de atributo, como NOT NULL.

As restrições de chave, integridade de entidade e integridade referencial podem ser especificadas na instrução CREATE TABLE, depois que os atributos forem declarados, ou acrescentadas depois, usando o comando ALTER TABLE.

Exemplo de CREATE TABLE:





O esquema SQL em que as relações são declaradas é especificado implicitamente no ambiente em que as instruções CREATE TABLE são executadas. Como alternativa, podem concectar explicitamente o nome do esquema ao nome da relação, separados por um ponto, como por exemplo, escrevendo:



As relações declaradas por meio das instruções CREATE TABLE são chamadas de tabelas de base, o que significa que a relação e suas tuplas são realmente criadas e armazenadas como um arquivo pelo SGBD. As relações de base são distintas das relações virtuais, criadas por meio da instrução CREATE VIEW.

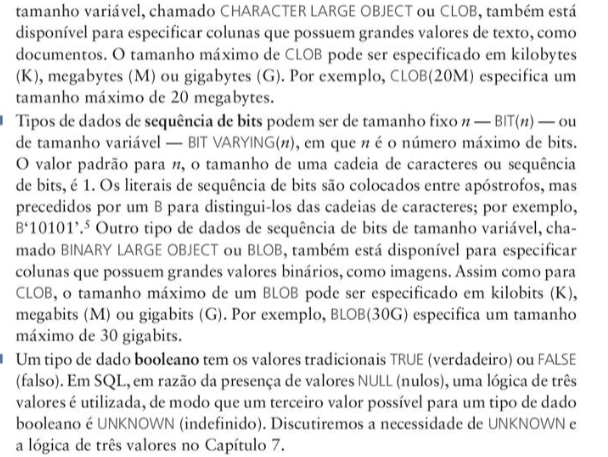
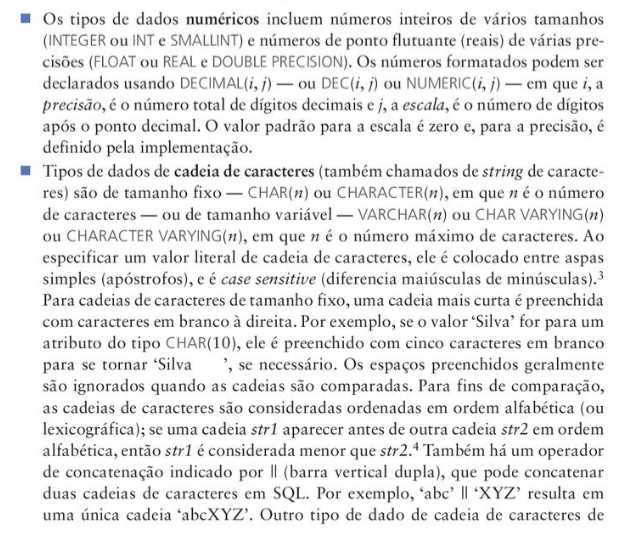
Em SQL, os atributos em uma tabela de base são considerados ordenados na sequência em que são especificados no comando CREATE TABLE. No entanto, as linhas não são consideradas ordenadas dentro de uma tabela.

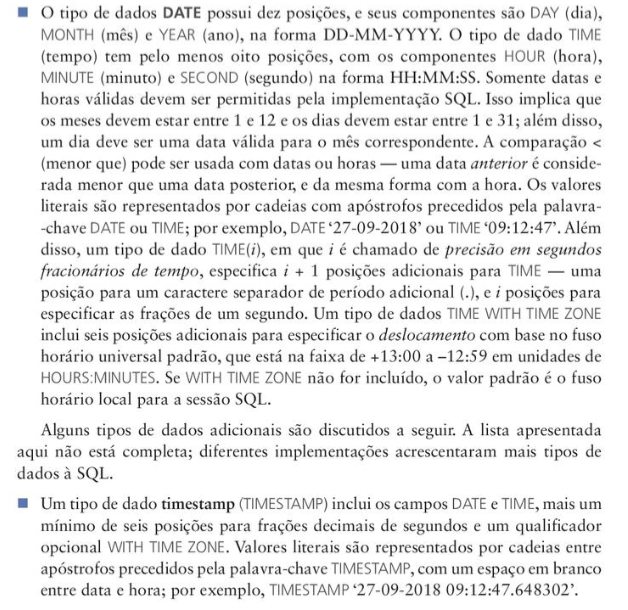
É importante observar, que nos comandos há algumas chaves estrangeiras, que podem causar erros, pois são especificadas por referências circulares ou por que dizem respeito a uma tabela que ainda não foi criada.

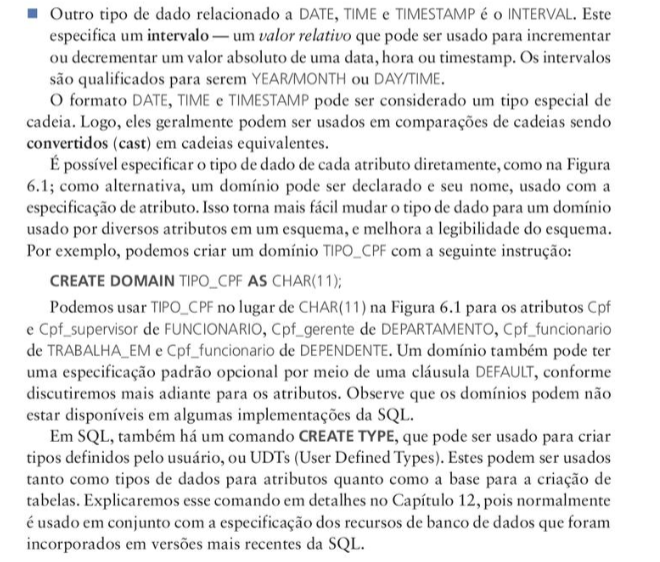
Por exemplo, a chave estrangeira Cpf\_supervisor na tabela FUNCIONARIO é uma referência circular, pois se refere à própria tabela. A chave estrangeira Numero\_departamento na tabela FUNCIONARIO se refere a tabela DEPARTAMENTO, que ainda não foi criada. Para lidar com esse tipo de problema, essas restrições podem ser omitidas do comando CREATE TABLE e depois acrscentadas usando a instrução ALTER TABLE.

**6.1.3 Tipos de dados de atributo e domínios em SQL**

Os tipos de dados básicos disponíveis para atributos são numéricos, cadeia ou sequência de caracteres, cada ou sequência de bits, booleano, data e hora.







**6.2 Especificando restrições em SQL**

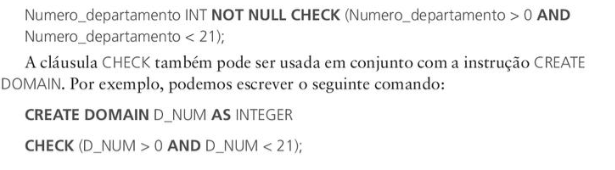
->Restrições de atributo e defaults de atributos

Como a SQL permite NULLs como valores de atributos, uma restrição NOT NULL pode ser especificada se o valor NULL não for permitido para determinado atributo. Isso sempre é especificado de maneira implícita para os atributos que fazem parte da chave primária de cada relação, mas pode ser especificado para quaisquer outros atributos cujos valores não podem ser NULL.

Também é possível definir um valor default para um atributo anexando a cláusula DEFAULT <valor> a uma definição de atributo. O valor default será incluído em qualquer nova tupla se um valor explícito não for fornecido para esse atributo. A figura abaixo, ilusta um exemplo de especificação deum gerente default para um novo departamento, e um departamento default para um novo funcionário. Caso nenhuma cláusula default for especificada, o valor default será NULL para atributos que não possuem a restrição NOT NULL



Outro tipo de restrição pode limitar valores de atributo ou domínio usando a cláusula CHECK ápos uma definição de atributo ou domínio. Por exemplo, suponha que números de departamento sejam restritos a números inteiros entre 1 e 20, então pode-se mudar a declaração de atributo de Numero\_departamento na tabela DEPARTAMENTO para:



Depois, podemos usar o domínio criado D\_Num como o tipo de atributo para todos os atributos que referem aos números de departamento, como Numero\_departamento de DEPARTAMENTO, Numero\_departamento de PROJETO e Numero\_departamento de FUNCIONARIO e assim por diante.

**6.2.2 Especificando restrições de chave e integridade referencial**

Como as restrições de chave e de integridade referencial são muito importantes existem cláusulas especiais dentro da instrução CREATE TABLE, para especificá-las. Alguns exemplos para ilustrar a especificação de chave e integridade referencial aparecem na figura principal. A cláusula PRIMARY KEY especifica um ou mais atributos que compõem a chave primária de uma relação. Caso uma chave primária tiver um único atributo, a cláusula, pode acompanhar o atributo diretamente. Por exemplo, a chave primária de DEPARTAMENTO pode ser especificada da seguinte forma:



A cláusula UNIQUE especifica chaves alternativas, também conhecidas como chaves candidatas. Essa cláusula, pode ser especificada diretamente para uma chave secundária, se esta for um unico atributo:



A integridade referencial é especificada por meio da cláusula FOREIGN KEY. Uma restrição de integridade referencial ṕde ser violada quando tuplas são inseridas ou excluídas, ou quando um valor de atributo de chave estrangeira ou chave primária é atualizado. A ação default que a SQL toma para uma violação de integridade é rejeitar a operação de atualização que causará uma violação, o que é conhecido como opção RESTRICT.

Porém o projetista do esquema pode especificar uma ação alternativa para ser tomada conectando um cláusula de ação de disparo referencial a quelaquer restrição de chave estrangeira.

As opções incluem SET NULL, CASCADE e SET DEFAULT. Uma opção deve ser qualificada com ON DELETE ou ON UPDATE. Na figura, o projetista escolhe ON DELETE SET NULL e ON UPDATE CASCADE para chave estrangeira Cpf\_supervisor de FUNCIONARIO. Isso significa que se a tupla para um funcionario supervisor for excluida, o valor de Cpf\_supervisor será automaticamente definido para NULL, para todas as tuplas de funcionários que estavam referenciando a tupla do funcionário excluído. Por sua vez, se o valor de CPF para um funcionário supervisor for atualizado, o novo valor será propagado em cascate de Cpf\_supervisor para todas as tuplas de funcionario que referenciam a tupla de funcionário atualizada.

**6.2.3 Dando nomes a restrições**

A palavra chave CONSTRAINT ilusta que as restrições podem receber um nome de restrição. Os nomes das restriçõs dentro de um esquema em particular precisam ser exclusivos, visto que é utilizado para identificar uma restrição em particular caso ela deva ser removida.

**6.2.4 Especificando restrições sobre tuplas usando CHECK**

Além das restrições de chave e integridade referencial, especificadas por plavras-chave especiais, outras restrições de tabela podem ser especificadas por meio de cláusula adicional CHECK ao final de uma instrução CREATE TABLE. Estas podem ser chamadas de restrições baseadas em tupla, visto que se aplicam a cada tupla individualmente e são verificadas sempre que uma tupa é inserida ou modificada.

Por exemplo, suponha que a tabela DEPARTAMENTO tivesse um atributo adicional Data\_criacao, que armazena a data em que o departamento foi criado, então poderiámos acrescentar a seguinte cláusular CHECK ao final da instrução CREATE TABLE para a tabela DEPARTAMENTO para garantir que a data de início de um gerente seja posterior à data de criação do departamento.

CHECK (Data\_criacao <= Data\_inicio\_gerente)

Essa cláusula, também pode ser usada para especificar restrições mais gerais usando a instrução CREATE ASSERTION da SQL.