



Developing Database

02– Data Integrity, Types e Variable Table

Sand Onofre

Sand.Onofre@FaculdadeImpacta.com.br

Sumário

- Data Integrity
 - Constraints Details
 - PK, FK, UQ, DF, CK
 - Identity, Sequence
- Internal Objects
 - Computed Columns
 - Metadados
 - Types
 - Variable Table
- Exercícios

Integridade de Dados - Tipos

- ***Integridade de Domínio***
 - Define quais serão os valores permitidos nas colunas (Data Type, Check, Default)

- ***Integridade de Entidade***
 - Garante a unicidade de cada linha da tabela (PK, UQ)

- ***Integridade Referencial***
 - Define o relacionamento entre as tabelas (FK)

Reforçando a Integridade de Dados

- ***Data Type***: define o tipo de dado que poderá ser armazenado na coluna
- ***Nulabilidade***: determina se algum valor DEVE necessariamente estar presente numa coluna
- ***Constraints***: definem regras que limitarão os valores que poderão ser armazenados na coluna/tabela ou como diferentes valores em colunas DEVEM estar relacionados.
- ***Triggers***: define o algoritmo (código) que será executado automaticamente quando a tabela é modificada

CONSTRAINTS DETAILS: PRIMARY KEY (PK)

- Torna a(s) coluna(s) com preenchimento obrigatório e atômico
- Não aceita duplicação de valores, não aceita NULL
- Permitido somente uma PK por tabela
- No SQL Server, dependendo de como for criada, gera a criação de Constraint e de Índice CLUSTERED
- Permite o relacionamento com FKs

CONSTRAINTS DETAILS: UNIQUE (UQ)

- Torna a(s) coluna(s) com preenchimento único
- Não aceita duplicação de valores, mas aceita um único elemento como NULL
- Permitido várias UQs por tabela
- No SQL Server, dependendo de como for criada, gera a criação de Constraint e de Índice
- Sendo uma Chave Candidata ou Alternativa, também permite o relacionamento com FKs

CONSTRAINTS DETAILS: FOREIGN KEY

- Torna a(s) coluna(s) com preenchimento relacionado a uma PK ou UQ de outra tabela
- Valores na coluna devem necessariamente EXISTIR na tabela referência
- Permite NULL
- Permitido várias FKs por tabela
- No SQL Server, gera a criação de Constraint mas NÃO cria nenhum índice
- Opção NO ACTION para CASCADE (on Delete, on Update)

CONSTRAINTS DETAILS: DEFAULT (DF)

- Torna o preenchimento da coluna AUTOMÁTICO, caso não forneçamos nenhum valor de entrada
- Atua somente na cláusula INSERT
- Não pode ser aplicado em coluna com IDENTITY
- Permite NULL, valor constante ou função escalar
- Os valores DEVEM ser compatíveis com o tipo de dados da coluna
- Permitido somente um DEFAULT por coluna da tabela

CONSTRAINTS DETAILS: CHECK (CK)

- Impõe restrição ou padronização no preenchimento da coluna e/ou tabela
- Atua nas cláusulas INSERT e UPDATE
- Permite NULL, valor constante, intervalo de valores e padrões de preenchimento
- Permitido vários CHECKs por coluna da tabela

Propriedade IDENTITY

- Automaticamente gera valores numéricos em colunas
- Podemos especificar um SEED (número inicial) e um INCREMENT
- O valor padrão (default) para SEED e INCREMENT é 1
- Apenas uma coluna da tabela poderá receber um IDENTITY e é intrínseco a esta tabela (não nomeamos)
- As funções SCOPE_IDENTITY() e @@IDENTITY retornam valores gerados nas colunas
- Em geral, não apontamos para coluna IDENTITY , INSERT ou UPDATE, a menos que utilizemos o SET IDENTITY_INSERT ON (apenas em casos de manutenção)

Objeto SEQUENCE

- Automaticamente gera valores numéricos em colunas
- Podemos especificar um START WITH (número inicial) e um INCREMENT BY
- Permite definir um MINVALUE, MAXVALUE e CYCLE (ao final da numeração, volta ao início)
- Várias colunas da tabela podem receber SEQUENCE e não é intrínseco a tabela
- Como é um objeto e não uma propriedade, recebe um nome e pode ser utilizado o mesmo SEQUENCE em várias tabelas
- Pode ser utilizado para migração de outros motores de banco de dados

COMPUTED COLUMN

- Coluna Computada é uma expressão de outras colunas da tabela.
- A expressão pode ser uma coluna não computada, função ou qualquer combinação com um ou mais operadores.
- A expressão não pode ser uma subquery.
- Pode ser usada em cláusulas SELECT, WHERE e ORDER BY ou qualquer lugar onde expressões regulares possam ser usadas
- Não podem ser alvos de declarações INSERT ou UPDATE
- **Exemplo:** Uma coluna de TOTAL pode ter sua definição como:

Total as (Qtd * Valor)

TYPE

- É possível a criação de um tipo de dados ALIAS definido pelo usuário
- Baseado em um tipo de dados nativo do SQL Server
- Podemos criar outros tipos de dados por CLR (Common Language Runtime)

VARIABLE TABLE

- Declaração de Variável com comportamento de tabela
- Permite aplicação de constraints (PK, UQ, DF e CK), mas sem nomeação explícita
- Como qualquer variável, seu escopo é local ao batch
- Só “existirá” no momento de execução do script - RUNTIME
- Localizada na memória, utilizar com parcimônia na inserção de registros

TYPE e VARIABLE TABLE

- Podemos criar um TYPE mais elaborado quando baseado em Variable Table
- Permite que objeto possa ser passado como parâmetro de STORED PROCEDURES
- Favorece a padronização de códigos e/ou objetos do banco

Exercícios com Variable Table

1. Elaborar uma Variável Table referente a **Cliente**, com os **menores tipos de dados possíveis** para armazenar as seguintes colunas:
 - nroCliente (100 clientes, numérico de geração automática, primary key)
 - nome (character até 60, unique)
 - dataNascimento (data, domínio: 18 a 50 anos)
 - cpf (11 caracteres fixos, obrigatório os 11 dígitos e entre 0 e 9)
 - sexo (domínio textual: Masculino ou Feminino)
 - estadoCivil (domínio: Solteiro (a), Casado (a) , Divorciado (a) , Viúvo (a), – Default: Solteiro (a))
 - nroFilhos (domínio: 0 a 20, Default: 0)

Nota: Os domínios deverão ter as restrições mencionadas assim como os padrões(default) sugeridos.

Exercícios com Variable Table

2. Teste de Restrições: Realizar 1 exemplo de inserção para cada domínio, provando que a tabela mantém a integridade definida nas regras de negócios.
3. Popular e testar a variável com os dados abaixo, listando o resultado (SELECT) :
 - Amadeu Abrantes (21 anos, data referente a idade, cpf(aleatório), Solteiro (a), 0 filhos)
 - Juliana Batista (36 anos, data referente a idade, cpf(aleatório), Casado (a), 2 filhos)
 - Sérgio Silva (47 anos , data referente a idade, cpf(aleatório), Viúvo (a), 4 filhos)
4. Query: Recuperar as seguintes informações

1a. coluna: <i>Label</i> : Sobrenome e Nome	Info: Sobrenome, Nome
2a. coluna: <i>Label</i> : Idade	Info: cálculo aproximado (apenas a diferença em anos entre a data de nascimento e a atual – utilizar a função datediff)
3a. coluna: <i>Label</i> : CPF	Info: cpf (colocar no formato 999.999.999-99)
4a. coluna: <i>Label</i> : Gênero	Info: sexo
5a. coluna: <i>Label</i> : Estado Civil	Info: estadoCivil
6a. coluna: <i>Label</i> : Filhos	Info: nroFilhos



Obrigado !

Sand Onofre
Sand.Onofre@FaculdadeImpacta.com.br