Banco de Dados II

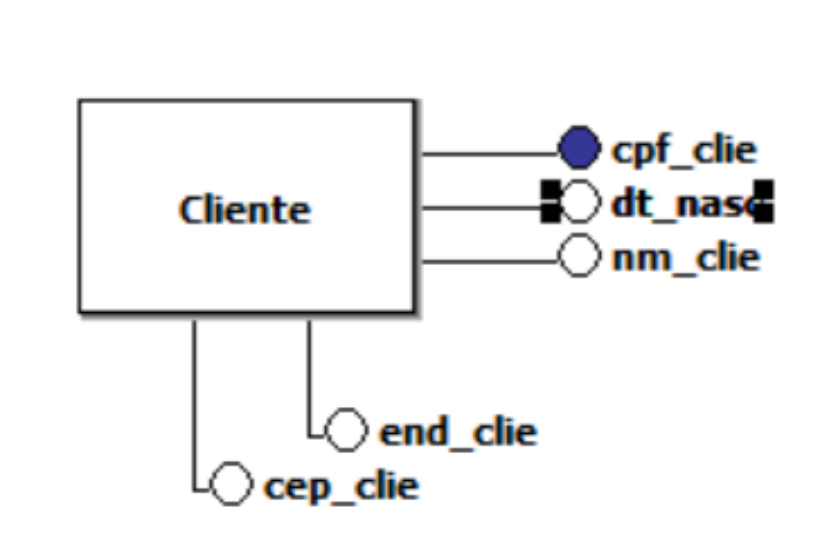
Professora Eliane Marion

ETEC PROF. HORÁCIO AUGUSTO DA SILVEIRA

SQL : DDL – Linguagem de Definição de Dados

Exercícios

1. Dado o Der, crie o MLR (sem código SQL)



Atributo	Tipo	Tamanho	Regra
cpf_clie	C	11	PK
dt_nasc	D	-	NN
nm_clie	A	30	NN
end_clie	A	50	-
cep_clie	C	8	-

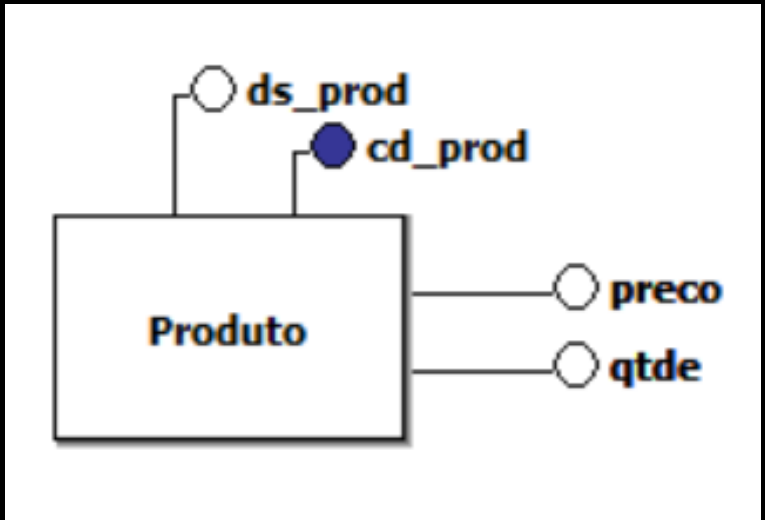
SQL : DDL – Linguagem de Definição de Dados

Correção:

```
create table cliente(
cpf_clie char(11) primary key,
dt_nasc date not null,
nm_clie varchar(30) not null,
end_clie varchar(50),
cep_clie char(8)
);
```

SQL : DDL – Linguagem de Definição de Dados

2. Dado o Der, crie o MLR (sem código SQL)



Atributo	Tipo	Tamanho	Regra
ds_prod	A	50	NN
cd_prod	N	6	PK
preco	N	10,2	NN
qtde	N	8,2	NN

SQL : DDL – Linguagem de Definição de Dados

2. Correção

```
create table produto(
ds_prod varchar(50) not null,
cd_prod numeric(6) primary key,
end_clie numeric(12,2) not null,
cep_clie numeric(10,2) not null
);
```

SQL : Tipos de Dados

Tipo de Dados: String

Tipo de Dados	Descrição	Tamanho Máximo	Tamanho (bytes)
char(n)	Tamanho fixo, completado com espaços em bracos	8,000 caracteres	Tamanho Definido
varchar(n)	Tamanho variável com limite	8,000 caracteres	2 bytes + número de caracteres
varchar(max)	Tamanho variável com limite	1,073,741,824 caracteres	2 bytes + número de caracteres
text	Tamanho variável	2GB de dados (texto)	4 bytes + número de caracteres
nchar	Tamanho fixo com espaços em bracos	4,000 caracteres	Tamanho definido x 2
nvarchar	Tamanho variável	4,000 caracteres	
nvarchar(max)	Tamanho variável	536,870,912 caracteres	
ntext	Tamanho variável	2GB de texto	
binary(n)	Tamanho fixo (binário)	8,000 bytes	
varbinary	Tamanho variável (binário)	8,000 bytes	
varbinary(max)	Tamanho variável (binário)	2GB	
image	Tamanho variável (binário)	2GB	

SQL : Tipos de Dados

Tipo de Dados: Numéricos

Tipo de Dado	Descrição	Tamanho (bytes)
bit	Número Inteiro que pode ser 0, 1 ou NULL	
tinyint	Permite números inteiros de 0 a 255	1 byte
smallint	Permite números inteiros entre -32,768 e 32,767	2 bytes
int	Permite números inteiros entre -2,147,483,648 e 2,147,483,647	4 bytes
bigint	Permite números inteiros entre -9,223,372,036,854,775,808 e 9,223,372,036,854,775,807	8 bytes
decimal(p,s)	Precisão de número flutuante e número de escala. Permite número de -10^38 +1 a 10^38 –1. O parâmetro p indica o número total máximo de dígitos que podem ser armazenados (ambos à esquerda e à direita do ponto decimal). p deve ser um valor de 1 a 38. O padrão é 18. O parâmetro s indica o número máximo de dígitos armazenados à direita do ponto decimal. s deve ser um valor de 0 a p. O valor padrão é 0.	5-17 bytes
numeric(p,s)	Precisão de número flutuante e número de escala. Permite número de -10^38 +1 a 10^38 –1. O parâmetro p indica o número total máximo de dígitos que podem ser armazenados (ambos à esquerda e à direita do ponto decimal). p deve ser um valor de 1 a 38. O padrão é 18. O parâmetro s indica o número máximo de dígitos armazenados à direita do ponto decimal. s deve ser um valor de 0 a p. O valor padrão é 0	5-17 bytes
smallmoney	Tipo de "Moeda" de -214,748.3648 a 214,748.3647	4 bytes
money	Tipo de "Moeda" de -922,337,203,685,477.5808 a 922,337,203,685,477.5807	8 bytes
float(n)	Precisão de número flutuante de -1.79E + 308 a 1.79E + 308.O parâmetro n indica se o campo deve conter 4 ou 8 bytes. float (24) contém um campo de 4 bytes e o float(53) mantém um campo de 8 bytes. O valor padrão de n é 53.	4 ou 8 bytes
real	Precisão de número flutuante de -3,40E + 38 a 3,40E + 38	4 bytes

SQL : Tipos de Dados

Tipo de Dados: Data

Tipo de Dado	Descrição	Tamanho (bytes)
datetime	De 1 de janeiro de 1753 a 31 de dezembro de 9999 com uma precisão de 3,33 milisegundos	8 bytes
datetime2	De 1º de janeiro de 0001 a 31 de dezembro de 9999 com precisão de 100 nanossegundos	6-8 bytes
smalldatetime	De 1 de janeiro de 1900 a 6 de junho de 2079 com precisão de 1 minuto	4 bytes
date	Armazena apenas uma data. De 1 de janeiro de 0001 a 31 de dezembro de 9999	3 bytes
time	Armazena um tempo apenas para uma precisão de 100 nanosegundos	3-5 bytes
datetimeoffset	O mesmo que datetime2 com a adição de um deslocamento de fuso horário	8-10 bytes
timestamp	Armazena um número único que é atualizado sempre que uma linha é criada ou modificada. O valor do timestamp é baseado em um relógio interno e não corresponde ao tempo real. Cada tabela pode ter apenas uma variável timestamp	

SQL : Constraints

São regras agregadas a colunas ou tabelas.

Tipo de Constraint	Função	Sintaxe
Primary Key	Permite identificar um único registro	`Primary Key (coluna)`
Not Null	Indica que o conteúdo de uma coluna não poderá ser Nulo	`NOT NULL`
Unique	Indica que não pode haver repetição no conteúdo da coluna	`UNIQUE`
Default	Serve para atribuir um conteúdo padrão para uma coluna	`DEFAULT VALOR`
Check	Defini possíveis valores para o conteúdo de um campo	`Sexo char(1) CHECK (Upper(sexo)='M' OR Upper(sexo) = 'F')`

SQL : Constraints

Chave Estrangeira: Foreign Key

É o campo que estabelece o relacionamento entre duas tabelas.

Sintaxe:

```
FOREIGN KEY  nome-chave-estrangeira (lista-de-colunas)
      REFERENCES nome-tabela (lista-de-colunas)
```

Onde:

Opção	Descrição
nome-chave-estrangeira	Nome opcional da constraint
lista-de-colunas	Lista de colunas da tabela que faz referência a outra tabela
Nome-tabela	Nome da tabela em que está a chave primária.

SQL : DDL – Linguagem de Definição de Dados

Exemplos de criação de tabelas:

Sem regras

```
CREATE TABLE  cargo(
      cod_cargo numeric(4),
      ds_cargo varchar(30),
      salario numeric(12,2)
);
```

Com regras, mas sem personalização:

```
CREATE TABLE  cargo(
      cod_cargo numeric(4) primary key,
      ds_cargo varchar(30) not null,
      salario numeric(12,2) not null
);
```

SQL : DDL – Linguagem de Definição de Dados

Exemplos de criação de tabelas:

Com regras, e personalização no nome da regra

```
CREATE TABLE  cargo(
      cod_cargo numeric(4) constraint cargo_cdPK primary key,
      ds_cargo varchar(30) constraint cargo_dsNN not null,
      salario numeric(12,2) constraint cargo_salNN not null
);
```

SQL : DDL – Linguagem de Definição de Dados

Exercício

Criar as tabelas Cds e Musicas no SQL.

Tabelas de Referências

Cds

Campo	Tipo	Tamanho	Chave
Codigo	N	3	*
Nome	A	50	
DataCompra	D		
ValorPago	N	8,2	
LocalCompra	A	20	
Album	C	1	

Musicas

Campo	Tipo	Tamanho	Chave
CodigoCD	N	3	*
Numero	N	2	*
Nome	A	50	
Artista	A	50	
Tempo	N	4	

SQL : DDL – Linguagem de Definição de Dados

02. Criar a tabela EMPR (Empregados), conforme a especificação abaixo.

Campo	Tipo	Nulo	Descrição	Tamanho	Chave
MATR	Char	Não	Matrícula única do empregado	6	PK
NOME	VARCHAR	Não	Primeiro nome	12	
SOBRENOME	VARCHAR	Não	Sobrenome	15	
DEPT	CHAR		Código de departamento do empregado	3	FK
FONE	CHAR		Número do telefone	14	
DTADMIN	DATE		Data de admissão		
CARGO	CHAR		Cargo do empregado	10	

SQL : DDL – Linguagem de Definição de Dados

Continuação da tabela EMPR (Empregados)

Campo	Tipo	Nulo	Descrição	Tamanho	Chave
EXPROFI	NUMBER		Experiência em anos	2	
SEXO	CHAR			1	
DATANASC	DATE		Data de nascimento		
SALARIO	NUMBER		Salário anual	9,2	
BONUS	NUMBER		Bônus anual	9,2	
COMIS	NUMBER		Comissão anual	9,2	

SQL : DDL – Linguagem de Definição de Dados

03. Criar a tabela DEPT (Departamento), conforme a especificação abaixo.

Campo	Tipo	Nulo	Descrição	Tamanho	Chave
DCODIGO	Char	Não	Código único do departamento	3	PK
DNOME	VARCHAR	Não	Nome do departamento	36	
GERENTE	CHAR		Matrícula do Gerente	6	FK
DSUPER	CHAR		Departamento do supervisor	3	

SQL : DDL – Linguagem de Definição de Dados

04. Criar a tabela PROJETO (Projetos), conforme a especificação abaixo.

Campo	Tipo	Nulo	Descrição	Tamanho	Chave
PCODIGO	Char	Não	Código único do projeto	6	PK
PNOME	VARCHAR	Não	Nome do projeto	24	
DCODIGO	CHAR	Não	Código do departamento	3	FK
RESP	CHAR	Não	Matrícula do responsável	6	FK
EQUIPE	NUMBER		Número de empregados no projeto	5	
DATAINI	DATE		Data de início		
DATAFIM	DATE		Data da finalização do projeto		
PSUPER	CHAR		Supervisor do projeto (funcionário)	6	FK