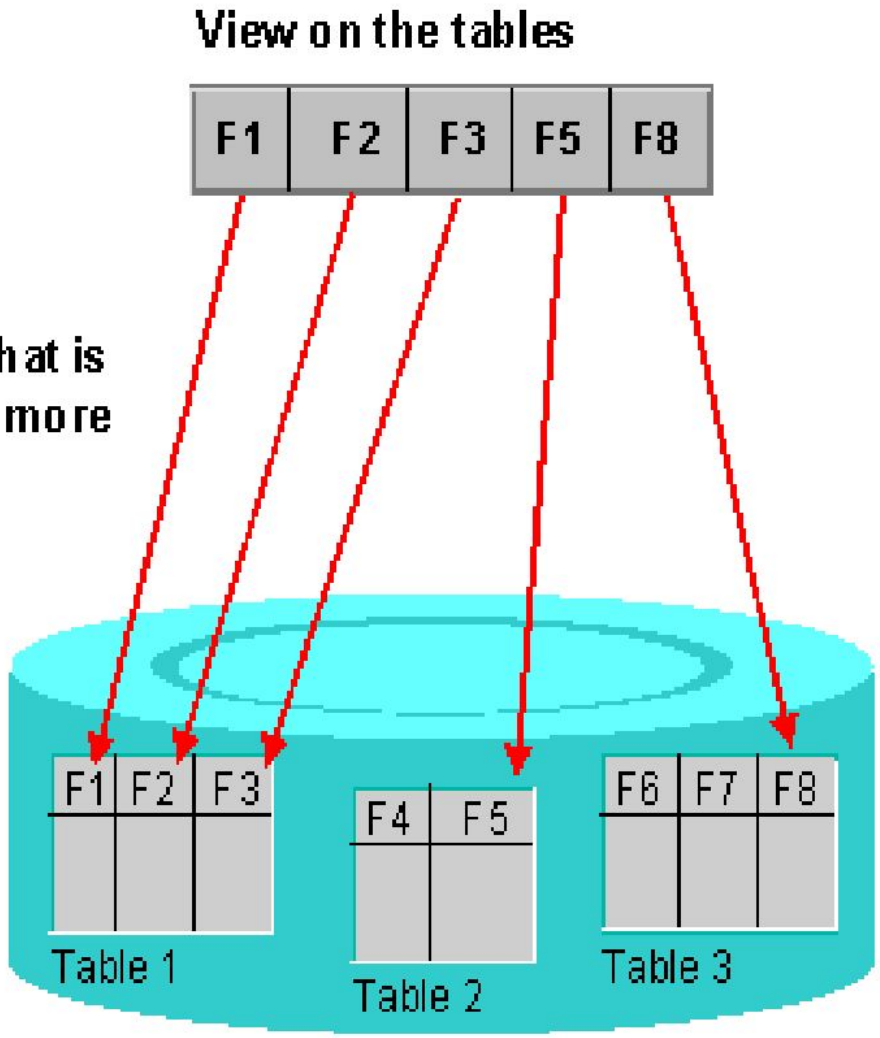
Visões em Banco de Dados/ Restrições de Integridade

Crícia Felício

Visões

O que são?

View on data that is distributed on more than one table



Sintaxe

Create view **nome_da_visao(col1,col2, ...)** As
Select col1,col2...-- nome das colunas na tabela base

From tabela1, tabela2 ...

Where condicoes ...

* Obs: Se não for informado o nome das colunas da visão, ela será criada com os mesmos nomes da tabela de origem.

Como definir(criar) uma visão?

* Exemplo: BD empresa
* – Criar uma visão emps_inf com o nome, sexo e data de admissão dos **funcionários** que são do **departamento** de **informática**

```
Create view vw_emps_inf as
Select f.nome,f.sexo,f.data_adm
From funcionario f, departamento d
Where d.nome='Informatica' and f.cod_depto=d.cod_depto;
```

Tabela virtual

Tabelas base

Visões

Visões (view) = Tabelas virtuais

- Criadas a partir de:
  - Outras tabelas do banco de dados;
  - Outras visões.
- Seu conteúdo é obtido através de:
  - Consultas sobre uma ou mais tabelas (visões);
  - Não são armazenadas fisicamente.

* O SGBD armazena a sua definição.

Como acessar o conteúdo de uma visão?

* Da mesma forma que é feito o acesso ao conteúdo de uma tabela.
* Exemplo: Listar os dados dos funcionários que foram contratado a mais que 10 anos

```
Select * From vw_emps_inf
Where  data_adm<date_sub(current_date, Interval 10 year);
```

Vantagens

* Por que usar visões?
  * Privacidade
  * Foco e personalização de dados
  * Uma tabela base pode ser vista de diferente formas
  * Controle de acesso aos dados pode ser restrito
    * Determinadas linhas e a determinadas colunas

Vantagens

* O acesso pode ser restrito a apenas resumos estatísticos da tabela
  * Somas, médias, contagens, etc.
* O acesso pode ser restrito a um subconjunto

de uma outra visão.

* A visão pode ser definida por uma consulta complexa.

Características

O conteúdo da visão é sempre atual.

- Uma visão pode ser definida a partir de uma outra visão.
- A definição da visão é armazenada no catálogo do sistema.
- A visão tem um comportamento semelhante a de uma tabela.

– Nem sempre é possível atualizá-la (UPDATE, INSERT, DELETE)

Criação de visões

* O setor de cardiologia de um hospital lida somente com pacientes cardiacos ou hipertensos.
* Dessa forma, o DBA pretende disponibilizar através de uma visão somente os dados dos pacientes com esses problemas

Criação de visões

Create view vw_cardio_hipertenso(codigo, nome, datanasc) as

Select codp, nomep,dt_nasc from paciente

Where problema='cardiaco' or problema='hipertensao';

Acesso aos dados

Select * from vw_cardio_hipertenso;

é semelhante a:

Select codp, nomep,dt_nasc from paciente

Where problema='cardiaco' or problema='hipertensao';

Exemplo

* Crie uma visão com os pacientes com mais de 60 anos

```
Create view vw_aposentados as
Select * From paciente
Where date_add(dt_nasc,Interval 60
year)<=current_date();
/* date_add soma um valor em dias,meses ou ano com
uma determinada data */
/*current_date() : retorna a data atual */
```

Visões

* Definições recursivas são permitidas, ou seja, é possível criar uma visão a partir de outra visão:

Exemplo: Crie uma visão com as informações dos pacientes cardiacos ou hipertensos que tenham idade menor ou igual a 40 anos

```
create view vw_cardio_hipertenso_jovem as
 select * from vw_cardio_hipertenso
where date_add(dataNasc, Interval 40 year) >= current_date();
```

Visões

```
insert into paciente(nomep,sexo,dt_nasc,problema)
values ('Joao Pedro', 'm', '2005-03-06','cardiaco');
Select * from vw_cardio_hipertenso_jovem;
/* os dados do Joao Pedro também serão listados por
esse select */
```

Exercício

* Usando BD Empresa

- Crie uma visão DadosFunc que contenha o nome do funcionario, a data de admissao, o nome do departamento, o nome do cargo e o salário.

- Criar uma visao funcionariosPorDepto, que contenha o nome do departamento e a quantidade de funcionarios que trabalham nele

Exercício

```
create view DadosFunc as
select f.nome nomefunc,f.data_adm,d.nome
nomeDepto,c.nome nome_cargo,c.salario
from funcionario f join departamento d
on f.cod_depto=d.cod_depto join
cargo c on f.cod_cargo=c.cod_cargo;
```

Exercício

```
create view funcionariosporDepto as
select d.nome nmDepto, count(*) qtdeFunc
from funcionario f join departamento d
 on f.cod_depto=d.cod_depto
 group by d.cod_depto;
```

Apagar visões

Drop view nomedavisao;

Exclui a definição da visão do catálogo do banco de dados.

Exemplo:
Drop view vw_cardio_hipertenso_jovem ;

Atualização de visões

* Atualizações na visão refletem na tabela base

• Atualizações que não são possíveis de serem realizadas, são rejeitadas.

- Ambiguidades, restrições de integridade, etc impedem a realização de uma atualização

Atualizações nas visões

* Funciona: **Reflete na tabela base.**

```
update vw_cardio_hipertenso
set nome='Maria Paula Carvalho'
where nome='Maria Paula';
```

* Não Funciona:

```
update funcionariosPorDepto
set nmdepto='Adm';
```

Essa atualização não pode ser realizada porque a definição da view possui funções agregadas e agrupamentos:

Restrições para atualização de visões

U(Update),I(Insert), D(Delete)

- Colunas obtidas através de um cálculo não podem ser atualizadas.
  - Não permitem U, I e D.
  - Exemplo: salário+gratificação
- Visões compostas por funções agregadas e/ou agrupamento.
  - Não permitem U,I e D.

Restrições para atualização de visões

A inserção de uma nova linha na visão não pode violar a restrição de chave primária

- Inserções devem incluir todas as colunas not null da tabela base.
- Se a visão foi obtida através de uma junção:

– Só é possível atualizar os dados de uma das tabelas por vez.

Restrições para atualização de visões

Os comandos abaixo irão funcionar?

update dadosFunc

set NomeFunc="Ana Silveira Mendes" where NomeFunc="Ana Silveira";

update dadosFunc

set NomeFunc="Ruth Souza Silva",NomeDepto="Setor de Servicos" where NomeFunc="Ruth Souza";

Anomalias nas atualizações de visões

É possível atualizar uma linha da visão e a linha atualizada não fazer mais parte da visão.

- Exemplo:

Exemplo: Atualizar a data de nascimento do paciente da visão PacCardioJovens

```
update vw_cardio_hipertenso_jovem
 set dataNasc="1975-01-05"
where nome="Joao Pedro";
```

Anomalias nas atualizações de visões

É possível inserir uma linha na visão e a nova linha não fazer parte da visão

```
insert into vw_cardio_hipertenso_jovem
(nome,dataNasc)
values("Joao Paulo","1970-02-03");
```

Nesse caso, o dados do paciente Joao Paulo só aparecerão na tabela de pacientes.

Restrição de Integridade

Restrição de Integridade

- São regras que servem para prevenir a entrada de informações inválidas por parte do usuário.
- Procuram evitar inconsistências nos dados armazenados no banco de dados

Restrição de Integridade

* NOT NULL
* Default
* UNIQUE
* CHECK
* PRIMARY KEY
* FOREIGN KEY
* AUTO_INCREMENT
* SET
* ENUM

Para evitar valores nulos

- Ao criar a tabela utilizar a cláusula NOT NULL, para evitar a inserção de valores nulos
  - Exemplo

- Create database aula_bda;
- Use aula_bda;

```
create table curso(
Id_curso int auto_increment,
Nome_curso varchar(20) not null,
Primary key(id_curso));
```

De acordo com a definição da tabela, o campo nome_curso não aceita valor nulo

Para evitar valores nulos

- Em alguns casos é possível definir um valor default para o campo utilizando a cláusula DEFAULT
  - Usado para inserir um valor default em uma coluna
  - Define um valor de preenchimento da coluna, caso não seja fornecido um valor durante a operação de inserção
    - Campos que aceitam valores nulos, são criados com default NULL

DEFAULT

- Exemplo:

```
create table venda(
cod_venda int auto_increment,
data_venda date,
local_venda varchar(30) default "Uberlandia",
primary key(cod_venda));
Insert into venda(data_venda) values (current_date);
```

Para evitar valores duplicados

- Cláusula UNIQUE
- Garante que haverá um único valor para cada registro inserido
  - Não precisa ser chave primária
  - É definido utilizando a palavra UNIQUE na frente do campo durante a criação da tabela
  - Exemplo:
    - A tabela projeto deve armazenar o código, sigla e nome do projeto
    - O nome de projeto não pode repetir

UNIQUE

- Exemplo

```
Create table projeto(
cod_projeto int auto_increment,
sigla_projeto varchar(10),
Nome_projeto varchar(20) unique,
Primary key(cod_projeto));
Insert into projeto(sigla_projeto,nome_projeto) values
('SisFin','Sistema Financeiro');
```

Se tentarmos inserir o mesmo nome de projeto novamente ocorrerá um erro:

```
Insert into projeto(sigla_projeto,nome_projeto) values
('SisFin','Sistema Financeiro');
```

UNIQUE

- Exemplo: Definindo a tabela filme

```
Create table filme(
cod_filme int auto_increment,
titulo varchar(30),
ano numeric(4),
diretor varchar(40),
Primary key(cod_filme),
Unique (titulo,ano));
```

UNIQUE

* /* essa inserção irá funcionar pois o titulo é o mesmo mas o ano é diferente*/
* Insert into filme(titulo,ano,diretor) values ('Titanic',1980,'Spielberg'),('Titanic',1994,'Hitckock');
* /*Porem a tentativa de inserção de um mesmo titulo e ano irá gerar um erro */
* Insert into filme(titulo,ano,diretor) values ('Titanic',1980,'Spielberg') ;

Para evitar valores inválidos

- Há situações onde o valor de um campo deve ficar restrito a um conjunto de valores
  - A cláusula CHECK faz essa verificação através de uma expressão condicional
    - Exemplo

```
CREATE TABLE aluno (
Cod_aluno int auto_increment primary key,
nome VARCHAR(50) NOT NULL,
sexo CHAR(1) CHECK(sexo IN ("M", "F")));
```

Chave Primária

- Quando um campo é criado como primary key, ele é definido como NOT NULL e UNIQUE

Restrições de Integridade Referencial

- A chave estrangeira referencia a chave primária de uma outra tabela
  - Regra da integridade referencial
    - Deve garantir que inserções ou alterações gere a ocorrência de valores para o campo que é chave estrangeira que não possuem correspondência na chave referenciada
    - Definida durante a criação da tabela
- Exemplo
- Create table departamento (cod_depto int auto_increment primary key, nome_depto varchar(50));
- Drop table curso;

```
create table curso(
Id_curso int auto_increment primary key,
Nome_curso varchar(20) not null,
Cod_depto int,
foreign key(cod_depto) references departamento(cod_depto));
```

Restrições de Integridade Referencial

- Como controlar as alterações?
  - Tratamento Default
    - Não permitir que alterações ou remoções na tabela referenciada gere tuplas orfãs
    - Corresponde as cláusulas ON DELETE RESTRICT ON UPDATE RESTRICT

Restrições de Integridade Referencial

- ON DELETE SET NULL e ON UPDATE SET NULL
  - ON DELETE SET NULL: Se o valor do atributo for deletado na tabela referenciada, ele é preenchido como null na tabela onde o campo é chave estrangeira
  - ON UPDATE SET NULL: Se o valor do atributo for atualizado na tabela referenciada, ele é preenchido como null na tabela onde o campo é chave estrangeira

Restrições de Integridade Referencial

- Exemplo

```
Drop table curso;
create table curso(
Id_curso int auto_increment primary key,
Nome_curso varchar(20) not null,
Cod_depto int,
foreign key(cod_depto) references departamento(cod_depto)
on delete set null
on update set null);
```

Restrições de Integridade Referencial

Insert into departamento(nome_depto)
Values ('Informatica');
Insert into curso(nome_curso,cod_depto)
Values('Sistemas',1),('Licenciatura',1) ;
Delete from departamento where cod_depto=1;
Como o departamento de codigo=1 foi apagado, os cursos que referenciavam esse código devem ter sido preenchidos com null
select * from curso;

Restrições de Integridade Referencial

- ON DELETE CASCADE e ON UPDATE CASCADE
  - ON DELETE CASCADE: A operação de delete na tabela referenciada gera a remoção dos registros que possuem como chave estrangeira o valor removido
  - ON UPDATE CASCADE: A operação de alteração na tabela referenciada gera a alteração dos registros que possuem como chave estrangeira o valor alterado

Restrições de Integridade Referencial

- Exemplo

```
Drop table curso;
create table curso(
Id_curso int auto_increment primary key,
Nome_curso varchar(20) not null,
Cod_depto int,
foreign key(cod_depto) references
    departamento(cod_depto)
on delete cascade
on update cascade);
```

Restrições de Integridade Referencial

```
Insert into departamento
Values (2,'Gestão');
select * from departamento;
Insert into curso(nome_curso,cod_depto)
Values('Marketing',2), ('Logistica',2) ;
Update departamento set cod_depto=3 where cod_depto=2;
select * from curso;
Delete from departamento where cod_depto=1;
 /*todos os cursos de cod_depto=1 também serão apagados*/
```

Exemplo – Criação de Tabelas c/ Restrições de Integridade

cargo

CdCargo	NmCargo	VrSalário
C1	COZINHEIRA	350
C3	AUX. ESCRITÓRIO	450
C7	VIGIA	400
C2	MECANICO	750
C5	GERENTE	2300
C4	ESCRITURARIO	600

depto

CdDepto	NmDepto	Ramal
D1	ADMINISTRACAO	221
D2	OFICINA	235
D3	SERVICOS GERAIS	243
D4	VENDAS	258

funcionário

NrMatric	NmFunc	DtAdm	Sexo	CdCargo	CdDepto
1001	JOAO SAMPAIO	10/08/93	M	C2	D2
1004	LUCIO TORRES	02/03/94	M	C2	D2
1034	ROBERTO PEREIRA	23/05/92	M	C3	D1
1021	JOSE NOGUEIRA	10/11/94	M	C3	D1
1029	RUTH DE SOUZA	05/01/92	F	C1	D3
1095	MARIA DA SILVA	03/09/92	F	C4	D1
1023	LUIZ DE ALMEIDA	12/01/93	M	C2	D2
1042	PEDRO PINHEIRO	29/07/94	M	C4	D1
1048	ANA SILVEIRA	01/06/93	F	C5	D1
1015	PAULO RODRIGUES	17/08/92	M	C2	D2

Exemplo – Criação de Tabelas c/ Restrições de Integridade

- Banco de dados Companhia
  - Considere as seguintes restrições:
    - Tabela cargo
      - O nome do cargo deve ser único
      - Caso não seja informado o valor do salario, preencher o campo com o valor 998
      - O campo salario não deve permitir valores nulos
    - Tabela depto
      - O nome do departamento deve ser único
      - Não pode haver um departamento sem ramal

Exemplo – Criação de Tabelas c/ Restrições de Integridade

- Tabela Funcionario
  - O nome do funcionario e a data de admissão não pode ser preenchido com valor nulo
  - O sexo deve ser preenchido com o valor F ou M
  - Ao atualizar o código do cargo na tabela cargo, ele também deve ser atualizado na tabela funcionário
  - Ao apagar o código do cargo na tabela cargo, ele deve ser preenchido com valor nulo na tabela funcionário
  - Ao atualizar o código do departamento na tabela depto, ele também deve ser atualizado na tabela funcionário
  - Ao apagar o código do departamento na tabela depto, ele deve ser preenchido com valor nulo na tabela funcionário

Resposta

```
create table cargo(
cdcargo char(3) primary key,
nmcargo varchar(20) unique,
vrsalario numeric(9,2) default 998
);
show create table cargo;

create table departamento(
cddepto char(3) primary key,
nmdepto varchar(20) unique,
ramal numeric(4) not null
);
```

Resposta

```
create table funcionario(
nrmatric int primary key,
nmfunc varchar(40) not null,
dtadm date not null ,
sexo char(1) check (sexo in('M','F')),
cdcargo char(3),
cddepto char(3),
foreign key(cdcargo) references cargo(cdcargo) on
  update cascade on delete set null,
foreign key(cddepto) references cargo(cddepto) on
  update cascade on delete set null);
```

Campos do tipo ENUM - MySQL

- ENUM: Objeto string com as seguintes propriedades:
  - Campo que pode ter um único valor de uma lista de valores que é especificada na criação da tabela
  - Caso não seja fornecido um valor para o campo:
    - Se não tiver sido informado um valor default:
      - Recebe NULL, para campos que permitem nulo;
      - Recebe o primeiro valor da lista, para campos NOT NULL.

Campos do tipo ENUM e SET - MYSQL

- SET: objeto string com as seguintes propriedades:
  - Pode conter nenhum, um ou vários valores de uma lista;
  - Quando é preenchido com um valor incorreto:
    - Não insere os dados
  - Caso não seja fornecido um valor para o campo:
    - Se não tiver sido informado um valor default:
      - Recebe NULL, para campos que permitem nulo;
      - Não insere os dados para campos obrigatórios.

Exemplo

```
CREATE TABLE professor (
codigop int auto_increment,
nome VARCHAR(50) NOT NULL,
sexo enum ('M', 'F'),
titulo set ('especialista','mestre','doutor'),
primary key(codigop)
);
```

Exemplo

```
--inserir um valor no campo do tipo set
insert into professor(nome,sexo,titulo)
values('Paulo Roberto', 'M','doutor');
--inserir dois valores no campo do tipo set
insert into professor(nome,sexo,titulo)
values('Rita de Cassia', 'F','especialista,mestre');
--inserir NULL no atributo sexo, tipo do campo não barra valores nulos
insert into professor(nome, titulo)
values('Maria do Carmo', 'especialista,mestre');
select * from professor;
insert into professor(nome, titulo)
values('Maria Jose', 'graduado');
```

Campos do tipo ENUM e SET - MYSQL

- Caso o MYSQL seja configurado com o modo **STRICT_TRANS_TABLES**, ao inserir os valores incorretos há a ocorrência de um erro.

```
set sql_mode="STRICT_TRANS_TABLES";
CREATE TABLE professor (
codigop int auto_increment,
nome VARCHAR(50) NOT NULL,
sexo enum ('M', 'F'),
titulo set ('especialista','mestre','doutor'),
primary key(codigop)
);
Insert into professor(nome,sexo, titulo) values('Maria','X','especialista');
Error Code: 1265. Data truncated for column 'sexo' at row 1
```

Campos do tipo ENUM e SET - MYSQL

* Quantidade máxima de elementos
  * ENUM: Máximo de 65.535 elementos
  * SET: 64 elementos