

7. SQL // View

conceito | vistas sobre dados detidos por tabelas reais que consistem numa relação virtual (não existe um segundo armazenamento físico dos dados) derivada de relações base e que permite manipular os dados da view

utilização | apresentar esquemas de bases de dados a diferentes aplicações
integridade e privacidade (segurança)
estruturas de queries mais complexa

criação | `CREATE VIEW <viewName> AS <SQLquery>;`

obs | no viewName podemos renomear os atributos, colocando-os entre parêntesis

exemplo `CREATE VIEW ABC(D,F) AS SELECT ATR1, ATR2 FROM`

utilização | as views podem ser utilizadas como fonte de dados das operações SQL já estudadas, podendo ser executadas de duas formas:

query modification → transferência da query definida na original

view materialization → criação de tabela temporária com resultado da view

alterar | `ALTER VIEW <viewName> AS <SQLquery>;`

eliminar | `DROP VIEW <viewName>;`

atualização de dados | embora não recomendada (especialmente por aplicações do cliente), é possível atualizar as tabelas representadas por uma view, com as restrições

- só pode incluir uma tabela
- tem de incluir a primary key e todos os atributos NOT NULL (com default value)

⚠ não pode agrupar atributos ou usar funções de agregação

`Create view ABC
as select A, B, C
from Table1, Table2
where A > 25`

`select A, B
from ABC
where B != 10`

`select A, B
from Table1, Table2
where A > 25 and B != 10`

query modification

atualização de
dados
validação

CREATE VIEW ... FROM ... WITH CHECK OPTION;
com esta instrução garantimos que as condições da cláusula
WHERE são verificadas na atualização dos dados da
view, impedindo a inserção de dados incompatíveis
com a view (mesmo que compatíveis com a relação)

exemplo

... WHERE DNO = 5 WITH CHECK OPTION;
INSERT INTO cinemaNome VALUES (-, - (8))
ERRO! pois DNO = 8 (≠ 5)