Vistas SQL (“views”)

Bases de Dados (CC2005)
Departamento de Ciência de Computadores
Faculdade de Ciências da Universidade do Porto

Eduardo R. B. Marques — DCC/FCUP

Vistas SQL (“views”)

- Algumas consultas sobre múltiplas tabelas, empregando junções, consultas encadeadas, etc, permitem dar muitas vezes uma **“vista” conveniente e “amigável ao utilizador” da informação na BD.**
- SQL suporta o conceito de **vista (“view”)**, que funciona como uma “tabela virtual” armazenada na BD.
- **Uma vista pode ser consultada tal e qual uma tabela** e sob certas condições (dependentes do SGBD) suportar também outras operações como inserções, remoções, ou actualizações. Focamo-nos no entanto apenas no uso de “views” para consultas.

Criação de vistas

DROP VIEW [IF EXISTS] **<NOME DA VISTA>;**
CREATE VIEW [IF NOT EXISTS]
<NOME DA VISTA>(<ATRIBUTO 1>, …, <ATRIBUTO N>)
AS <CONSULTA>;

- Tal como uma tabela, uma vista tem um **nome e atributos** associados.
- Os dados são no entanto "virtuais", isto é, obtidos a partir de uma consulta que está associada à vista.
- Os comandos **DROP VIEW** e **CREATE VIEW** servem para respectivamente apagar e criar vistas.
- Vamos ver alguns exemplos.

Exemplo de vista

```
DROP VIEW IF EXISTS COUNTRY_DATA;

CREATE VIEW COUNTRY_DATA
(Name, Region, Customers)
AS (
  SELECT COUNTRY.Name, REGION.Name, COUNT(*)
  FROM COUNTRY JOIN REGION USING(RegionId)
  JOIN CUSTOMER ON(CUSTOMER.Country=COUNTRY.Name)
  GROUP BY COUNTRY.Name
  ORDER BY COUNTRY.Name
);
```

Nome

Atributos

Consulta

Consulta sobre vistas — exemplo

SELECT * FROM COUNTRY_DATA;

Name	Region	Customers
Afghanistan	Asia	1
Algeria	Africa	3
American Samoa	America	1
Angola	Africa	2
Anguilla	Other countries	1
Argentina	America	13

…

- Criada uma vista, podemos executar consultas sobre a mesma, tal e qual como numa tabela.

Consulta sobre vistas — exemplo (cont.)

```
SELECT Name, Customers FROM COUNTRY_DATA
WHERE Region = 'America'
ORDER BY Customers DESC;
```

```
+--------------------+-----------+
| Name               | Customers |
+--------------------+-----------+
| United States      |        36 |
| Mexico             |        30 |
| Brazil             |        28 |
. . .
| Puerto Rico        |         2 |
| French Guiana      |         1 |
| American Samoa     |         1 |
+--------------------+-----------+
16 rows in set (0.00 sec)
```

Vistas — 2º exemplo

```
CREATE VIEW MOVIE_DATA
(Title, Year, Actors, Genres)
AS
  SELECT M.Title, M.Year,  A.Actors, G.Genres
 FROM MOVIE M
  NATURAL JOIN (
    SELECT MovieId, GROUP_CONCAT(Name) AS Actors
    FROM ACTOR NATURAL JOIN MOVIE_ACTOR
    GROUP BY MovieId ) A
  NATURAL JOIN (
    SELECT MovieId, GROUP_CONCAT(Label) AS Genres
    FROM GENRE NATURAL JOIN MOVIE_GENRE
    GROUP BY MovieId ) G
;
```

Nota: veja documentação de GROUP_CONCAT [**aqui**].

Vistas — 2º exemplo (cont.)

```
SELECT * FROM MOVIE_DATA WHERE Title='Toy Story';
```

Title	Year	Actors	Genres
Toy Story	1995	Don Rickles,Tim Allen,Jim Varney,Tom Hanks	Adventure,Comedy,Animation

```
SELECT Title FROM MOVIE_DATA
WHERE Actors LIKE '%Tom Cruise%'
AND Genres LIKE '%Action%'
ORDER BY Title;
```

Title
Edge of Tomorrow
Jack Reacher
Knight and Day
Minority Report
Mission: Impossible
Mission: Impossible - Ghost Protocol
Mission: Impossible - Rogue Nation
Mission: Impossible II
Mission: Impossible III
Oblivion
The Last Samurai
Top Gun

12 rows in set (0.01 sec)

Vantagem do uso de “views”

- Uma “view” pode abstrair/esconder detalhes do modelo relacional, por exemplo:
  - as tabelas que são consultadas e a relação entre estas;
  - a necessidade de agregar dados (com contagens, somas, etc).
- “Views” não tomam grande espaço na BD, apenas o necessário para armazenar as consultas SQLs associadas.