

SQL: Subqueries

Fernando Lobo

Base de Dados, Universidade do Algarve

Subqueries

- podemos ter uma query dentro de outra query.
- uma expressão do tipo SELECT-FROM-WHERE entre parêntesis é uma subquery.
- o resultado de uma subquery é uma tabela.
- pode ser usada em vários locais, incluindo as cláusulas FROM e WHERE.

Subqueries que retornam um tuplo

Se a subquery retorna apenas um tuplo, então podemos usá-la como se fosse um valor.

- um tuplo único é geralmente garantido através do conceito de chave.
- obtêm-se um erro de execução se não existir nenhum tuplo, ou se existir mais do que um tuplo.
- caso mais habitual: tuplo com um só atributo.

Exemplo

Qual a morada do estúdio que fez o Star Wars de 1977?

- abordagem 1: fazer um join de estúdios com filmes.
- abordagem 2: decompor o problema em dois. Primeiro, encontrar o nome do estúdio que fez o Star Wars de 1977. Depois encontrar a morada desse estúdio.

Abordagem 1

- Qual a morada do estúdio que fez o Star Wars de 1977?

```
SELECT morada
FROM Estudios, Filmes
WHERE Estudios.nome = Filmes.nomeEstudio
      AND Filmes.nome = 'Star Wars'
      AND Filmes.ano = 1977;
```

Abordagem 2

- Qual a morada do estúdio que fez o Star Wars de 1977?

```
SELECT morada
FROM Estudios
WHERE nome = (
    SELECT nomeEstudio
    FROM Filmes
    WHERE nome = 'Star Wars'
    AND ano = 1977
);
```

- reparem na “scoping rule” de nome.
- o primeiro nome é o nome do estúdio, o segundo nome é o nome do filme.
- subquerie (SELECT nomeEstudio ...) produz um só tuplo. Porquê?

Subqueries que retornam mais do que um tuplo

- E se o resultado da subquery tiver mais do que 1 tuplo?
- ... também se pode usar subqueries recorrendo a um destes 4 operadores:
 - ① IN
 - ② EXISTS
 - ③ ALL
 - ④ ANY

Operador IN

- IN é equivalente a \in
- *tuplo* IN *tabela* é verdadeiro se e só se o *tuplo* pertencer à *tabela*.
 - ▶ *tuplo* NOT IN *tabela* significa o oposto.
- IN pode aparecer na cláusula WHERE.
- a *tabela* é normalmente uma subquery.

Exemplo

- Qual a morada dos estúdios que já produziram filmes a preto e branco?

```
SELECT morada
FROM Estudios
WHERE nome IN (
    SELECT nomeEstudio
    FROM Filmes
    WHERE aCores=FALSE
);
```

Outro exemplo

- O operador IN também pode ser aplicado a tuplos.
 - ▶ Quais são os estúdios que já produziram filmes do Harrison Ford?
 - ▶ Primeiro encontramos os filmes do Harrison Ford. Depois temos de encontrar os estúdios que produziram esses filmes.

```
SELECT nomeEstudio
FROM Filmes
WHERE (nome,ano) IN
    (
        SELECT nomeFilme, anoFilme
        FROM Participa
        WHERE nomeActor = 'Harrison Ford'
    );
```

Operador EXISTS

- EXISTS(*tabela*) é verdadeiro se e só se a *tabela* não é vazia.
 - ▶ NOT EXISTS(*tabela*) significa o oposto.
- EXISTS pode aparecer na cláusula WHERE.

Exemplo

- Quais os estúdios que produziram mais do que 1 filme?

```
SELECT DISTINCT nomeEstudio
FROM Filmes AS f
WHERE EXISTS
    (
        SELECT *
        FROM Filmes
        WHERE nomeEstudio = f.nomeEstudio
        AND NOT (ano = f.ano AND nome = f.nome)
    );
```

- a query acima é uma “correlated subquery”.
- subquery dá o conjunto de filmes que têm o mesmo estúdio que f, mas excluindo o próprio f.
- reparem novamente na “scoping rule” dos atributos.

Operador IN

- ALL e ANY são usados juntamente com os operadores relacionais =, >, >=, ...
- $x > \text{ALL}(\textit{tabela})$ é verdadeiro se e só se x for maior do que todos os tuplos da *tabela*.
- $x < \text{ANY}(\textit{tabela})$ é verdadeiro se e só se x for menor do que pelo menos um tuplo da *tabela*.
 - ▶ em vez de '>' e '<' podemos utilizar qualquer operador relacional.
 - ▶ os tuplos da *tabela* têm de ter apenas um atributo.

Exemplo

- Qual o filme com a maior duração?

```
SELECT nome, ano
FROM Filmes
WHERE duracao >= ALL (
    SELECT duracao
    FROM Filmes
    WHERE duracao IS NOT NULL
);
```

- Há que ter cuidado com os NULLs. Porquê?

Outro exemplo

- Quais os filmes que não têm a maior duração?

```
SELECT nome, ano
FROM Filmes
WHERE NOT duracao >= ALL (
    SELECT duracao
    FROM Filmes
    WHERE duracao IS NOT NULL
);
```

- Ou então:

```
SELECT nome, ano
FROM Filmes
WHERE duracao < ANY (
    SELECT duracao
    FROM Filmes
    WHERE duracao IS NOT NULL
);
```