SUBCONSULTAS E TIPOS DE JUNÇÃO

Banco de Dados

Profa. Ana Paula Wauke

Operador IN e NOT IN

- IN: Dados de um conjunto, membros de um conjunto ou uma consulta;
- Filmes de categoria: Drama, Terror, Suspense ou Comedia:
  - SELECT T.nome_titulo AS Titulo, C.nome_categoria AS Categoria
  - FROM tbTitulo T
  - INNER JOIN tbCategoria C ON C.codigo_categoria = T.codigo_categoria
  - WHERE C.nome_categoria IN (‘Drama’, ‘Terror’, ‘Suspense’, ’Comedia’);
- Filmes produzidos pela ABC Distribuidora:
  - SELECT T.nome_titulo
  - FROM tbTitulo T
  - WHERE T.codigo_titulo IN (SELECT F.codigo_titulo

    FROM tbFilme F

    WHERE f.nome_distribuidor = ‘ABC Distribuidora’

    );

- NOT IN – ausência de membros de um conjunto;
- Filmes lançados entre 1995 e 2009 e que nunca foram emprestados:
  - SELECT T.nome_titulo;
  - FROM tbTitulo T
  - WHERE T.codigo_titulo NOT IN

    (SELECT F.codigo_titulo

    FROM tbFilme F

    INNER JOIN tbEmprestimoDevolucao ED ON F.codigo_filme = ED.codigo_filme)
  - AND T.ano BETWEEN 1995 AND 2009;

- E o operador "=" ?
  - Apenas para valores específicos e não um grupo de valores.

Operador EXISTS

- Verifica se o resultado da consulta possui algum registro (true caso algum registro satisfaça condição)
- EX.: listar clientes e data de cadastro que possuem algum empréstimo
  - SELECT C.nome_cli, C.data_cadastro
  - FROM tbCliente C
  - WHERE EXISTS (SELECT ED.codigo_cli
    FROM tbEmprestimoDevolucao ED
    WHERE C.codigo_cli = ED.codigo_cli
    );

- A mesma consulta com uso do IN
  - SELECT C.nome_cli, C.data_cadastro
  - FROM tbCliente C
  - WHERE C.codigo_cli IN (SELECT ED.codigo_cli
    FROM tbEmprestimoDevolucao ED
    );

EXISTS, IN → INNER JOIN

- A mesma consulta com uso do INNER JOIN:
  - SELECT DISTINCT C.nome_cli, C.data_cadastro
  - FROM tbCliente C
  - INNER JOIN tbEmprestimoDevolucao ED ON

  C.codigo_cli = ED.codigo_cli;

- DISTINCT – Porque o retorno é da junção de todos empréstimos do cliente.
  O que leva a uma consulta mais custosa em termos de desempenho.

NOT EXISTS

- Ao Contrário Do EXISTS, o NOT EXISTS verifica se resultado é falso
- Ex.: encontrar todos clientes que não possuem empréstimo
  - SELECT C.nome_cli, C.data_cadastro
  - FROM tbCliente C
  - WHERE NOT EXISTS (SELECT ED.codigo_cli

    FROM TBEMPRESTIMODEVOLUCAO ED

    WHERE C.CODIGO_CLI = ED.CODIGO_CLI

    );

NOT EXISTS

- UPDATE e DELETE:
- Ex2.: atualizar para que a data de devolução prevista para as locações feitas em novembro de 2014 receba dois dias a mais
  - UPDATE tbEmprestimoDevolucao ED
  - SET ED.data_devolução_prevista BETWEEN ‘2014-11-01’ AND ‘2014-11-30’
  - FROM tbCliente C
  - WHERE NOT EXISTS (SELECT ED.codigo_cli
    FROM TBEMPRESTIMODEVOLUCAO ED
    WHERE C.CODIGO_CLI = ED.CODIGO_CLI
    );
- Ex3.: exclua todos os clientes que não emprestaram nenhum filme em 2008, 2009 e 2010
  - DELETE FROM tbCliente
  - WHERE codigo_cli NOT IN
    (SELECT ED.codigo_cli
    FROM tbEmprestimoDevolucao ED
    WHERE ED.data_empréstimo BETWEEN ‘2008-01-01’ AND ‘2010-12-31);’

Operadores ALL e SOME

- Operadores relacionais comparam com um conjunto de valores

Operador + ALL	Significado
=ALL	Igual a todos
<>ALL	Diferente de todos
>ALL	Maior que todos
>=ALL	Maior ou igual a todos
<ALL	Menor que todos
<= ALL	Menor ou igual a todos

Operador + SOME	Significado
=SOME	Igual a pelo menos um
<>SOME	Diferente de pelo menos um
>SOME	Maior que pelo menos um
>=SOME	Maior ou igual a pelo menos um
<SOME	Menor que pelo menos um
<= SOME	Menor ou igual a pelo menos um

EXEMPLO ALL

- Cliente que locou o maior número de filmes em 2009
  - SELECTC.codigo_cli,C.nome_cli,COUNT(ED.data_emprestimo) AS 'Quantidade de filmes locados'
  - FROM tbCliente C
  - INNER JOIN tbEmprestimoDevolucao ED ON C.codigo_cli = ED.codigo_cliWHEREED.data_empréstimo BETWEEN '20090101' AND '20091231'
  - GROUP BY C.codigo_cli
  - HAVING COUNT (ED.data_emprestimo) >= ALL (SELECT COUNT(ED.date_emprestimo)

    FROM tbEmprestimoDevolucao ED

    WHERE ED.data_emprestimo BETWEEN '20090101' AND '20091231' GROUP BY ED.codigo_cli);

EXEMPLO ALL

- Cliente que locou o maior número de filmes em 2009
  - SELECTC.codigo_cli,C.nome_cli,COUNT(ED.data_emprestimo) AS 'Quantidade de filmes locados'
  - FROM tbCliente C
  - INNER JOIN tbEmprestimoDevolucao ED ON C.codigo_cli = ED.codigo_cliWHEREED.data_empréstimo BETWEEN '20090101' AND '20091231'
  - GROUP BY C.codigo_cli
  - HAVING COUNT (ED.data_emprestimo) >= ALL (SELECT COUNT(ED.date_emprestimo)

    FROM tbEmprestimoDevolucao ED

    WHERE ED.data_emprestimo BETWEEN '20090101' AND '20091231' GROUP BY ED.codigo_cli);

EXEMPLO ALL – substituindo por LIMIT

- Cliente que locou o maior número de filmes em 2009
  - SELECTC codigo_cli, C.nome_cli, COUNT(*) AS 'Quantidade de filmes locados'
  - FROM tbCliente C
  - INNER JOIN tbEmprestimoDevolucao ED ON C.codigo_cli = ED.codigo_cli
  - WHERE ED.data_empréstimo BETWEEN '20090101' AND '20091231'
  - GROUP BY C.codigo_cli
  - ORDER BY COUNT(*) DESC
  - LIMIT 1;

EXEMPLO SOME

- Clientes que tiveram pelos menos 1 empréstimo em 2009
  - SELECT C.codigo_cli, C.nome_cli, COUNT(*) AS 'Quantidades de filmes locados'
  - FROM tbCliente C
  - INNER JOIN tbEmprestimoDevolucao ED ON C.codigo_cli = ED.codigo_cli
  - WHERE ED.data_emprestimo BETWEEN '20090101' AND '20091231'
  - GROUP BY C.codigo_cli
  - HAVING COUNT(*) = SOME (SELECT COUNT(*)

    FROM tbEmprestimoDevolucao ED

    WHERE ED.data_emprestimo BETWEEN '20090101' AND '20091231'

    GROUP BY ED.codigo_cli);

TIPOS DE JUNÇÃO

- INNER JOIN – retorna apenas as que possuem conexão em junção
- OUTER JOIN – retorna
  - LEFT OUTER JOIN – prioriza tabela da esquerda
  - RIGHT OUTER JOIN – prioriza tabela da direita
  - FULL OUTER JOIN – ambos os lados

Codigo_cli	Nome_cli	CPF_cli	Cidade	UF
1	Anna	12345678911	Curitiba	PR
2	Pedro	12345678912	Curitiba	PR
3	Janaína	12345678913	Curitiba	PR
4	Elaine	12345678914	Curitiba	PR
5	Gustavo	12345678915	Curitiba	PR

Codigo_cli	Codigo_filme	Data_empréstimo	Data_devolução_prevista	Data_devolucao_efetiva	Multa
1	1	2006-03-15	2006-03-17	2006-03-17	0,00
2	2	1996-03-23	1996-03-27	1996-03-28	3,00
3	1	2014-11-20	2014-11-25		
3	5	2009-05-08	2009-05-11	2009-05-11	0,00

LEFT OUTER JOIN

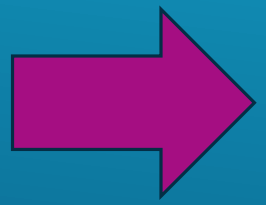
- SELECT C.nome_cli, ED.data_emprestimo
- FROM tbCliente C
- LEFT OUTER JOIN tbEmprestimoDevolucao ED ON C.codigo_cli = ED.codigo_cli;

Nome_cli	Data_empréstimo
Anna	2006-03-15
Pedro	1996-03-23
Janaína	2014-11-20
Janaína	2009-05-08
Elaine	NULL
Gustavo	NULL

LEFT OUTER JOIN

- SELECT C.nome_cli
- FROM tbCliente C LEFT OUTER JOIN tbEmprestimoDevolucao ED ON C.codigo_cli = ED.codigo_cli
- WHERE ED.data_empréstimo IS NUL;

Nome_cli	Data_empréstimo
Anna	2006-03-15
Pedro	1996-03-23
Janaína	2014-11-20
Janaína	2009-05-08
Elaine	NULL
Gustavo	NULL



Nome_cli
Elaine
Gustavo

Codigo_cli	Nome_cli	CPF_cli	Cidade	UF
1	Anna	12345678911	Curitiba	PR
2	Pedro	12345678912	Curitiba	PR
3	Janaína	12345678913	Curitiba	PR
4	Elaine	12345678914	Curitiba	PR
5	Gustavo	12345678915	Curitiba	PR

Codigo_cli	Codigo_filme	Data_empréstimo	Data_devolução_prevista	Data_devolucao_efetiva	Multa
1	1	2006-03-15	2006-03-17	2006-03-17	0,00
2	2	1996-03-23	1996-03-27	1996-03-28	3,00
3	1	2014-11-20	2014-11-25		
3	5	2009-05-08	2009-05-11	2009-05-11	0,00

RIGHT OUTER JOIN

- SELECT C.nome_cli, ED.data_emprestimo
- FROM tbEmprestimoDevolucao ED
- RIGHT OUTER JOIN tbCliente C ON C.codigo_cli = ED.codigo_cli

Nome_cli	Data_empréstimo
Anna	2006-03-15
Pedro	1996-03-23
Janaína	2014-11-20
Janaína	2009-05-08
Elaine	NULL
Gustavo	NULL

FULL OUTER JOIN

- SELECT C.nome_cli, ED.data_emprestimo
- FROM tbEmprestimoDevolucao ED
- FULL OUTER JOIN tbCliente C ON C.codigo_cli = ED.codigo_cli

ERROR 1064 (42000): You have an error in your SQL syntax; check the manual that corresponds to your MySQL server version for the right syntax to use near 'full outer join Orders O on P.P_id=O.P_Id
where P.P_id IS NULL and O.P_Id IS NUL' at line 1

UNION (Alternativa)

- (SELECT C.nome_categoria, T.nome_titulo FROM
- tbCategoria C LEFT OUTER JOIN tbTitulo T ON
- C.codigo_categoria = T.codigo_categoria WHERE
- T.nome_titulo IS NULL)
- UNION
- (SELECT C.nome_categoria, T.nome_titulo FROM
- tbCategoria C RIGHT OUTER JOIN tbTitulo T ON
- C.codigo_categoria = T.codigo_categoria WHERE
- C.nome_categoria IS NULL)