BASE DE DADOS

SQL –DML

Teóricas
Ano Lectivo 2018/2019

SQL - SUBQUERIES

DEPARTAMENTO DE ENGENHARIA INFORMÁTICA

- *Uma* Subquery é uma instrução SELECT que está embebida numa cláusula de outra instrução SELECT.
- Elas podem ser úteis quando precisamos de selecionar linhas de uma tabela com uma condição que depende dos dados na própria tabela.
- Podemos colocar a subconsulta na cláusula SELECT, FROM, WHERE e HAVING.

```
SELECT      ListaDeColunas
FROM        TabelaExterna
WHERE       Expressao Operador

                    (SELECT     ListaDeColunas
                    FROM        TabelaInterna
                    [WHERE      Expressão Operador]);
```

SQL - SUBQUERIES

DEPARTAMENTO DE ENGENHARIA INFORMÁTICA

- A *subquery* geralmente será executada primeiro, e a sua saída é usada para completar a condição de consulta para a consulta principal (ou externa).

- As *subqueries* podem retornar **uma só linha** ou **várias linhas**

  - As que retornam um só linha só podem mencionar operadores aritméticos.

  - As que retornam várias linhas podem mencionar os operadores:
    IN (NOT IN) ; ANY(SOME); ALL ; EXISTS (NOT EXISTS)

- As *subqueries* **podem ser de dois tipos:**
  - **Não correlacionadas**
  - **Correlacionadas**

SQL - SUBQUERIES

DEPARTAMENTO DE ENGENHARIA INFORMÁTICA

Não Correlacionadas

- SELECT Interior
  - Não depende de valores do SELECT exterior.
  - Não referência colunas do SELECT exterior.
- SELECT Interior é executado em 1º lugar (uma vez) porque o SELECT exterior é que depende do SELECT interior (executado depois).

```
SELECT Nome
FROM  Empregado
WHERE salario  = ( SELECT MIN (salario)
                   FROM Empregado )
```

Operador aritmético porque subquery retorna uma lnha (registo único)

SQL - SUBQUERIES

DEPARTAMENTO DE ENGENHARIA INFORMÁTICA

Correlacionadas

- SELECT Interior depende de valores do SELECT exterior.
- SELECT Interior é executado tantas vezes quanto o SELECT exterior e espera por um valor do SELECT exterior.

```
SELECT Nome, Salario
FROM Empregado E
WHERE Salario < ( SELECT SUM(Valor)
                  FROM Comissao C WHERE C.Id = E.Id )
```

valor variável;
depende da linha do
select exterior

faz a ligação ao
select exterior

SQL - SUBQUERIES

DEPARTAMENTO DE ENGENHARIA INFORMÁTICA

Resumo

Tipo de SubQuery	Select Interior	Sentido de Execução	Execução do select interior
Não correlacionada	Não depende do select exterior	1º o select Interior 2º select Exterior	Uma vez
Correlacionada	Depende do select exterior	Execução do interior intercalada com o exterior	nr. vezes = ao numero de vezes do select exterior

SQL - SUBQUERIES

DEPARTAMENTO DE ENGENHARIA INFORMÁTICA

Operador IN/NOT IN

- Seleciona as linhas em que os campos indicados antes do operador existam na *subquery*.
- Os campos indicados têm de ser no mesmo número dos campos retornados pela query e têm de ter domínios compatíveis.
- O operador NOT IN permite obter o resultado inverso.

```
SELECT nome FROM empregado
WHERE nr_emp NOT IN   (SELECT diretor FROM departamento);
```

- Aparecem na forma:

<attribute-name> IN (subquery) Ou

<attribute-name> NOT IN (subquery)

SQL - SUBQUERIES

DEPARTAMENTO DE ENGENHARIA INFORMÁTICA

Operador ANY(SOME)

- Seleciona os resultados cujos campos indicados sejam iguais (=), maiores (>), menores(<) ou diferentes (<>) do **que pelo menos uma linh**a da subquery.
- Os campos indicados têm de ser no mesmo número dos campos retornados pela subquery e têm de ter domínios compatíveis.
- **=ANY é o mesmo que IN**

```
SELECT nome, salario, nrdep FROM empregado

WHERE salario < ANY (SELECT distinct salario FROM
                     empregado where nrDep=2)
```

- A Condição ANY retorna TRUE quando qualquer dos valores do resultado da subquery satisfaz a condição.
- A palavra-chave **Distinct** é frequentemente utilizada para evitar que os registos sejam selecionados mais do que uma vez

SQL - SUBQUERIES

DEPARTAMENTO DE ENGENHARIA INFORMÁTICA

Operador ALL

- Seleciona os resultados cujos campos indicados sejam iguais (=), maiores (>), menores(<) ou diferentes(<>) **do que todos os** tuplos da subquery.
- Os campos indicados têm de ser no mesmo número dos campos retornados pela query e tem de ter domínios compatíveis.
- **<>ALL é o mesmo que NOT IN**

```
SELECT nome, salario, nrdep FROM empregado

WHERE salario > ALL (SELECT Distinct salario FROM
                     empregado where nrDep=2)
```

Operador EXISTS/NOT EXISTS

- Em subqueries correlacionadas todos os operadores lógicos são aplicados, contudo usa-se o operador **EXISTS ou NOT EXISTS** para testar se um valor recuperado pela consulta externa existe no conjunto de valores recuperados pela consulta interna.

- O operador EXISTS ou NOT EXISTS **é usado para determinar se há dados numa lista de valores** (restringe o conjunto de resultados de uma consulta externa para as linhas que atendam a subquery).
  - Verifica se resultado de subquery **é ou não um conjunto vazio.**

- O operador EXISTS e NOT EXISTS retorna TRUE ou FALSE, dependendo se as queries devolvem linhas ou não;

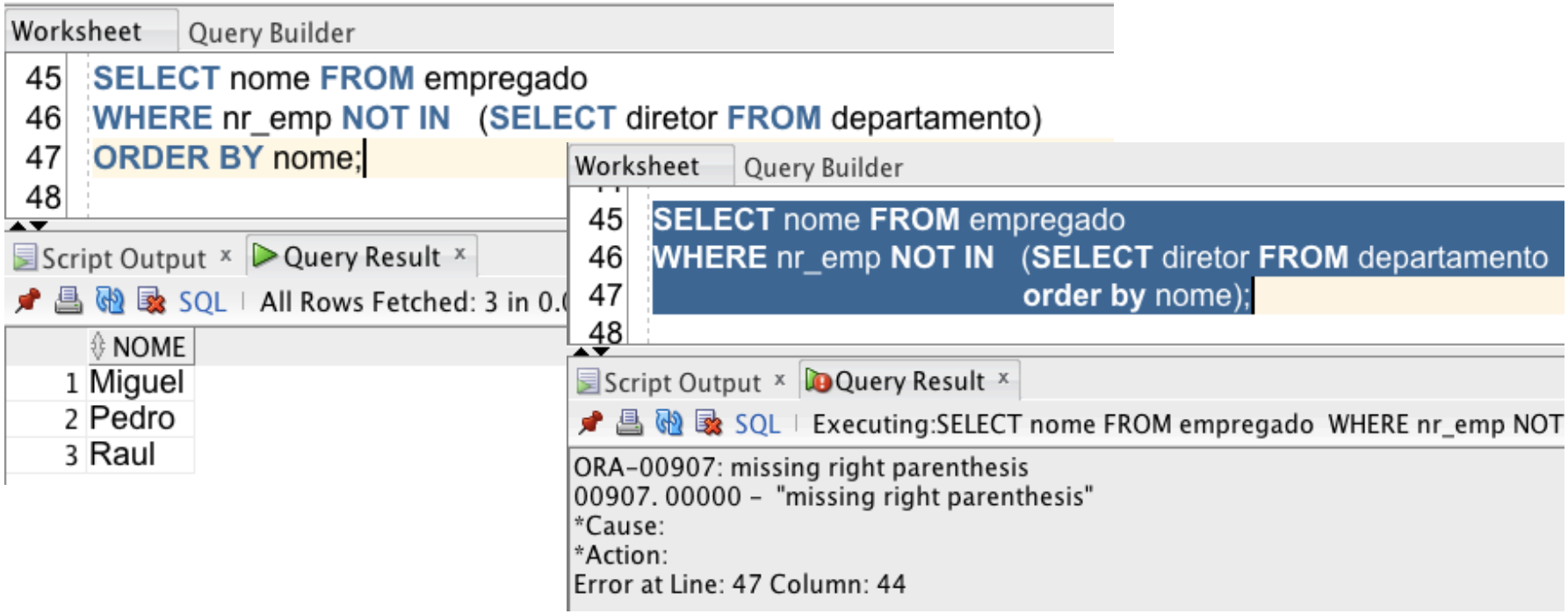
SQL - SUBQUERIES

DEPARTAMENTO DE ENGENHARIA INFORMÁTICA

Ordenação de Dados

- NÃO se pode utilizar uma cláusula ODER BY numa subconsulta.

  Mantém-se a regra de que só se pode ter uma única cláusula ORDER BY para uma instrução de SELECT e, se especificada, terá de ser na última cláusula no comando SELECT.



SQL - SUBQUERIES

DEPARTAMENTO DE ENGENHARIA INFORMÁTICA

Subconsulta como uma tabela derivada

- É um conjunto de registos dentro de uma consulta que funciona como uma tabela;
- Ela toma o lugar da tabela na cláusula FROM.
- É otimizada com o resto da consulta.

Worksheet | Query Builder

```
31  SELECT empregado.*, TH.hora
32  FROM  empregado, ( SELECT   nr_emp, sum(TotalHoras) hora
33                     FROM trabalho
34                     GROUP BY nr_emp)  TH
35  WHERE  empregado.nr_emp = TH.nr_emp;
36
```

Script Output | Script Output 1 | Query Result

SQL | All Rows Fetched: 3 in 0.261 seconds

	NR_EMP	NOME	MORADA	SALARIO	NRDEP	HORA
1	1	Antonio	R. Tome	10000	1	21
2	2	Jose	Av. aliados	800	1	8
3	3	Ana	R. Boavista	1000	2	15

```
SELECT empregado.*, TH.hora
FROM  empregado, ( SELECT   nr_emp, sum(TotalHoras) hora
                   FROM trabalho
                   GROUP BY nr_emp)  TH
WHERE  empregado.nr_emp = TH.nr_emp;
```

ipt Output | Query Result

SQL | All Rows Fetched: 3 in 0.02 seconds

NR_EMP	HORA
1	21
2	8
3	15

SQL - SUBQUERIES

DEPARTAMENTO DE ENGENHARIA INFORMÁTICA

Subconsulta como uma expressão

Utilizando uma subconsulta como uma expressão

- É executada uma vez para toda a instrução;

```
SELECT nr_emp, nome, (SELECT MIN(totalhoras)  FROM trabalho T WHERE T.nr_emp = E.nr_emp) minimohora
FROM empregado E;
```

cript Output x | Query Result x

SQL | All Rows Fetched: 6 in 0.033 seconds

	NR_EMP	NOME	MINIMOHORA
1	1	Antonio	4
2	2	Jose	8
3	3	Ana	7
4	4	Raul	(null)
5	5	Miguel	(null)
6	6	Pedro	(null)

SQL - SUBQUERIES

DEPARTAMENTO DE ENGENHARIA INFORMÁTICA

Subconsulta na cláusula Where e Having

- Numa cláusula WHERE/ HAVING temos sempre uma condição e a subquery atua operando dentro dessa condição

```
1  SELECT nrdep , COUNT(*)  TotalEmpregados
2  FROM empregado
3  GROUP BY nrdep
4  HAVING COUNT(*) = (SELECT MAX(COUNT(*)) FROM empregado  GROUP BY nrdep);
5
```

Script Output

Task completed in 0.33 seconds

```
   NRDEP TOTALEMPREGADOS
--------- ---------------
        2               3
```

Consultas – todos

Pergunta: Quais são os alunos matriculados em **todas** as disciplinas?

- **por Inclusão de Conjuntos - usa-se EXISTS / IN e MINUS**

```
SELECT I1.ID_ALUNO
FROM INSCRITOS I1
WHERE NOT EXISTS ( SELECT ID_DISCIPLINA --------- todas as disciplinas
                   FROM DISCIPLINAS
                   MINUS
                   SELECT ID_DISCIPLINA --------- todas as disciplinas de I1
                   FROM INSCRITOS I2
                   WHERE I2.ID_ALUNO = I1.ID_ALUNO);
```

Consultas – todos

Pergunta: Quais são os alunos matriculados em **todas** as disciplinas?

- **Por comparação de Cardinalidades – usa-se GROUP BY e COUNT()**

```
SELECT ID_ALUNO
FROM INSCRITOS
GROUP BY ID_ALUNO
HAVING COUNT(*)=( SELECT COUNT(*)
                  FROM DISCIPLINAS)
```

-nr. disciplinas aluno = total disciplinas

SQL - SUBQUERIES

DEPARTAMENTO DE ENGENHARIA INFORMÁTICA

Consultas – todos

Pergunta: Quais são os alunos matriculados em **todas** as disciplinas?

- **por Quantificação - usa-se EXISTS e/ou IN (quantificadores de existência)**
  - Reformulando a pergunta:
    - "Quais são os alunos, para os quais **não existe** nenhuma disciplina **sem** a sua inscrição?"

```
SELECT DISTINCT ID_ALUNO
FROM INSCRITOS I1
WHERE NOT EXISTS (
        SELECT *
        FROM DISCIPLINAS D WHERE NOT EXISTS (
                SELECT *
                FROM INSCRITOS I2
                WHERE I2.ID_ALUNO = I1.ID_ALUNO
                AND I2.ID_DISCIPLINA = D.ID_DISCIPLINA));
```

SQL - SUBQUERIES

DEPARTAMENTO DE ENGENHARIA INFORMÁTICA

Linhas de Orientação

- A consulta interna deverá ser colocada entre parênteses e, terá de estar no lado direito da condição.
- A subconsulta não pode conter uma cláusula ORDER BY.
- A cláusula ORDER BY aparece no fim da instrução de SELECT principal.
- Coluna múltiplas na lista de SELECT da consulta interna terão de estar pela mesma ordem que as colunas que apareçam na cláusula de condição da consulta principal. O tipo de dados e o número de coluna tem também de corresponder.
- As subconsultas são sempre executadas primeiro a partir da mais interior para menos interior no encadeamento, a não ser que sejam subconsultas correlacionadas.
- Podem-se utilizar operadores lógicos e operadores de SQL, bem como ANY e ALL.

SQL - SUBQUERIES

DEPARTAMENTO DE ENGENHARIA INFORMÁTICA

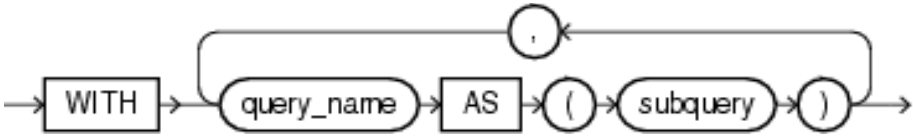
Linhas de Orientação

- As subconsultas podem :
  - produzir uma ou mais linhas.
  - produzir uma ou mais colunas.
  - Utilizar uma ou mais colunas.
  - utilizar GRPOUP BY ou funções de grupo.
  - Ser utilizadas em vários predicados AND e OR da mesma consulta
  - Juntar tabelas.
  - extrair dados de uma tabela diferente da tabela da consulta externa.
  - aparecer em instruções de SELECT, UPDATE; DELETE; INSERT; CREATE TABLE
  - Serem correlacionadas com uma consulta externa.
  - Utilizar operadores SET

SQL - SUBQUERIES

DEPARTAMENTO DE ENGENHARIA INFORMÁTICA

Cláusula With



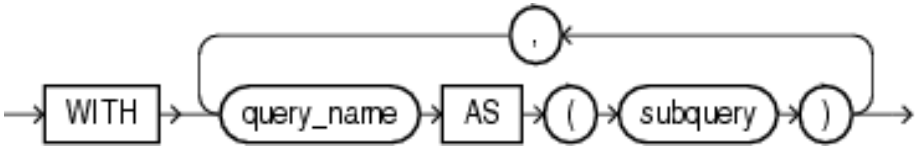
- A cláusula WITH formalmente é chamada "*factoring subquery*".
- A cláusula SQL WITH é usada quando uma subconsulta é executada várias vezes.
- A cláusula WITH query_name permite atribuir um nome a um bloco de subconsulta.
- Cada subconsulta na cláusula WITH especifica um nome de tabela, uma lista opcional de nomes de colunas e uma expressão de consulta que avalia uma tabela (instrução SELECT).
- Podemos referenciar o bloco de subconsultas em vários locais na consulta, especificando o nome da consulta.
- O Oracle otimiza a consulta tratando o nome da consulta como uma exibição in-line ou como uma tabela temporária.
- Cada subconsulta define uma tabela temporária, que pode ser mencionada na cláusula FROM e só é usada somente durante a execução da consulta à qual pertencem.

```
with xyz as
  (
  select owner, count(*) c
 from t
  group by owner
 )
 select * from xyz;
```

SQL - SUBQUERIES

DEPARTAMENTO DE ENGENHARIA INFORMÁTICA

Cláusula With



Exemplo:

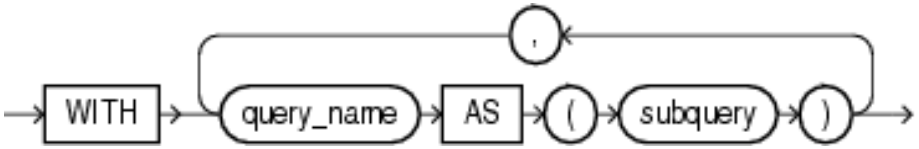
Usando a cláusula WITH, escreva uma query para mostrar o nome do departamento e o total de salários para os departamentos cujo salário total é superior à media dos salários dos departamentos.

```
With
depart_custo  as
    (SELECT d.nome,  SUM(e.salario) dept_total
      FROM departamento d, empregado e
      WHERE  d.nr_dep= e.nrdep
      GROUP BY d.nome),
AVG_custo as (SELECT sum(dept_total)/count(*)  media
                    FROM depart_custo)
SELECT * FROM depart_custo
WHERE dept_total > (SELECT media FROM avg_custo)
 ORDER BY nome;
```

SQL - SUBQUERIES

DEPARTAMENTO DE ENGENHARIA INFORMÁTICA

Cláusula With



Restrições:

- Podemos especificar apenas uma subquery_factoring_clause numa única instrução SQL.
- Não podemos especificar um query_name na sua própria subconsulta. No entanto, qualquer query_name definido na subquery_factoring_clause pode ser usado em qualquer bloco de consulta nomeado subsequente na subquery_factoring_clause.