



Banco de Dados

Sub-Query

- Conceito
- Sintaxe
- Usando Subquery para resolver problemas;
- Hands-on
- Exemplos
- Subquery com funções de grupo
 - Single Row Subquery
 - Multiple Row Subquery
 - Aninhamento de Subquery
 - Operador ANY
 - Operador ALL
 - Exercícios

CONCEITO

- Um comando select dentro de um outro comando select que retorna **uma ou mais linhas** a fim de satisfazer uma clausula WHERE;

QUERY PRINCIPAL

SUBQUERY

- A subquery (inner query) geralmente é executada antes da consulta principal;
- O resultado da subquery é, então, avaliado pelo da query principal (outer query).
- Uma sub-pesquisa (subquery) é uma declaração SELECT que é **aninhada** com outra declaração SELECT e a qual retorna resultados intermediários.

- A subquery é executada primeiro.
- O resultado da subquery é utilizado na query principal

```
SELECT    select_list  
FROM      table  
WHERE     expr operator
```

```
(SELECT    select_list  
FROM      table);
```

- Quais funcionários possuem um salário maior que Abel's?

Main query:



Which employees have salaries greater than Abel's salary?

Subquery:



What is Abel's salary?



```
SELECT last_name, salary
FROM employees
WHERE salary >
      (SELECT salary
       FROM employees
       WHERE last_name = 'Abel');
```

```
SELECT last_name, salary
FROM employees
WHERE salary >
```

11000

```
(SELECT salary
 FROM employees
 WHERE last_name = 'Abel')
```

Single Row Subquery
- Retorno de
um único
valor na
subquerie

- Mostre o sobrenome (last_name) e o cargo (job_id) dos funcionários que possuem o mesmo cargo do funcionário de número 141.

1

```
SELECT last_name, job_id
```

```
FROM employees
```

```
WHERE job_id =
```

2

```
(SELECT job_id
```

```
FROM employees
```

```
WHERE employee_id = 141);
```

1) Para encontrar todos os empregados que tem o mesmo cargo do David Lee fazemos:

```
SELECT FIRST_NAME, LAST_NAME, JOB_ID  
FROM EMPLOYEES
```

```
WHERE JOB_ID = (SELECT JOB_ID  
                FROM EMPLOYEES  
                WHERE FIRST_NAME = 'David' and  
                      LAST_NAME = 'Lee')
```

```
ORDER BY FIRST_NAME
```

A pesquisa interna retorna o cargo de David Lee (SA_REP) o qual é usado na condição WHERE da pesquisa principal.

2) Para encontrar todos os empregados do mesmo departamento do David Lee fazemos:

```
SELECT FIRST_NAME, LAST_NAME, JOB_ID, DEPARTMENT_ID  
FROM EMPLOYEES
```

```
WHERE DEPARTMENT_ID = (SELECT DEPARTMENT_ID  
                       FROM EMPLOYEES  
                       WHERE FIRST_NAME = 'David' and  
                             LAST_NAME = 'Lee')
```

```
ORDER BY FIRST_NAME
```


A seguinte pesquisa atende para encontrar os empregados que ganham acima da média salarial.

```
SELECT FIRST_NAME, LAST_NAME, JOB_ID,  
       DEPARTMENT_ID, SALARY  
FROM EMPLOYEES  
WHERE SALARY > (SELECT AVG(SALARY)  
                FROM EMPLOYEES)  
ORDER BY FIRST_NAME
```

Single Row Subquery
- Retorno de
um único
valor na
subquerie

A seguinte pesquisa atende para encontrar os empregados que trabalham nos departamentos de Marketing e Administração.

```
SELECT FIRST_NAME, LAST_NAME, JOB_ID,  
       DEPARTMENT_ID, SALARY  
FROM EMPLOYEES  
WHERE DEPARTMENT_ID IN (SELECT DEPARTMENT_ID  
                        FROM DEPARTMENTS  
                        WHERE DEPARTMENT_NAME IN  
                          ('Marketing', 'Administration'))  
  
ORDER BY FIRST_NAME
```

Multiple Row
Subquery -
Retorno de
vários
valores na
subquerie

Sub-pesquisas Aninhadas.

Mostrar o primeiro e último nome, cargo, código do departamento e salário para os empregados cujo o salário é maior que o maior salário no departamento 'SALES'.

```
SELECT FIRST_NAME, LAST_NAME, JOB_ID,  
       DEPARTMENT_ID, SALARY  
FROM EMPLOYEES  
WHERE SALARY > ( SELECT MAX(SALARY)  
                FROM EMPLOYEES  
                WHERE DEPARTMENT_ID = (SELECT DEPARTMENT_ID  
                                       FROM DEPARTMENTS  
                                       WHERE DEPARTMENT_NAME = 'Sales'))  
ORDER BY FIRST_NAME
```

FIRST_NAME	LAST_NAME	JOB_ID	DEPARTMENT_ID	SALARY
Lex	De Haan	AD_VP	90	17000
Neena	Kochhar	AD_VP	90	17000
Steven	King	AD_PRES	90	24000

3 linhas retornadas em 0,00 segundos

[Exportação para CSV](#)

Operadores de comparação para múltiplas linhas

Operator	Meaning
IN	Igual a qualquer membro da lista
ANY	Compara o valor com cada valor retornado pela subquerie.
ALL	Compara o valor com todos os valores retornados pela subqueire.
EXISTS	Testa se um valor existe.

O operador EXISTS é frequentemente usado com Sub-pesquisas correlatas. Ele testa quando um valor existe. (NOT EXISTS garante que não existe). Se o valor existir será retornado Verdadeiro, se não existir será retornado Falso.

Para encontrar os empregados que tem no mínimo uma pessoa subordinada a ele, faça:

```
SELECT EMPLOYEE_ID, MANAGER_ID, FIRST_NAME, LAST_NAME,  
       JOB_ID, DEPARTMENT_ID, SALARY  
FROM EMPLOYEES E  
WHERE EXISTS (SELECT EMPLOYEE_ID  
               FROM EMPLOYEES  
               WHERE EMPLOYEES.MANAGER_ID =  
                     E. EMPLOYEE_ID)  
ORDER BY EMPLOYEE_ID;
```

Encontrar os empregados que não tem departamento relacionado na tabela DEPARTAMENTOS:

```
SELECT EMPLOYEE_ID, MANAGER_ID, FIRST_NAME, LAST_NAME,  
       JOB_ID, DEPARTMENT_ID, SALARY  
FROM EMPLOYEES  
WHERE NOT EXISTS (SELECT DEPARTMENT_ID  
                     FROM DEPARTMENTS  
                     WHERE DEPARTMENTS.DEPARTMENT_ID =  
                           EMPLOYEES.DEPARTMENT_ID);
```

EMPLOYEE_ID	MANAGER_ID	FIRST_NAME	LAST_NAME	JOB_ID	DEPARTMENT_ID	SALARY
178	149	Kimberely	Grant	SA_REP	-	7000

1 linhas retornadas em 0,01 segundos

[Exportação para CSV](#)

Outro caminho para encontrar o departamento que não tem nenhum empregado :

```
SELECT DEPARTMENT_ID, DEPARTMENT_NAME
FROM DEPARTMENTS D
WHERE NOT EXISTS (SELECT 'X'
                   FROM EMPLOYEES E
                   WHERE E.DEPARTMENT_ID =
                        D.DEPARTMENT_ID)

ORDER BY 2
```

DEPARTMENT_ID	DEPARTMENT_NAME
160	Benefits
180	Construction
190	Contracting
140	Control And Credit
130	Corporate Tax
240	Government Sales
230	IT Helpdesk
210	IT Support
170	Manufacturing
220	NOC
200	Operations
270	Payroll
260	Recruiting
250	Retail Sales
150	Shareholder Services
120	Treasury

16 linhas retornadas em 0,00 segundos

- Compara o valor com cada valor retornado pela subquerie.
- < ANY = menor que o **maior** valor
- > ANY = maior que o **menor** valor

```
SELECT employee_id, last_name, job_id, salary
FROM employees
WHERE salary < ANY 9000, 6000, 4800, 4200
                   (SELECT salary
                    FROM employees
                    WHERE job_id = 'IT_PROG')
AND job_id <> 'IT_PROG';
```

EMPLOYEE_ID	LAST_NAME	JOB_ID	SALARY
124	Mourgos	ST_MAN	5800
141	Rajs	ST_CLERK	3500
142	Davies	ST_CLERK	3100
143	Matos	ST_CLERK	2600
144	Vargas	ST_CLERK	2500

- Compara o valor com todos os valores retornados pela subquerie.

```
SELECT employee_id, last_name, job_id, salary
FROM employees
WHERE salary < ALL 9000, 6000, 4200
AND job_id <> 'IT_PROG';
```

(SELECT salary
FROM employees
WHERE job_id = 'IT_PROG')

EMPLOYEE_ID	LAST_NAME	JOB_ID	SALARY
141	Rajs	ST_CLERK	3500
142	Davies	ST_CLERK	3100
143	Matos	ST_CLERK	2600
144	Vargas	ST_CLERK	2500

- A cláusula **having** filtra o resultado do agrupamento (group by).
- O exemplo abaixo mostra os códigos de departamentos e os menores salários da tabela de empregados, agrupado por departamento, que tenha o menor salário maior que o menor salário dos funcionários do departamento 50.

```
SELECT    department_id, MIN(salary)
FROM      employees
GROUP BY  department_id
HAVING    MIN(salary) >
```

2

3

2100

```
(SELECT MIN(salary)
FROM    employees
WHERE   department_id = 50)
```

1

1 Retorno da querie mais interna

```
(SELECT MIN(salary)
FROM employees
WHERE department_id = 50)
```

MIN(SALARY)

2100

2 Retorno da querie principal

```
SELECT department_id, MIN(salary)
FROM employees
GROUP BY department_id
```

DEPARTMENT_ID MIN(SALARY)

100	6900
30	2500
-	7000
90	17000
20	6000
70	10000
110	8300
50	2100
80	6100
40	6500

Há mais de 10 linhas disponíveis. Aumente o seletor de linhas para ver mais linhas.

3 Retorno do having

DEPARTMENT_ID	MIN(SALARY)
100	6900
30	2500
-	7000
90	17000
20	6000
70	10000
110	8300
50	2100
80	6100
40	6500

Há mais de 10 linhas disponíveis. Aumente o seletor de linhas para ver mais linhas.

HAVING MIN(salary) >

MIN(SALARY)

2100

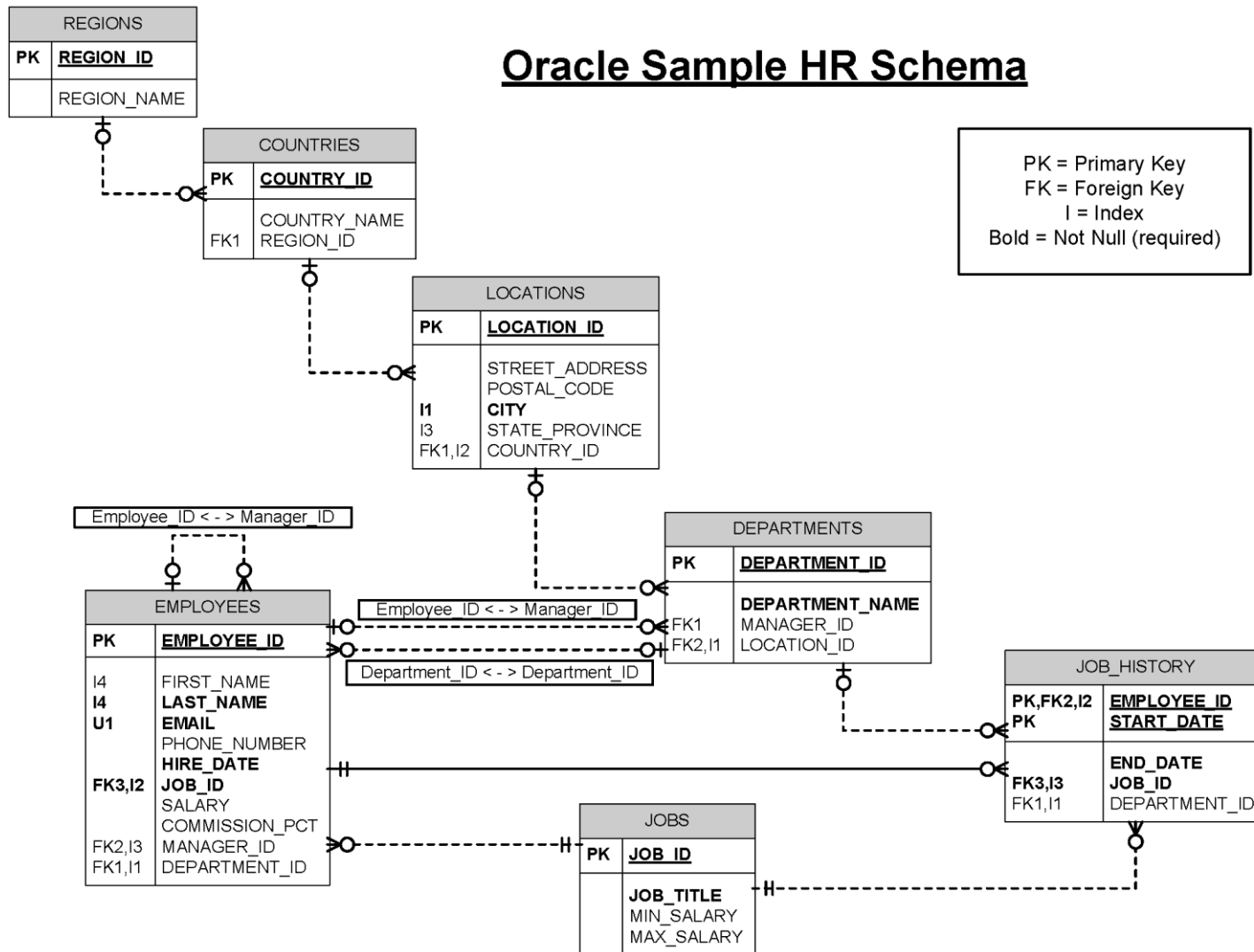
4

Resultado Final

```
SELECT  department_id, MIN(salary)
FROM    employees
GROUP BY department_id
HAVING  MIN(salary) >
        (SELECT MIN(salary)
         FROM  employees
         WHERE department_id = 50)
```

DEPARTMENT_ID	MIN(SALARY)
100	6900
30	2500
-	7000
90	17000
20	6000
70	10000
110	8300
80	6100
40	6500
60	4200
10	4400

Oracle Sample HR Schema



1) Com base no DER da página anterior, resolva os exercícios abaixo

- a) Mostrar o nome, a data de admissão, cargo e o cod. do depto de todos os empregados que tenham o mesmo depto que o funcionário de nome Steven King. Tabela employees.
- b) Mostrar o nome, cod. depto e cargo de todos os empregados que trabalhem nos departamentos que fazem parte da cidade "São Paulo". Tabelas: Employees, departments e locations.
- c) Exibir cod. do depto, nome, cargo de todos os empregados que pertençam ao departamento denominado 'Sales'. Tabelas employees e departments.
- d) Selecionar nome, cargo e salário de todos os empregados cujo salário seja maior que a média salarial de todos os empregados.
- e) Exibir, nome, salário, data de admissão e cod. depto de todos os empregados que trabalhem no mesmo depto que o funcionário de nome Steven King e tenham o salário menor que a média salarial dos empregados.

Exercícios – Série 2

- 1) Crie uma query que mostre o número do empregado, o último nome e o salário de todos os empregados que ganhem mais que a média salarial. Ordenar o resultado em ordem crescente de salários. Seguir alias conforme cabeçalho.

Número Do Empregado	Último Nome	Salário
123	Vollman	6500
203	Mavris	6500
165	Lee	6800
113	Popp	6900
155	Tuvault	7000
161	Sewall	7000
178	Grant	7000
164	Marvins	7200
172	Bates	7300
171	Smith	7400
154	Cambrault	7500
160	Doran	7500
111	Sciarra	7700
112	Urman	7800
122	Kaufling	7900
120	Weiss	8000

- 2) Crie uma query que mostre o último nome, o número do departamento e o código do cargo de todos os empregados cuja localização do departamento seja igual a 1700

Último Nome	Código Do Depto	Código Do Cargo
King	90	AD_PRES
Kochhar	90	AD_VP
De Haan	90	AD_VP
Greenberg	100	FI_MGR
Faviet	100	FI_ACCOUNT
Chen	100	FI_ACCOUNT
Sciarra	100	FI_ACCOUNT
Urman	100	FI_ACCOUNT
Popp	100	FI_ACCOUNT
Raphaely	30	PU_MAN
Khoo	30	PU_CLERK
Baida	30	PU_CLERK
Tobias	30	PU_CLERK
Himuro	30	PU_CLERK
Colmenares	30	PU_CLERK

3) Crie uma query que mostre o último nome e o salário de todos os funcionários que tenham o King como gerente.

LAST_NAME	SALARY
Kochhar	17000
De Haan	17000
Raphaely	11000
Weiss	8000
Fripp	8200
Kaufling	7900
Vollman	6500
Mourgos	5800
Russell	14000
Partners	13500
Errazuriz	12000
Cambraut	11000
Zlotkey	10500
Hartstein	13000

4) Crie uma query que mostre o número do departamento, o último nome e o código do cargo para cada empregado que esteja no departamento 'Executive'

DEPARTMENT_ID	LAST_NAME	JOB_ID
90	King	AD_PRES
90	Kochhar	AD_VP
90	De Haan	AD_VP