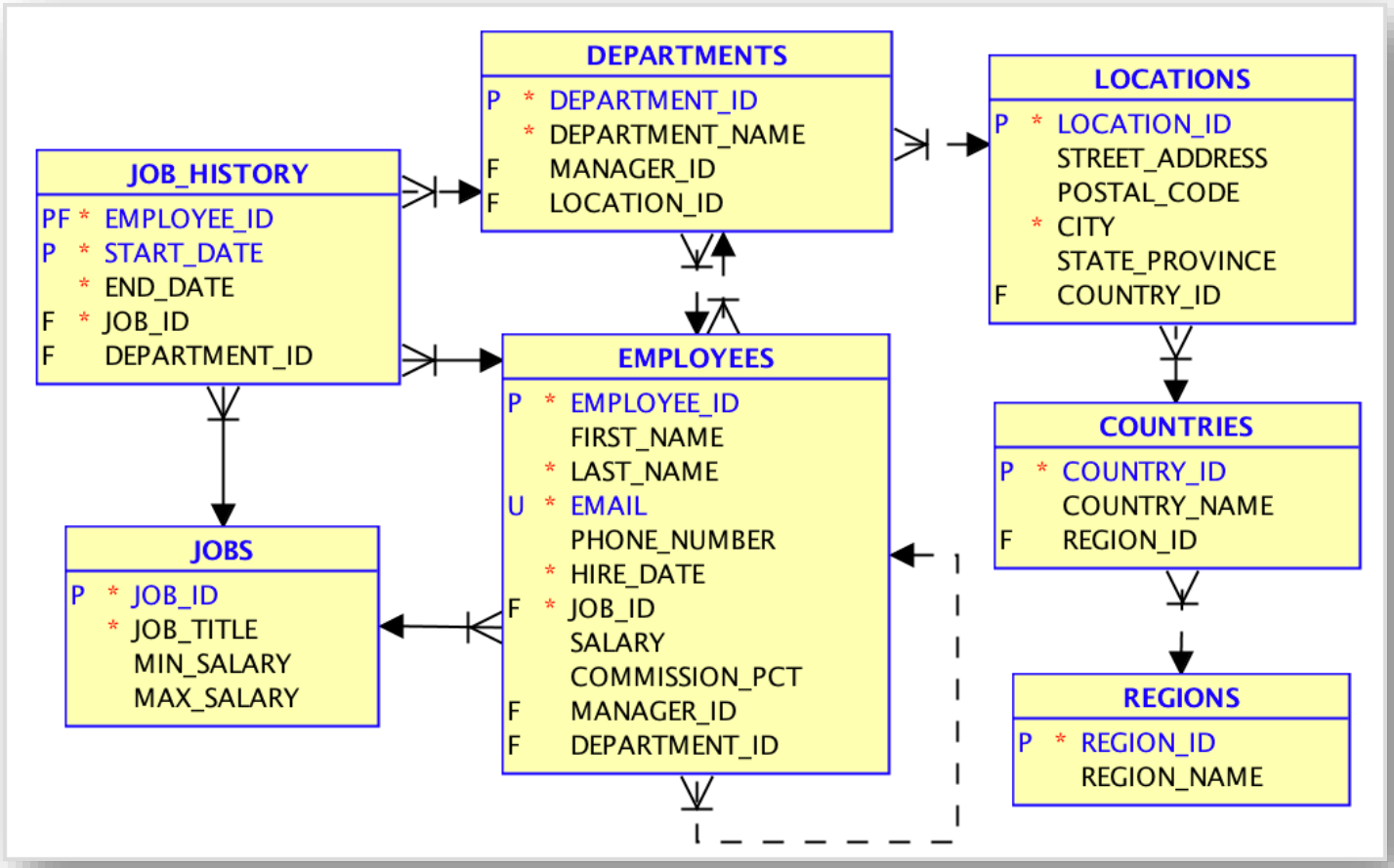
Design e Desenvolvimento de Banco de dados

DQL - SELECT

Luciano Melo
profluciano.melo@fiap.com.br

FIAP
A MELHOR FACULDADE DE TECNOLOGIA

Modelo Utilizado nas Aulas sobre SELECT



Execute o procedimento aprsentado pelo professor para criar o modelo físico do banco de dados de acordo com o modelo relacional acima

Recursos de Instruções SQL `SELECT`

FIAP

Parte 1

Projeção

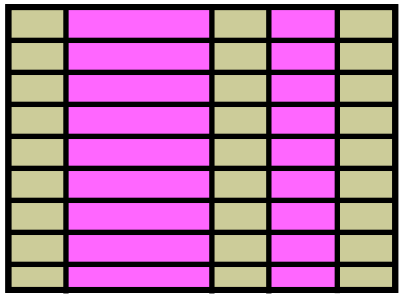


Tabela 1

Parte 2

Seleção

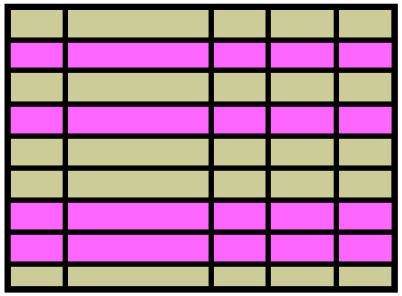


Tabela 1

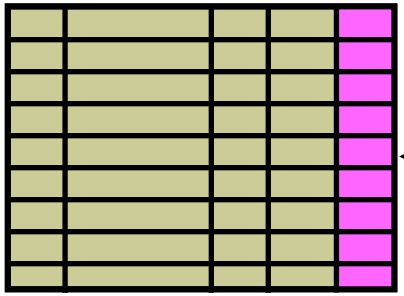


Tabela 1

Junção

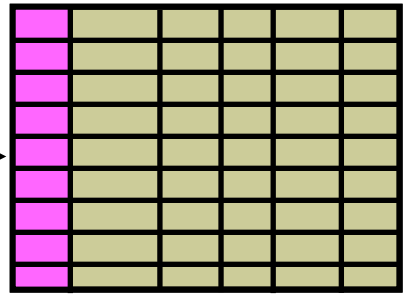
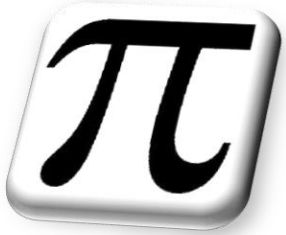


Tabela 2

FIAP



Parte 1 – Projetando dados

Comando **SELECT**

Luciano Melo
profluciano.melo@fiap.com.br

Sintaxe do comando SELECT

FIAP

- **Sintaxe do comando SELECT para operação de projeção**
- **Acessando apenas uma tabela**

```
SELECT *|{[DISTINCT] coluna|expressão [alias],...}
FROM    table;
```

1. Você pode projetar todas as colunas *(*)*
2. Você pode projetar apenas algumas colunas *(coluna,coluna,…)*
3. Você pode projetar expressões (exemplo: salario*2, 'TESTE')
4. Você pode colocar apelidos "alias" para as colunas ou expressões (operação renomear para colunas na álgebra relacional)

FIAP

1. Projetando (mostrando) todas as colunas de uma tabela

```
SELECT *
FROM   departments;
```

DEPARTMENT_ID	DEPARTMENT_NAME	MANAGER_ID	LOCATION_ID
10	Administration	200	1700
20	Marketing	201	1800
50	Shipping	124	1500
60	IT	103	1400
80	Sales	149	2500
90	Executive	100	1700
110	Accounting	205	1700
190	Contracting		1700

8 rows selected.

FIAP

2. Projetando apenas algumas colunas da tabela

```
SELECT department_id, location_id
FROM   departments;
```

DEPARTMENT_ID	LOCATION_ID
10	1700
20	1800
50	1500
60	1400
80	2500
90	1700
110	1700
190	1700

8 rows selected.

FIAP

3. Usando expressões aritiméticas

```
SELECT last_name, salary, salary + 300
FROM   employees;
```

LAST_NAME	SALARY	SALARY+300
King	24000	24300
Kochhar	17000	17300
De Haan	17000	17300
Hunold	9000	9300
Ernst	6000	6300

...

20 rows selected.

FIAP

➢ Precedência de Operadores

```
SELECT last_name, salary, 12*salary+100
FROM   employees;
```

1

LAST_NAME	SALARY	12*SALARY+100
King	24000	288100
Kochhar	17000	204100
De Haan	17000	204100

...

20 rows selected.

```
SELECT last_name, salary, 12*(salary+100)
FROM   employees;
```

2

LAST_NAME	SALARY	12*(SALARY+100)
King	24000	289200
Kochhar	17000	205200
De Haan	17000	205200

...

20 rows selected.

SELECT

FIAP

- **Pratique: Execute os comandos SELECT abaixo**

```
SELECT * FROM  employees;
```

```
SELECT * FROM  departments;
```

```
SELECT first_name, salary, commission_pct FROM employees;
```

```
SELECT first_name, salary*1.2, (salary*1.2)*12 FROM employees;
```

```
SELECT job_title, min_salary, max_salary,
       max_salary-min_salary
FROM  departments;
```

4. Definindo apelidos (alias) para as colunas

Um apelido de coluna:

- Renomeia um cabeçalho de coluna
- É útil em cálculos
- Aparece imediatamente após o nome da coluna (Também é possível incluir a palavra-chave opcional `AS` entre o nome e o apelido da coluna.)
- Requer aspas duplas quando contém espaços ou caracteres especiais, ou quando faz distinção entre maiúsculas e minúsculas

SELECT

FIAP

- **Usando apelido de colunas**

```
SELECT last_name AS name, commission_pct comm
FROM   employees;
```

NAME	COMM
King	
Kochhar	
De Haan	

...

20 rows selected.

```
SELECT last_name "Name" , salary*12 "Annual Salary"
FROM employees;
```

Name	Annual Salary
King	288000
Kochhar	204000
De Haan	204000

...

20 rows selected.

FIAP

➢ Operação de concatenação de strings

Um operador de concatenação:

- Vincula colunas ou strings de caracteres a outras colunas
- É representado por duas barras verticais (||)
- Cria uma coluna resultante que é uma expressão de caracteres

```
SELECT   last_name||job_id AS "Employees"
FROM     employees;
```

Employees
KingAD_PRES
KochharAD_VP
De HaanAD_VP

...

20 rows selected.

FIAP

➢ Strings de caracteres literais

- Um literal é um caractere, uma data ou um número incluído na instrução `SELECT`.
- É necessário delimitar os valores dos literais de caractere e data por **aspas simples.**
- Cada string de caracteres da saída corresponde a apenas uma linha retornada.

SELECT

FIAP

➢ Trabalhando com strings no SELECT

```
SELECT last_name ||' is a '||job_id
       AS "Employee Details"
FROM   employees;
```

Employee Details
King is a AD_PRES
Kochhar is a AD_VP
De Haan is a AD_VP
Hunold is a IT_PROG
Ernst is a IT_PROG
Lorentz is a IT_PROG
Mourgos is a ST_MAN
Rajs is a ST_CLERK

. . .

20 rows selected.

Dica: Se você precisar mostrar uma aspas simples, use duas aspas simples consecutivas.

SELECT

FIAP

- **Pratique: Execute os comandos SELECT abaixo**

```
SELECT first_name||' '||last_name "Nome Completo"
FROM  employees;
```

```
SELECT first_name Nome, salary salario FROM employees;
```

```
SELECT first_name, (salary*1.2)*12 as Salario Annual
FROM employees;
```

```
SELECT first_name||' '||last_name||' ganha '||salary as "Relatório"
FROM  departments;
```

Alguns comandos podem dar erro. Isto é de propósito. Verifique o erro e Corrija-os

FIAP

➢ Entendendo o NULO em banco de dados

- Nulo é ausência de valor. Entretanto, nos referimos ao nulo como “valor nulo”.
- Um valor nulo não está disponível nem é designado e não é conhecido ou aplicável.
- Um valor nulo é diferente de zero ou de um espaço em branco.

```
SELECT last_name, job_id, salary, commission_pct
FROM   employees;
```

LAST_NAME	JOB_ID	SALARY	COMMISSION_PCT
King	AD_PRES	24000	
Kochhar	AD_VP	17000	
...			
Zlotkey	SA_MAN	10500	.2
Abel	SA_REP	11000	.3
Taylor	SA_REP	8600	.2
...			
Gietz	AC_ACCOUNT	8300	

20 rows selected.

FIAP

➢ O efeito do NULO em expressões aritiméticas

- Como o nulo é uma ausência de valor, qualquer expressão aritimética que inclui um valor nulo resulta sempre como nulo.

  Exemplos:

  10 + null = null

  20 * 2 - null = null

  salario + null = null

- Como o nulo é uma ausência de valor, em operações de concatenação de strings com valores nulos, o resultado é a própria string

  Exemplos:

  ‘ABC’ || null = ‘ABC’

  ’20’ || null = ‘20’

  ‘10/10/2012’ || null = ‘10/10/2012’

SELECT

FIAP

- **O efeito do NULO em expressões aritiméticas em comandos SQL**

```
SELECT last_name, 12*salary*commission_pct
FROM   employees;
```

LAST_NAME	12*SALARY*COMMISSION_PCT
...	
Zlotkey	25200
Abel	39600
Taylor	20640
...	
Gietz	

20 rows selected.

FIAP

OPERADOR DISTINCT

A exibição default de consultas mostra todas as linhas, inclusive as linhas duplicadas. Para eliminar linhas duplicadas utilize o *DISTINCT.*

```
SELECT department_id
FROM   employees;
```

1

DEPARTMENT_ID
90
90
90

...

20 rows selected.

```
SELECT DISTINCT department_id
FROM   employees;
```

2

DEPARTMENT_ID
10
20
50

...

8 rows selected.

SELECT

FIAP

- **Pratique: Execute os comandos SELECT abaixo**

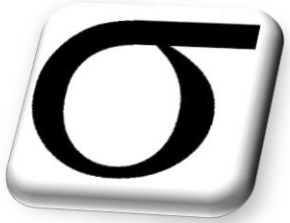
```
SELECT first_name,
(salary*1.2)*12 + (salary*1.2)*12*commision_pct as “Salario Annual”,
FROM employees;
```

```
SELECT job_id from jobs;
```

```
SELECT distinct job_id from jobs
```

```
SELECT distinct department_id, job_id from employees;
```

SELECT



Parte 2

Filtrando e Ordenando Resultados

Luciano Melo
profluciano.melo@fiap.com.br

FIAP

Filtrando Linhas (Seleção)

EMPLOYEES

EMPLOYEE_ID	LAST_NAME	JOB_ID	DEPARTMENT_ID
100	King	AD_PRES	90
101	Kochhar	AD_VP	90
102	De Haan	AD_VP	90
103	Hunold	IT_PROG	60
104	Ernst	IT_PROG	60
107	Lorentz	IT_PROG	60
124	Mourgos	ST_MAN	50

. . .

20 rows selected.

"recuperar todos os funcionários do departamento 90"

EMPLOYEE_ID	LAST_NAME	JOB_ID	DEPARTMENT_ID
100	King	AD_PRES	90
101	Kochhar	AD_VP	90
102	De Haan	AD_VP	90

FIAP

Sintaxe

- **Restrinja as linhas retornadas com a cláusula `WHERE`:**

```
SELECT *|{[DISTINCT] column|expression [alias],...}
FROM   table
[WHERE condition(s)];
```

- **A cláusula `WHERE` é especificada após a cláusula `FROM`.**

Exemplo:

```
SELECT employee_id, last_name, job_id, department_id
FROM   employees
WHERE  department_id = 90 ;
```

EMPLOYEE_ID	LAST_NAME	JOB_ID	DEPARTMENT_ID
100	King	AD_PRES	90
101	Kochhar	AD_VP	90
102	De Haan	AD_VP	90

FIAP

Strings de Caracteres e Datas

- As strings de caracteres e os valores de data são delimitados por aspas simples.
- Os valores de caractere fazem distinção entre maiúsculas e minúsculas, e os valores de data fazem distinção de formato.
- O formato default da data depende da linguagem e formatação do banco.

```
SELECT last_name, job_id, department_id
FROM   employees
WHERE  last_name = 'Whalen' ;
```

FIAP

Operadores

Operador	Significado
=	Igual a
>	Maior que
>=	Maior que ou igual a
<	Menor que
<=	Menor que ou igual a
<>	Diferente de
`BETWEEN ...AND...`	Entre dois valores (inclusivo)
`IN(set)`	Corresponde a qualquer valor
`LIKE`	Corresponde a um padrão de
`IS NULL`	É um valor nulo

Operador BETWEEN

Use a condição BETWEEN para exibir linhas com base em uma faixa de valores:

```
SELECT last_name, salary
FROM   employees
WHERE  salary BETWEEN 2500 AND 3500 ;
```

Limite inferior Limite superior

LAST_NAME	SALARY
Rajs	3500
Davies	3100
Matos	2600
Vargas	2500

FIAP

Operador IN

Use a condição de associação IN para testar os valores de uma lista:

```
SELECT employee_id, last_name, salary, manager_id
FROM   employees
WHERE  manager_id IN (100, 101, 201) ;
```

EMPLOYEE_ID	LAST_NAME	SALARY	MANAGER_ID
202	Fay	6000	201
200	Whalen	4400	101
205	Higgins	12000	101
101	Kochhar	17000	100
102	De Haan	17000	100
124	Mourgos	5800	100
149	Zlotkey	10500	100
201	Hartstein	13000	100

8 rows selected.

FIAP

Operador `LIKE`

- **Use a condição `LIKE` para executar pesquisas com curinga de valores válidos de strings de pesquisa.**
- **As condições de pesquisa podem conter números ou caracteres literais:**
  - **`%` indica zero ou vários caracteres.**
  - **`_` indica um caractere.**

```
SELECT   first_name
FROM     employees
WHERE    first_name LIKE 'S%' ;
```

FIAP

Operador LIKE

- Use a condição LIKE para executar pesquisas com curinga de valores válidos de strings de pesquisa.
- As condições de pesquisa podem conter números ou caracteres literais:
  - % indica zero ou vários caracteres.
  - _ indica um caractere.

```
SELECT   first_name
FROM     employees
WHERE    first_name LIKE 'S%' ;
```

```
SELECT last_name
FROM   employees
WHERE  last_name LIKE '_o%' ;
```

FIAP

Operador `LIKE: Exemplos`

Expressão	Explicação
LIKE 'A%'	Todas as palavras que iniciem com A.
LIKE '%A'	Todas que terminem com a letra A.
LIKE '%A%'	Todas que tenham a letra A em qualquer posição.
LIKE 'A_'	String de dois caracteres que tenham a primeira letra A e o segundo caractere seja qualquer outro.
LIKE'_A'	String de dois caracteres cujo primeiro caractere seja qualquer um e a última seja A.
LIKE'_A_'	String de três caracteres cuja segunda letra seja A independentemente do primeiro ou do último caractere.
LIKE '%A'_	Todos que tenham a letra A na penúltima posição e aúltima seja qualquer outro caractere.
LIKE '_A%'	Todos que tenham a letra A na segunda posição e o primeiro caractere seja qualquer um.

FIAP

Operador IS NULL

Teste valores nulos com o operador `IS NULL`.

```
SELECT last_name, manager_id
FROM   employees
WHERE  manager_id IS NULL ;
```

LAST_NAME	MANAGER_ID
King	

SELECT

FIAP

Pratique: Execute os comandos SELECT abaixo

Alguns comandos podem dar erro. Isto é de propósito. Verifique o erro e Corrija-os

```
SELECT * FROM employees WHERE job_id = 'SA_MAN';
```

```
SELECT First_name, salary FROM employees WHERE salary >= 10000;
```

```
SELECT first_name, job_id FROM employees where
department_id in (20,30,40);
```

```
SELECT * from employees from salary between 10000 and 20000;
```

```
SELECT first_name||' '||last_name nome FROM employees
Where first_name like 'A%';
```

```
SELECT * FROM employees WHERE department_id IS NULL;
```

FIAP

Operadores Lógicos

Operador	Significado
AND	Retornará `TRUE` se as duas condições componentes forem verdadeiras
OR	Retornará `TRUE` se uma das condições componentes for verdadeira
NOT	Retornará `TRUE` se a condição seguinte for falsa

FIAP

Usando o Operador AND

AND exige que as duas condições sejam verdadeiras:

```
SELECT employee_id, last_name, job_id, salary
FROM   employees
WHERE  salary >=10000
AND    job_id LIKE '%MAN%' ;
```

EMPLOYEE_ID	LAST_NAME	JOB_ID	SALARY
149	Zlotkey	SA_MAN	10500
201	Hartstein	MK_MAN	13000

FIAP

Usando o Operador OR

OR exige que uma das condições seja verdadeira:

```
SELECT employee_id, last_name, job_id, salary
FROM   employees
WHERE  salary >= 10000
OR     job_id LIKE '%MAN%' ;
```

EMPLOYEE_ID	LAST_NAME	JOB_ID	SALARY
100	King	AD_PRES	24000
101	Kochhar	AD_VP	17000
102	De Haan	AD_VP	17000
124	Mourgos	ST_MAN	5800
149	Zlotkey	SA_MAN	10500
174	Abel	SA_REP	11000
201	Hartstein	MK_MAN	13000
205	Higgins	AC_MGR	12000

8 rows selected.

FIAP

Usando o Operador NOT

```
SELECT last_name, job_id
FROM   employees
WHERE  job_id
       NOT IN ('IT_PROG', 'ST_CLERK', 'SA_REP') ;
```

LAST_NAME	JOB_ID
King	AD_PRES
Kochhar	AD_VP
De Haan	AD_VP
Mourgos	ST_MAN
Zlotkey	SA_MAN
Whalen	AD_ASST
Hartstein	MK_MAN
Fay	MK_REP
Higgins	AC_MGR
Gietz	AC_ACCOUNT

10 rows selected.

FIAP

Regras de Precedência

Operador	Significado
1	Operadores aritméticos
2	Operador de concatenação
3	Condições de comparação
4	`IS [NOT] NULL, LIKE, [NOT] IN`
5	`[NOT] BETWEEN`
6	Diferente de
7	Condição lógica `NOT`
8	Condição lógica `AND`
9	Condição lógica `OR`

Você pode usar parênteses para sobrepor as regras de precedência.

FIAP

Exemplo de Precedência

```
SELECT last_name, job_id, salary
FROM   employees
WHERE  job_id = 'SA_REP'
OR     job_id = 'AD_PRES'
AND    salary > 15000;
```

1

LAST_NAME	JOB_ID	SALARY
King	AD_PRES	24000
Abel	SA_REP	11000
Taylor	SA_REP	8600
Grant	SA_REP	7000

```
SELECT last_name, job_id, salary
FROM   employees
WHERE  (job_id = 'SA_REP'
OR     job_id = 'AD_PRES')
AND    salary > 15000;
```

2

LAST_NAME	JOB_ID	SALARY
King	AD_PRES	24000

FIAP

Ordenando resultados: ORDER BY

- Ordenar as linhas recuperadas com a cláusula ORDER BY:
  - ASC: ordem crescente, default
  - DESC: ordem decrescente
- A cláusula ORDER BY é inserida por último na instrução SELECT:

```
SELECT   last_name, job_id, department_id, hire_date
FROM employees
ORDER BY hire_date;
```

LAST_NAME	JOB_ID	DEPARTMENT_ID	HIRE_DATE
King	AD_PRES	90	17-JUN-87
Whalen	AD_ASST	10	17-SEP-87
Kochhar	AD_VP	90	21-SEP-89
Hunold	IT_PROG	60	03-JAN-90
Ernst	IT_PROG	60	21-MAY-91

...

20 rows selected.

FIAP

Exemplos de Ordenação

- **Ordenação em ordem decrescente:**

```
SELECT   last_name, job_id, department_id, hire_date
FROM employees
ORDER BY hire_date DESC;
```

1

- **Ordenação por apelido de coluna:**

```
SELECT employee_id, last_name, salary*12 annsal
FROM   employees
ORDER BY annsal ;
```

2

- **Ordenação por várias colunas:**

```
SELECT last_name, department_id, salary
FROM   employees
ORDER BY department_id, salary DESC;
```

3

FIAP

Top N linhas

- **Retornando os 10 maiores salários**

```
select first_name, salary
from employees
order by salary desc
fetch first 7 rows only;
```

1

```
FIRST_NAME                SALARY
------------------------- ----------
Steven                         24000
Neena                          17000
Lex                            17000
John                           14000
Karen                          13500
Michael                        13000
Shelley                        12008

7 rows selected.
```

FIAP

Top N linhas

- **Retornando os 10 maiores salários, mostrando todos os valores referentes ao último valor**

```
select first_name, salary
from employees
order by salary desc
fetch first 7 rows with ties;
```

2

```
FIRST_NAME                SALARY
------------------------- ----------
Steven                         24000
Neena                          17000
Lex                            17000
John                           14000
Karen                          13500
Michael                        13000
Shelley                        12008
Nancy                          12008

8 rows selected.
```

FIAP

Top N linhas – Ignorando x primeiras linhas

- **Retornando os 10 maiores salários menores que os 4 primeiros (mesmo que: ignorando os 4 maiores)**

```
select first_name, salary
from employees
order by salary desc
OFFSET 4 ROWS FETCH NEXT 10 ROWS ONLY;
```

3

```
FIRST_NAME                SALARY
------------------------- ----------
Karen                          13500
Michael                        13000
Shelley                        12008
Nancy                          12008
Alberto                        12000
Lisa                           11500
Den                            11000
Ellen                          11000
Gerald                         11000
Eleni                          10500

10 rows selected.
```

Retornando x% de linhas

- **Retornando 20 % das linhas da tabela employees**

```
SELECT * from employees
ORDER BY first_name
FETCH FIRST 20 PERCENT ROWS ONLY;
```

4

```
SELECT * from employees
FETCH FIRST 20 PERCENT ROWS ONLY;
```

5

SELECT

FIAP

Pratique: Execute os comandos SELECT abaixo

Alguns comandos podem dar erro. Isto é de propósito. Verifique o erro e Corrija-os

```
SELECT * FROM employees WHERE job_id = 'SA_MAN' and salary > 10000;
```

```
SELECT First_name, job_id, salary, department_id FROM employees
WHERE department_id not in (10,20) OR job_id = 'IT_PROG'
ORDER BY department_id, salary desc;
```

```
SELECT first_name||' '||last_name nome,
       salary*12 "salario anual"
FROM employees
Where (job_id='IT_PROG' or job_id='ST_CLERK') and salary*12 > 50000
Order by "salario anual" desc
Fetch first 10 rows only;
```

```
SELECT * FROM employees order by department_id;
```

Exercícios

Partes 1 e 2

profluciano.melo@fiap.com.br

FIM

profluciano.melo@fiap.com.br

Bibliografia Utilizada

Oracle SQL References: http://docs.oracle.com
Manuais Oracle – Introdução ao Oracle e SQL I e II