Instrução SELECT

Objetivos

- Descrevendo uma tabela
- Diferenciar projeções de seleções
- Sintaxe básica da instrução SELECT
- Operadores
- Strings, valores literais, valor NULL
- Linhas repetidas
- Cláusula WHERE
- Funções
- Cláusula ORDER BY

Descrevendo Estrutura de uma Tabela

- Comando: **`describe`** ou **`descr`**

```
1  descr hr.employees
```

TABLE EMPLOYEES

Column	Null?	Type
EMPLOYEE_ID	NOT NULL	NUMBER(6,0)
FIRST_NAME	-	VARCHAR2(20)
LAST_NAME	NOT NULL	VARCHAR2(25)
EMAIL	NOT NULL	VARCHAR2(25)
PHONE_NUMBER	-	VARCHAR2(20)
HIRE_DATE	NOT NULL	DATE

```
1  descr hr.departments
```

TABLE DEPARTMENTS

Column	Null?	Type
DEPARTMENT_ID	NOT NULL	NUMBER(4,0)
DEPARTMENT_NAME	NOT NULL	VARCHAR2(30)
MANAGER_ID	-	NUMBER(6,0)
LOCATION_ID	-	NUMBER(4,0)

SELECT Cláusulas

- SELECT → quais dados (colunas)
- FROM → quais informações (tabelas)
- WHERE → filtros / restrições (linhas)
- ORDER BY → em que ordem

```
select  first_name, last_name, job_id, salary
from    hr.employees
where   salary > 5000
order   by salary ;
```

Projeção, Seleção e Joins

Projeção

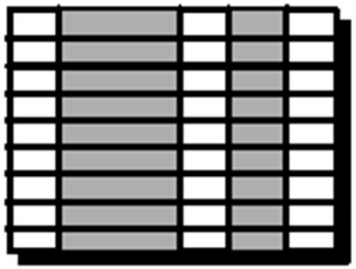

Tabela 1

Seleção

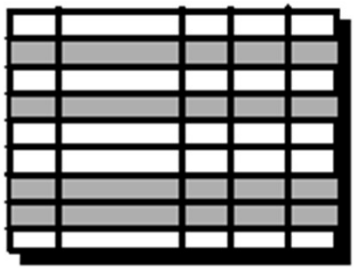

Tabela 2

Join

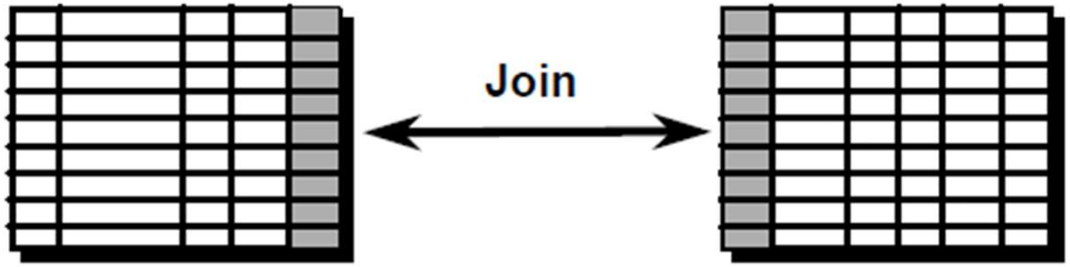

Tabela 1

Tabela 2

Select Básico

- Clausulas SELECT e FROM são obrigatórias

```
select   *
from     tabela;
```

todas as colunas

```
select   col1, col2 ...
from     tabela;
```

colunas específicas (projeção)

```
select   col1 as descricao, ...
from     tabela;
```

alias de colunas (apelidos)

```
select   distinct col1, col2
from     tabela;
```

elimina linhas repetidas

Cláusulas FROM é obrigatória!
* Tabela DUAL *

```
1  select 100;
```

```
ORA-00923: FROM keyword not found where expected
```

```
desc sys.dual;
```

TABLE DUAL

Column	Null?	Type
DUMMY	-	VARCHAR2(1)

- Tabela DUAL é usada

```
1  select 100 from dual;
```

100
100

```
select * from dual;
```

DUMMY
X

Aliases (Apelidos) de Colunas

- Aliases simples
- Aliases complexos (entre aspas duplas)
- Concatenação strings ||

```
1  select  first_name as nome, last_name as SOBRENOME,
2          first_name || ' ' || last_name as "Nome Completo"
3  from    hr.employees
4
```

NOME	SOBRENOME	Nome Completo
Ellen	Abel	Ellen Abel
Sundar	Ande	Sundar Ande
Mozhe	Atkinson	Mozhe Atkinson
David	Austin	David Austin

SELECT – regras gerais

- Palavras chave e nomes tabelas/colunas
  - são case Insensitive
  - Valores literais (variáveis) SÃO case sensitive
- Pode ficar em múltiplas linhas
- No final precisa de um “;” ou uma “/”
  - Se for “/” que que ser na última linha sozinha
- Nada pode ser abreviado
- Boas práticas **(legibilidade)**
  - Organizar clausulas em diferentes linhas
  - Padronizar uppercase/lowercase
  - Identação…

```
SELECT   col1, col2 ...
FROM     tabela;

select   col1, COL2 ...
from     taBEla;

select   col1, COL2 ...
from     taBEla
/
```

Operadores Áritméticos

- Multiplicação *
- Divisão /
- Adição +
- Subtração -

- Precedência
- Use parêntesis ((…) …)

```
select   last_name, salary,
         salary*1.1   as salary_with_10pct
from     employees;

select   last_name, salary,
         1000+salary*12   as sal_ann_1k
from     employees;

select   last_name, salary,
         (1000+salary)*12   as sal_ann_1k
from     employees;
```

O valor NULL

- NULL
  - Conceitualmente NULL é um valor não disponível, ou não conhecido
    - Nao é a mesma coisa que o valor zero, nem espaço em branco
    - Portanto não é igual, nem diferente
  - IS NULL buscar registros nulos

```
select first_name from hr.employees where manager_id = NULL;

    no data found

select first_name from hr.employees where manager_id is null;

    FIRST_NAME
    ------------
    Steven
```

O valor NULL

- Em operações com NULL, resultado é NULL

```
select  first_name, salary, commission_pct, salary* commission_pct
from    hr.employees
```

FIRST_NAME	SALARY	COMMISSION_PCT	SALARY*COMMISSION_PCT
Adam	8200		
Alana	3100		
Alberto	12000	.3	3600
Alexander	9000		
Alexander	3100		
Alexis	4100		
Allan	9000	.35	3150
Alyssa	8800	.25	2200
Amit	6200	.1	620

O valor NULL – função NVL

- Em operações com NULL, resultado é NULL

```
select  first_name, salary, commission_pct, nvl(salary* commission_pct, 0)
from    hr.employees
```

FIRST_NAME	SALARY	COMMISSION_PCT	NVL(SALARY*COMMISSION_PCT,0)
Adam	8200		0
Alana	3100		0
Alberto	12000	.3	3600
Alexander	9000		0
Alexander	3100		0
Alexis	4100		0
Allan	9000	.35	3150
Alyssa	8800	.25	2200
Amit	6200	.1	620

Concatenação de strings

- Duas barras verticais ||

```
1  select  first_name, last_name,
2          first_name || ' ' || last_name as full_name
3  from    hr.employees
4
```

FIRST_NAME	LAST_NAME	FULL_NAME
Ellen	Abel	Ellen Abel
Sundar	Ande	Sundar Ande
Mozhe	Atkinson	Mozhe Atkinson
David	Austin	David Austin

Valores Literais

Valores fixos

- Caracteres (entre aspas simples)
- Numéricos
- Datas
  (função to_date)

```
1 select  first_name || ' - ' || last_name as full_name,
2     salary * 12 as annual_salary,
3     to_date('01/01/2024', 'dd/mm/yyyy')
4 from    hr.employees
```

FULL_NAME	ANNUAL_SALARY	TO_DATE('01/01/2024','DD/MM/YYYY')
Steven - King	288000	01-JAN-24
Neena - Kochhar	204000	01-JAN-24
Lex - De Haan	204000	01-JAN-24
Alexander - Hunold	108000	01-JAN-24

Valores Repetidos

- SELECT DISTINCT …

```
1 select  department_id
2 from    hr.employees
```

DEPARTMENT_ID
90
90
90
60
60
60
60

```
1 select  DISTINCT department_id
2 from    hr.employees
```

DEPARTMENT_ID
50
40
110
90
30
70

Cláusula WHERE

- Limitar (ou restringir) linhas no resultado (result-set)
- SELECT … FROM … **WHERE** … AND …

```
select  first_name, last_name,
        salary, department_id
from    hr.employees
where   department_id = 90;

select  first_name, last_name,
        salary, department_id
from    hr.employees
where   department_id = 90
and     salary > 20000;
```

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

FIRST_NAME	LAST_NAME	SALARY	DEPARTMENT_ID
Steven	King	24000	90

Cláusula WHERE

```
1  select  first_name, last_name, salary, department_id, hire_date
2  from    hr.employees
3  where   department_id = 90;
4
```

FIRST_NAME	LAST_NAME	SALARY	DEPARTMENT_ID	HIRE_DATE
Steven	King	24000	90	17-JUN-03
Neena	Kochhar	17000	90	21-SEP-05
Lex	De Haan	17000	90	13-JAN-01

```
1  select  first_name, last_name, salary, department_id, hire_date
2  from    hr.employees
3  where   department_id = 90
4  and     last_name = 'King';
```

FIRST_NAME	LAST_NAME	SALARY	DEPARTMENT_ID	HIRE_DATE
Steven	King	24000	90	17-JUN-03

Cláusula WHERE

```
1  select  first_name, last_name, salary, department_id, hire_date
2  from    hr.employees
3  where   department_id = 90;
4
```

FIRST_NAME	LAST_NAME	SALARY	DEPARTMENT_ID	HIRE_DATE
Steven	King	24000	90	17-JUN-03
Neena	Kochhar	17000	90	21-SEP-05
Lex	De Haan	17000	90	13-JAN-01

```
1  select  first_name, last_name, salary, department_id, hire_date
2  from    hr.employees
3  where   department_id = 90
4  and     hire_date > to_date('01/01/2002','dd/mm/yyyy');
```

FIRST_NAME	LAST_NAME	SALARY	DEPARTMENT_ID	HIRE_DATE
Steven	King	24000	90	17-JUN-03
Neena	Kochhar	17000	90	21-SEP-05

Operadores de Comparação

```
=                              Igual a
>                              Maior que
>=                             Maior ou igual que
<                              Menor que
<=                             Menor ou igual que
<>  ou  !=                     Diferente de

between  <valor1>  and  <valor2>
in (lista de valores)
like 'string%'
IS NULL
```

Condições Lógicas

- AND
- OR
- NOT

  – Se utilizar ambos AND e OR usar parêntesis para organizar e explicitar a lógica apropriada.

Condições Lógicas

- Identação também pode ajudar

```
1  select  first_name, last_name, salary, department_id, hire_date
2  from    hr.employees
3  where   department_id = 90
4  or  (hire_date > to_date('01/01/2002','dd/mm/yyyy')
5       and    department_id = 60);
6
```

FIRST_NAME	LAST_NAME	SALARY	DEPARTMENT_ID	HIRE_DATE
Steven	King	24000	90	17/06/2003 00:00
Neena	Kochhar	17000	90	21/09/2005 00:00
Lex	De Haan	17000	90	13/01/2001 00:00
Alexander	Hunold	9000	60	03/01/2006 00:00
Bruce	Ernst	6000	60	21/05/2007 00:00
David	Austin	4800	60	25/06/2005 00:00
Valli	Pataballa	4800	60	05/02/2006 00:00
Diana	Lorentz	4200	60	07/02/2007 00:00

Condições Lógicas

```
1  select  first_name, last_name, salary, department_id, hire_date, job_id
2  from    hr.employees
3  where   salary > 10000
4  and     job_id like '%MAN%'
5
```

FIRST_NAME	LAST_NAME	SALARY	DEPARTMENT_ID	HIRE_DATE	JOB_ID
Den	Raphaely	11000	30	07/12/2002 00:00	PU_MAN
John	Russell	14000	80	01/10/2004 00:00	SA_MAN
Karen	Partners	13500	80	05/01/2005 00:00	SA_MAN
Alberto	Errazuriz	12000	80	10/03/2005 00:00	SA_MAN
Gerald	Cambrault	11000	80	15/10/2007 00:00	SA_MAN

Ordenação

- ORDER BY

```
1  select  first_name, last_name, salary, department_id, hire_date, job_id
2  from    hr.employees
3  where   salary > 10000
4  and     job_id like '%MAN%'
5  order   by salary;
```

FIRST_NAME	LAST_NAME	SALARY	DEPARTMENT_ID	HIRE_DATE	JOB_ID
Eleni	Zlotkey	10500	80	29/01/2008 00:00	SA_MAN
Den	Raphaely	11000	30	07/12/2002 00:00	PU_MAN
Gerald	Cambrault	11000	80	15/10/2007 00:00	SA_MAN
Alberto	Errazuriz	12000	80	10/03/2005 00:00	SA_MAN
Michael	Hartstein	13000	20	17/02/2004 00:00	MK_MAN
Karen	Partners	13500	80	05/01/2005 00:00	SA_MAN
John	Russell	14000	80	01/10/2004 00:00	SA_MAN

Ordenação

- ORDER BY DESC

```
1  select  first_name, last_name, salary, department_id, hire_date, job_id
2  from    hr.employees
3  where   salary > 10000
4  and     job_id like '%MAN%'
5  order   by salary DESC;
```

FIRST_NAME	LAST_NAME	SALARY	DEPARTMENT_ID	HIRE_DATE	JOB_ID
John	Russell	14000	80	01/10/2004 00:00	SA_MAN
Karen	Partners	13500	80	05/01/2005 00:00	SA_MAN
Michael	Hartstein	13000	20	17/02/2004 00:00	MK_MAN
Alberto	Errazuriz	12000	80	10/03/2005 00:00	SA_MAN
Gerald	Cambrault	11000	80	15/10/2007 00:00	SA_MAN
Den	Raphaely	11000	30	07/12/2002 00:00	PU_MAN
Eleni	Zlotkey	10500	80	29/01/2008 00:00	SA_MAN

Consultando Dicionário de Dados

DBA_TABLES → todas tabelas do BD

ALL_TABLES → todas tabelas que o usuário conectado possui acesso

USER_TABLES → todas as tabelas pertencentes ao schema do usuário conectado

Resumo e Dúvidas

- Dúvidas ou comentários ... ?