

BASE DE DADOS



Teórico-Práticas
Ano Lectivo 2018/2019

SQL
Structured Query
Language – DML

Views & Funções
Oracle

Views

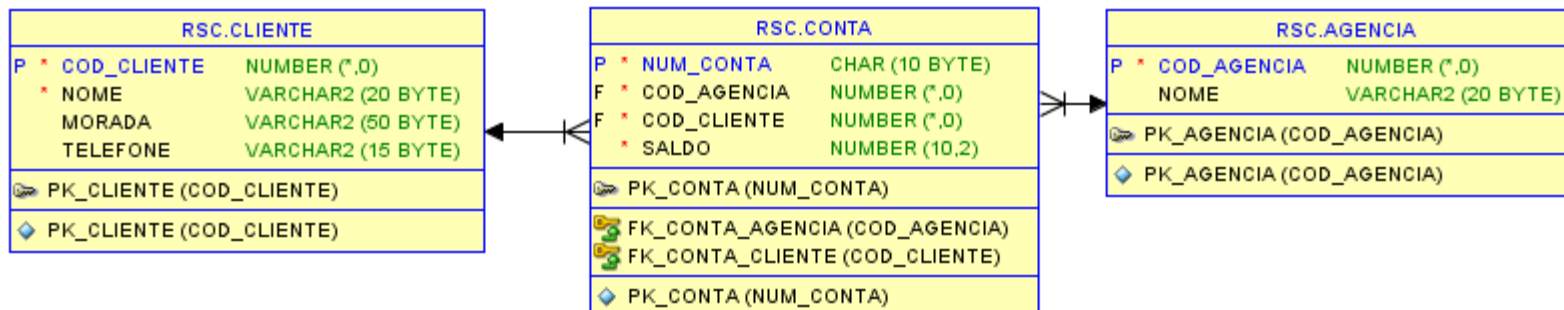
- Uma view (vista) é uma tabela virtual cuja estrutura e conteúdo é definido através de uma query. A view é criada dinamicamente a partir do momento que é invocada. Uma view actua como um filtro sobre os dados da BD, e podem ser resultados de uma ou mais tabelas.
- As views são normalmente utilizadas para simplificar a manipulação dos dados, e definir os dados na perspectiva do utilizador.
- Podem também ser utilizadas como mecanismos de segurança de dados, pois é possível definir seguranças ao nível de uma view.
- É também possível alterar informação através de views
- As views podem ser usadas em pesquisas tal qual uma tabela normal

CREATE VIEW Nome_view AS <Query>

Linguagem SQL – DDL e DML

Exemplo

```
create table cliente(  
  cod_cliente int constraint pk_cliente primary key,  
  nome varchar(20) not null,  
  morada varchar(50),  
  telefone varchar(15));  
  
create table agencia(  
  cod_agencia int constraint pk_agencia primary key,  
  nome varchar(20));  
  
create table conta(  
  num_conta char(10) constraint pk_conta primary key,  
  cod_agencia int not null constraint fk_conta_agencia  
    references agencia(cod_agencia),  
  cod_cliente int not null constraint fk_conta_cliente  
    references cliente(cod_cliente),  
  saldo numeric(10, 2) not null);
```



Linguagem SQL – DDL e DML

Exemplo

```
-- Inserir dados nas tabelas
```

```
insert into cliente (cod_cliente, nome) values(1, 'A1');  
insert into cliente (cod_cliente, nome) values(2, 'B1');  
insert into cliente (cod_cliente, nome) values(3, 'C1');
```

```
insert into agencia (cod_agencia, nome) values(1, 'AgA1');  
insert into agencia (cod_agencia, nome) values(2, 'AgB1');  
insert into agencia (cod_agencia, nome) values(3, 'AgC1');
```

```
insert into conta (num_conta, cod_cliente, cod_agencia, saldo) values (1, 1, 1, 12000);  
insert into conta (num_conta, cod_cliente, cod_agencia, saldo) values (2, 1, 2, 8000);  
insert into conta (num_conta, cod_cliente, cod_agencia, saldo) values (3, 2, 1, 1000);  
insert into conta (num_conta, cod_cliente, cod_agencia, saldo) values (4, 3, 3, 16000);  
insert into conta (num_conta, cod_cliente, cod_agencia, saldo) values (5, 3, 2, 10000);  
insert into conta (num_conta, cod_cliente, cod_agencia, saldo) values (6, 3, 1, 500);
```

Exemplo

- Suponhamos o seguinte conceito existente no nosso sistema:
 - Os "Super clientes" são aqueles cujo saldo individual de pelo menos uma das suas contas é superior ao saldo médio das contas do banco
 - Há interesse em ter uma tabela que indique quais são os "Super clientes".
 - A informação necessária é o código, nome e saldo total (todas as suas contas)

Exemplo

- Como é óbvio, esta informação não é estática.
 - Num instante uma determinada pessoa é "Super cliente" e noutro pode deixar de o ser. Basta movimentar as contas.
- Para resolver este problema, a solução passa pela criação de uma view.

Linguagem SQL – DDL e DML



Exemplo

A informação pretendida é dada pelo seguinte Select:

```
-- Exemplo | Solução--
select a.cod_cliente, a.nome, sum(b.saldo)
  from cliente a, conta b
 where a.cod_cliente = b.cod_cliente
 and a.cod_cliente in (select a.cod_cliente
                       from cliente a, conta b
                      where a.cod_cliente = b.cod_cliente
                        and b.saldo > (select avg(saldo) from conta))
 group by a.cod_cliente, a.nome;
```

Exemplo

- Não é um comando SQL trivial. E se é necessário usar frequentemente esta informação relativa aos clientes, ou...
 - Há utilizadores do sistema que não sabem SQL...
- O administrador do sistema escreve uma view que resolve vários problemas:
 - Esconde dos utilizadores a complexidade do Sql
 - Ficamos com uma "tabela" que contém a informação sobre os "Super clientes"
 - Esconde dos utilizadores alguns dos campos das tabelas "Cliente" e "Conta".
- É importante perceber que muitas vezes é importante esconder parte da informação que está contida nas tabelas.

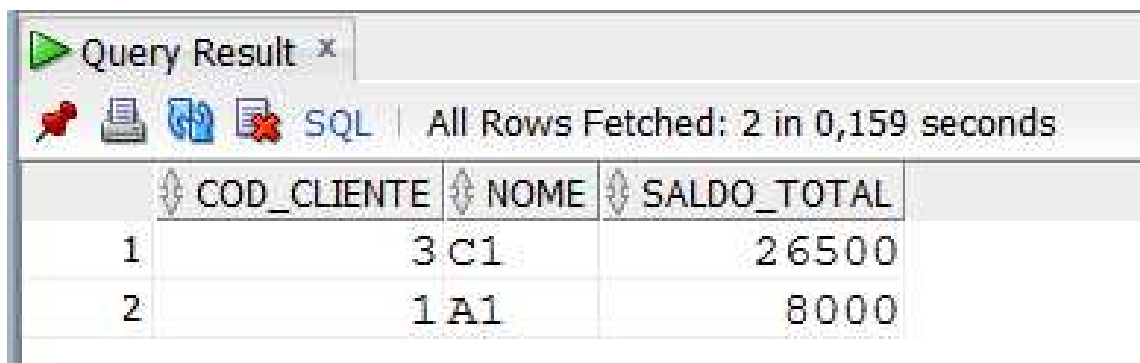
Linguagem SQL – DDL e DML

Exemplo

```
create view "Super clientes" as
select a.cod_cliente, a.nome, sum(b.saldo) saldo_total
  from cliente a, conta b
 where a.cod_cliente = b.cod_cliente
    and a.cod_cliente in (select a.cod_cliente
                          from cliente a, conta b
                         where a.cod_cliente = b.cod_cliente
                          and b.saldo > (select avg(saldo) from conta))
 group by a.cod_cliente, a.nome;

-- Para usar a View--

select * from "Super clientes";
```



| | COD_CLIENTE | NOME | SALDO_TOTAL |
|---|-------------|------|-------------|
| 1 | 3 | C1 | 26500 |
| 2 | 1 | A1 | 8000 |

Linguagem SQL – DDL e DML



Exemplo

O nosso modelo de dados passou a ser:







- cliente(cod_cliente, nome, morada, telefone)
- agencia(cod_agencia, nome)
- conta(num_conta, cod_agencia, cod_cliente, saldo)
- "Super clientes"(cod_cliente, nome, saldo_total)

Linguagem SQL – DDL e DML

Read Only

- É possível o utilizador aceder a definição da view. Para o evitar podemos usar uma opção de **read only**:

```
create view "Super clientes read only" as
select a.cod_cliente, a.nome, sum(b.saldo) saldo_total
  from cliente a, conta b
 where a.cod_cliente = b.cod_cliente
    and a.cod_cliente in (select a.cod_cliente
                        from cliente a, conta b
                       where a.cod_cliente = b.cod_cliente
                        and b.saldo > (select avg(saldo) from conta))
group by a.cod_cliente, a.nome
with read only;
```

| Columns | Data | Grants | Dependencies | Details | Triggers | SQL |
|---|-------------|--------|--------------|---------|----------|-----|
|       Sort.. Filter: | | | | | | |
| | COD_CLIENTE | NOME | SALDO_TOTAL | | | |
| 1 | 3 | C1 | 26500 | | | |
| 2 | 1 | A1 | 8000 | | | |

Linguagem SQL – DDL e DML

Alteração de informação na View

```
-- Exemplo de alterar dados numa view
```

```
create view v_cliente as  
select cod_cliente, nome, morada from cliente;
```

```
-- Podemos inserir dados nas views
```

```
insert into v_cliente values(4, 'D1', 'Rua X');
```

```
-- Também podemos eliminar ou alterar registos...
```

```
update v_cliente set morada = 'Rua Y'  
where cod_cliente = 4;
```

```
delete from v_cliente  
where cod_cliente = 4;
```

| COD_CLI... | NOME | MORADA |
|------------|------|--------|
| 1 | A1 | (null) |
| 2 | B1 | (null) |
| 3 | C1 | (null) |

| COD_CLIENTE | NOME | MORADA |
|-------------|------|--------|
| 1 | A1 | (null) |
| 2 | B1 | (null) |
| 3 | C1 | (null) |
| 4 | D1 | Rua X |

| COD_CLIENTE | N... | MORADA |
|-------------|------|--------|
| 1 | A1 | (null) |
| 2 | B1 | (null) |
| 3 | C1 | (null) |
| 4 | D1 | Rua Y |

| COD_CLI... | NOME | MORADA |
|------------|------|--------|
| 1 | A1 | (null) |
| 2 | B1 | (null) |
| 3 | C1 | (null) |

Linguagem SQL – DDL e DML



Alteração de informação com read only não é possível

```
113 create view v_cliente_read as
114     select cod_cliente, nome, morada from cliente
115     with read only;
116
117 -- Não Podemos inserir dados nas views read only
118 insert into v_cliente_read values(4, 'D1', 'Rua X');
119
120 -- Também não podemos eliminar ou alterar registos...
121 update v_cliente_read set morada = 'Rua Y' where cod_cliente = 3;
122 delete from v_cliente_read where cod_cliente = 3;
```

Script Output x

Task completed in 0,07 seconds

Error starting at line : 122 in command -

delete from v_cliente_read where cod_cliente = 3

Error at Command Line : 122 Column : 13

Error report -

SQL Error: ORA-42399: não é possível efetuar uma operação de DML numa visualização só de leitura
42399.0000 - "cannot perform a DML operation on a read-only view"

Linguagem SQL – DDL e DML

Alteração de informação – view com agregação

- Mas não podemos alterar informação através da view “Super Clientes”

```
-- Criação da view|  
  
create view "Super clientes" as  
select a.cod_cliente, a.nome, sum(b.saldo) saldo_total  
  from cliente a, conta b  
 where a.cod_cliente = b.cod_cliente  
       and a.cod_cliente in (select a.cod_cliente  
                             from cliente a, conta b  
                             where a.cod_cliente = b.cod_cliente  
                             and b.saldo > (select avg(saldo) from conta))  
 group by a.cod_cliente, a.nome;
```

- Porque tem informação de agregação

| COD_CL... | NOME | SALDO_... |
|-----------|------|-----------|
| 3 C1 | | 26500 |
| 1 A1 | | 20000 |

Linguagem SQL – DDL e DML



Alteração de informação – view com agregação

- Mas não podemos alterar informação através da view “Super Clientes”

```
124 -- *****
125 -- E se tiver várias tabelas e tiver informação agregada?
126
127 delete from "Super clientes" where cod_cliente = 3;
128 -- obtemos mensagem de erro na View
129
```

Script Output x

Task completed in 0,38 seconds

Error starting at line : 127 in command -
delete from "Super clientes" where cod_cliente = 3
Error at Command Line : 127 Column : 13
Error report -
SQL Error: ORA-01732: operação de manipulação de dados não é legal nesta visualização
01732. 00000 - "data manipulation operation not legal on this view"
*Cause:
*Action:

- Porque tem informação de agregação

Alteração de informação – view com join

- E se tiver várias tabelas e não tiver informação agregada?

```
create view v_clienteconta as
select a.cod_cliente, a.nome, a.morada, a.telefone, b.num_conta,
       b.cod_agencia, b.saldo
from cliente a, conta b
where a.cod_cliente = b.cod_cliente;
```

- Se tentarmos inserir um registo através da view...
 - o registo vai para a tabela cliente, conta, ou ambas?
 - e o nome, morada e telefone do cliente?
 - se por exemplo, para o cliente 1 não especificar o nome 'A1' vou gerar inconsistência?

Linguagem SQL – DDL e DML

Alteração de informação – view com join

```
127 create view v_clienteconta as
128 select a.cod_cliente, a.nome, a.morada, a.telefone, b.num_conta,
129        b.cod_agencia, b.saldo
130 from cliente a, conta b
131 where a.cod_cliente = b.cod_cliente;
132
133 insert into v_clienteconta
134 (cod_cliente, nome, morada, telefone, num_conta, cod_agencia, saldo)
135 values (1, 'A2', 'Morada X', '22-1234567', 7, 1, 0);
136
137 -- Iremos obter Msg para a View or function 'v_clienteconta'
```

Script Output x

Task completed in 0,07 seconds

SQL Error: ORA-01779: não é possível modificar uma coluna que corresponde a uma t
01779. 00000 - "cannot modify a column which maps to a non key-preserved table"
*Cause: An attempt was made to insert or update columns of a join view which
map to a non-key-preserved table.
*Action: Modify the underlying base tables directly.

Linguagem SQL – DDL e DML

Alteração de informação – view com join

--No entanto é possível alterar parte da informação.

```
update v_clienteconta
  set cod_agencia = 3
  where cod_cliente = 1 and num_conta = 2;
```

-- É possível alterar a tabela conta

```
update v_clienteconta
  set saldo = saldo + 10
  where cod_cliente = 1 and num_conta = 2;
```

Registo 2 (cliente 1 e conta 2)
passou a agência de 2 para 3
e o saldo passou de
8000 para 8010

| ⚡ COD_CLIENTE | ⚡ NOME | ⚡ MORADA | ⚡ TELEFONE | ⚡ NUM_CONTA | ⚡ COD_AGENCIA | ⚡ SALDO |
|---------------|--------|----------|------------|-------------|---------------|---------|
| 1 | A1 | (null) | (null) | 1 | 1 | 12000 |
| 1 | A1 | (null) | (null) | 2 | 2 | 8000 |
| 2 | B1 | (null) | (null) | 3 | 1 | 1000 |
| 3 | C1 | (null) | (null) | 4 | 3 | 16000 |
| 3 | C1 | (null) | (null) | 5 | 2 | 10000 |
| 3 | C1 | (null) | (null) | 6 | 1 | 500 |

| ⚡ COD_CLIENTE | ⚡ NOME | ⚡ MORADA | ⚡ TELEFONE | ⚡ NUM_CONTA | ⚡ COD_AGENCIA | ⚡ SALDO |
|---------------|--------|----------|------------|-------------|---------------|---------|
| 1 | A1 | (null) | (null) | 1 | 1 | 12000 |
| 1 | A1 | (null) | (null) | 2 | 3 | 8010 |
| 2 | B1 | (null) | (null) | 3 | 1 | 1000 |
| 3 | C1 | (null) | (null) | 4 | 3 | 16000 |
| 3 | C1 | (null) | (null) | 5 | 2 | 10000 |
| 3 | C1 | (null) | (null) | 6 | 1 | 500 |

Linguagem SQL – DDL e DML



Alteração de informação view com join

```
152 -- Mas não é possível alterar simultaneamente as tabelas cliente e conta.
153 update v_clienteconta
154     set morada = 'xx', saldo = saldo + 10
155     where cod_cliente = 1 and num_conta = 2;
156 -- obtemos Msg de erro
157 -- View or function 'v_clienteconta' is not updatable
```

Script Output x

Task completed in 0,23 seconds

```
update v_clienteconta
    set morada = 'xx', saldo = saldo + 10
    where cod_cliente = 1 and num_conta = 2
```

Error at Command Line : 154 Column : 8

Error report -

SQL Error: ORA-01779: não é possível modificar uma coluna que corresponde a uma t
01779. 00000 - "cannot modify a column which maps to a non key-preserved table"

*Cause: An attempt was made to insert or update columns of a join view which
map to a non-key-preserved table.

*Action: Modify the underlying base tables directly.

Linguagem SQL – DDL e DML

Alteração de informação – view com join

```
-- É possível apagar registos com as condições de join  
delete from v_clienteconta  
where cod_cliente = 3 and num_conta = 4;  
delete from v_clienteconta  
where cod_cliente = 3 and cod_agencia = 1;
```

| ⚡ COD_CLIENTE | ⚡ NOME | ⚡ MORADA | ⚡ TELEFONE | ⚡ NUM_CONTA | ⚡ COD_AGENCIA | ⚡ SALDO |
|---------------|--------|----------|------------|-------------|---------------|---------|
| 1 | A1 | (null) | (null) | 1 | 1 | 12000 |
| 1 | A1 | (null) | (null) | 2 | 3 | 8010 |
| 2 | B1 | (null) | (null) | 3 | 1 | 1000 |
| 3 | C1 | (null) | (null) | 4 | 3 | 16000 |
| 3 | C1 | (null) | (null) | 5 | 2 | 10000 |
| 3 | C1 | (null) | (null) | 6 | 1 | 500 |

| ⚡ COD_CLIENTE | ⚡ NOME | ⚡ MORADA | ⚡ TELEFONE | ⚡ NUM_CONTA | ⚡ COD_AGENCIA | ⚡ SALDO |
|---------------|--------|----------|------------|-------------|---------------|---------|
| 1 | A1 | (null) | (null) | 1 | 1 | 12000 |
| 1 | A1 | (null) | (null) | 2 | 3 | 8010 |
| 2 | B1 | (null) | (null) | 3 | 1 | 1000 |
| 3 | C1 | (null) | (null) | 5 | 2 | 10000 |

Concluindo

Recapitulando:

- É possível efetuar qualquer ação (INSERT, UPDATE, DELETE) sobre uma view que tem apenas uma tabela.
- Quando a view tem várias tabelas, é possível alterar registos se a alteração apenas se repercutir numa única tabela
- Quando a view tem várias tabelas, não é possível inserir registos.
- Todas estas restrições que existem quando a view tem várias tabelas desaparecem, quando usarmos INSTEAD OF TRIGGERS

Linguagem SQL – DDL e DML



Funções do Oracle

★ Funções DATE

- Armazena a data num formato numérico interno : century, year, month, day;
- Por omissão o formato é : DD-MON- YY;
- Sysdate é uma função que retorna a data e hora;

| Function | Result |
|-----------------------|---|
| MONTHS_BETWEEN | Number of months between two dates |
| ADD_MONTHS | Add calendar months to date |
| NEXT_DAY | Next day of the date specified |
| LAST_DAY | Last day of the month |
| ROUND | Round date |
| TRUNC | Truncate date |

Linguagem SQL – DDL e DML

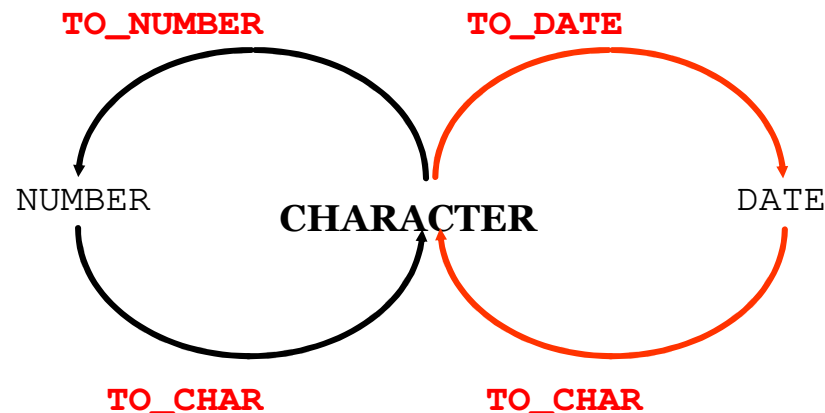
Funções do Oracle

★ Funções de conversão

Conversão implícita de tipo de dados



| From | To |
|------------------|----------|
| VARCHAR2 or CHAR | NUMBER |
| VARCHAR2 or CHAR | DATE |
| NUMBER | VARCHAR2 |
| DATE | VARCHAR2 |
| VARCHAR2 or CHAR | NUMBER |
| VARCHAR2 or CHAR | DATE |



Conversão explícita de tipo de dados

Linguagem SQL – DDL e DML

Funções do Oracle

★ **TO_CHAR** (date, 'format_model')

- Converte uma data para uma string usando o formato especificado.

```
SELECT TO_CHAR(SYSDATE, 'YYYY-MM-DD') FROM dual;
```

Format_model



| Element | Result |
|---------|--|
| YYYY | Full year in numbers |
| YEAR | Year spelled out (in English) |
| MM | Two-digit value for month |
| MONTH | Full name of the month |
| MON | Three-letter abbreviation of the month |
| DY | Three-letter abbreviation of the day of the week |
| DAY | Full name of the day of the week |
| DD | Numeric day of the month |

Linguagem SQL – DDL e DML



Funções do Oracle

★ **TO_DATE** (char [, 'format_model'])

- Converte uma string para um formato de data.

```
SELECT TO_DATE('2015/05/15 8:30:25', 'YYYY/MM/DD HH:MI:SS')  
FROM dual;
```

★ **TO_EXTRACT**

- Permite extrair um valor a partir de uma data ou intervalo.

```
EXTRACT (  
  { YEAR | MONTH | DAY | HOUR | MINUTE | SECOND }  
  | { TIMEZONE_HOUR | TIMEZONE_MINUTE }  
  | { TIMEZONE_REGION | TIMEZONE_ABBR }  
FROM { date_value | interval_value } )
```

Linguagem SQL – DDL e DML

Funções do Oracle : exemplos

```
-- Exemplos Extract  
SELECT EXTRACT(YEAR FROM DATE '2003-08-22') FROM dual;  
-- Resultado: 2003
```

Query Result x

SQL | All Rows Fetched: 1 in 0,02 seconds

| | EXTRACT(YEARFROMDATE'2003-08-22') |
|---|-----------------------------------|
| 1 | 2003 |

```
196 SELECT EXTRACT(DAY FROM DATE '2011-01-12') FROM dual;  
197 -- Resultado: 12  
198
```

Query Result x


SQL | All Rows Fetched: 1 in 0,019 seconds


| | EXTRACT(DAYFROMDATE'2011-01-12') |
|---|----------------------------------|
| 1 | 12 |

Linguagem SQL – DDL e DML

Funções do Oracle - exemplos


```
199 SELECT EXTRACT(HOUR FROM SYSDATE) FROM dual;  
200 -- ERROR at line 1:  
201 -- ORA-30076: invalid extract field for extract source
```





 Query Result x

    SQL | Executing: SELECT EXTRACT(HOUR FROM SYSDATE) FROM dual in 0 seconds

ORA-30076: campo de extração para origem de extração inválido
30076. 00000 - "invalid extract field for extract source"
*Cause: The extract source does not contain the specified extract field.
*Action:
Error at Line: 199 Column: 26

```
203 SELECT EXTRACT(HOUR FROM TIMESTAMP '2012-01-12 10:11:00') FROM dual;
```

 Query Result x

    SQL | All Rows Fetched: 1 in 0,022 seconds

| | EXTRACT(HOUR FROM TIMESTAMP '2012-01-12 10:11:00') |
|---|--|
| 1 | 10 |

Linguagem SQL – DDL e DML

```
171 | --*****
172 | -- Funções de Oracle
173 | --*****
174 | -- What is a DUAL Table in Oracle?
175 | -- This is a single row and single column dummy table provided by oracle.
176 | -- This is used to perform mathematical calculations without using a table.
177 | Select * from DUAL;
178 |
```

| Query Result x | |
|--|---|
| SQL All Rows Fetched: 1 in 0,401 seconds | |
| DUMMY | |
| 1 | X |

```
179 | Select 777 * 888 from Dual;
180 |
```

| Query Result x | |
|---|--------|
| SQL All Rows Fetched: 1 in 0,69 seconds | |
| 777*888 | |
| 1 | 689976 |

Linguagem SQL – DDL e DML

```
231 -- *****
232 -- ADD / LAST
233 -- *****
234 Select add_months(sysdate, -1) as prev_month , sysdate, add_months (sysdate, 1)
235 as next_month from dual;
236 -- mês anterior - mês atual - próximo mês para o mesmo dia
```

Query Result x

SQL | All Rows Fetched: 1 in 2,631 seconds

| | PREV_MONTH | SYSDATE | NEXT_MONTH |
|---|------------|----------|------------|
| 1 | 11.09.18 | 11.10.18 | 11.11.18 |

```
238 select sysdate, last_day(sysdate) as last_day_curr_month,
239 last_day(sysdate) + 1 as first_day_next_month from dual;
240 -- último dia do mês anterior e primeiro dia do mês seguinte
```

Query Result x

SQL | All Rows Fetched: 1 in 1,237 seconds

| | SYSDATE | LAST_DAY_CURR_MONTH | FIRST_DAY_NEXT_MONTH |
|---|----------|---------------------|----------------------|
| 1 | 11.10.18 | 31.10.18 | 01.11.18 |

```
242 select last_day(sysdate) - sysdate as days_left from dual;
243 -- dias em falta para terminar o mês
```

Query Result x

SQL | All Rows Fetched: 1 in 0,915 seconds

| | DAYS_LEFT |
|---|-----------|
| 1 | 20 |

Linguagem SQL – DDL e DML



--Character String:

--Function 1: UPPER Purpose : Returns the string in uppercase

--Syntax : UPPER('str')

--Example :

SELECT UPPER('karuvachi') **from** Dual;

--Output:KARUVACHI

--Function 2: lower Purpose : Returns the string in lowercase

--Syntax : lower('str')

--Example :

SELECT LOWER('KaRuVaChi') **FROM** DUAL;

--Output:karuvachi

--Function 3: Initcap Purpose : Returns the string with first letter in uppercase
--and rest of the letters in lowercase

--Syntax : Initcap('str')

--Example :

SELECT Initcap('KaRuVaChi') **FROM** DUAL;

--Output:KaruVachi

--Function 4: Concat Purpose : Concatenate two strings

--Syntax : concat('str1?','str2')

--Example :

SELECT CONCAT('Karu','Nand') **FROM** DUAL;

--Output:KaruNand

Linguagem SQL – DDL e DML



```
--*****
```

```
--ABS: Absolute value of the number
```

```
SELECT ABS(12) FROM DUAL;
```

```
--ABS(12)
```

```
--output: 12
```

```
--CEIL: Integer value that is Greater than or equal to the number
```

```
SELECT CEIL(48.99) FROM DUAL;
```

```
--CEIL(48.99)
```

```
-----
```

```
--output: 49
```

```
--FLOOR: Integer value that is Less than or equal to the number
```

```
SELECT FLOOR(49.99) FROM DUAL;
```

```
--FLOOR(49.99)
```

```
-----
```

```
--output: 49
```

```
--ROUND: Rounded off value of the number 'x' up to the number 'y' decimal places
```

```
SELECT ROUND(49.11321,2) FROM DUAL;
```

```
--ROUND(49.11321,2)
```

```
-----
```

```
--output: 49.11
```