10

SQLs Avançados

Objetivos

- Consulta SELECT – WITH
- GROUP BY (revisão)
- GROUP BY HAVING
- GROUP BY ROLLUP
- GROUP BY CUBE
- GROUP BY GROUPING SETS
- Views
- Materialized Views
- Índices

Consulta SELECT – WITH

```
WITH alias1 as ( subquery ),
     ... ,
     aliasn as ( subquery )
SELECT ...
FROM   ...
WHERE  ... ;
```

Consulta SELECT – WITH

```
with d1 as (
        select  avg(salary) as media_salario
        from    hr.employees
        where   department_id = 80
        ),
     d2 as (
        select  avg(salary) as media_salario
        from    hr.employees
        where   department_id = 100
        and     salary > (select media_salario from d1)
        )
select  *
from    hr.employees
where   salary > (select media_salario from d2)
;
```

Funções de AGRUPAMENTO (revisão)

Agrupam várias linhas:

- COUNT
- SUM
- AVG
- MIN
- MAX

```
select count(1) ,  sum(salary),  avg(salary)
from hr.employees;
```

podem ser utilizadas em conjunto com a clausula GROUP BY

```
select   department_id,
         count(1) as qtd_funcs,
         avg(salary) as media_salarios
from     hr.employees
group    by department_id;
```

GROUP BY .... HAVING

```
select   department_id,
         count(1) as qtd_funcs,
         avg(salary) as media_salarios
from     hr.employees
group    by department_id;
```

DEPARTMENT_ID	QTD_FUNCS	
50	45	3475.555555
40	1	6500
110	2	10154

```
select   department_id,
         count(1) as qtd_funcs,
         avg(salary) as media_salarios
from     hr.employees
group    by department_id
having   count(1) > 10;
```

DEPARTMENT_ID	QTD_FUNCS	
50	45	3475.555555
80	34	8973.529411

Cenário

Tabela Vendas e Agrupamentos (Fato x Dimensões)


```
create table vendas (
        PRODUTO          varchar2(100)  NOT NULL,
        UF               char(2)        NOT NULL,
        MEIO_PAGAMENTO   varchar2(10)   NOT NULL,
        CANAL_VENDAS     varchar2(15)   NOT NULL,
        ANO              number(4)      NOT NULL,
        MES              number(2)      NOT NULL,
        QTD_VENDIDA      number(6)      NOT NULL,
        VALOR_VENDA      number(16,2)   NOT NULL
);
```

Dimensões

```
select   SUM(VALOR_VENDA) as soma_valor_vendas,
         SUM(QTD_VENDIDA) as soma_qtd
from     vendas f;
```

```
select   PRODUTO,
         SUM(VALOR_VENDA) as soma_valor_vendas,
         SUM(QTD_VENDIDA) as soma_qtd
from     vendas f
group    by PRODUTO;
```

```
select   PRODUTO,
         CANAL_VENDAS,
         SUM(VALOR_VENDA) as soma_valor_vendas,
         SUM(QTD_VENDIDA) as soma_qtd
from     vendas f
group    by PRODUTO, CANAL_VENDAS;
```

```
select   PRODUTO,
         CANAL_VENDAS,
         MEIO_PAGAMENTO,
         SUM(VALOR_VENDA) as soma_valor_vendas,
         SUM(QTD_VENDIDA) as soma_qtd
from     vendas f
group    by PRODUTO, CANAL_VENDAS, MEIO_PAGAMENTO;
```

ROLLUP

- Cálculo de Múltiplos Níveis
- Dimensões Hierarquicas
- Subtotatis dos níveis (sintéticos → até os mais analíticos)
- Total Geral

ROLLUP

- Antes…

  pelo menos 4 consultas →

Dimensões:

```
select   SUM(VALOR_VENDA) as soma_valor_vendas,
         SUM(QTD_VENDIDA) as soma_qtd
from     vendas f;

                              select   PRODUTO,
                                       SUM(VALOR_VENDA) as soma_valor_vendas,
                                       SUM(QTD_VENDIDA) as soma_qtd
                              from     vendas f
select   PRODUTO,                                    TO;
         CANAL_VENDAS,
         SUM(VALOR_VENDA) as soma_valor_vendas,
         SUM(QTD_VENDIDA) as soma_qtd
from     vendas f
group    by PRODUTO, CANAL_VENDAS;

                                    select   PRODUTO,
                                             CANAL_VENDAS,
                                             MEIO_PAGAMENTO,
                                             SUM(VALOR_VENDA) as soma_valor_vendas,
                                             SUM(QTD_VENDIDA) as soma_qtd
                                    from     vendas f
                                    group    by PRODUTO, CANAL_VENDAS, MEIO_PAGAMENTO;
```

- Agora, 1 consulta:

```
select   PRODUTO,
         CANAL_VENDAS,
         MEIO_PAGAMENTO,
         SUM(VALOR_VENDA) as soma_valor_vendas,
         SUM(QTD_VENDIDA) as soma_qtd
from     vendas f
group    by ROLLUP( PRODUTO, CANAL_VENDAS, MEIO_PAGAMENTO);
```

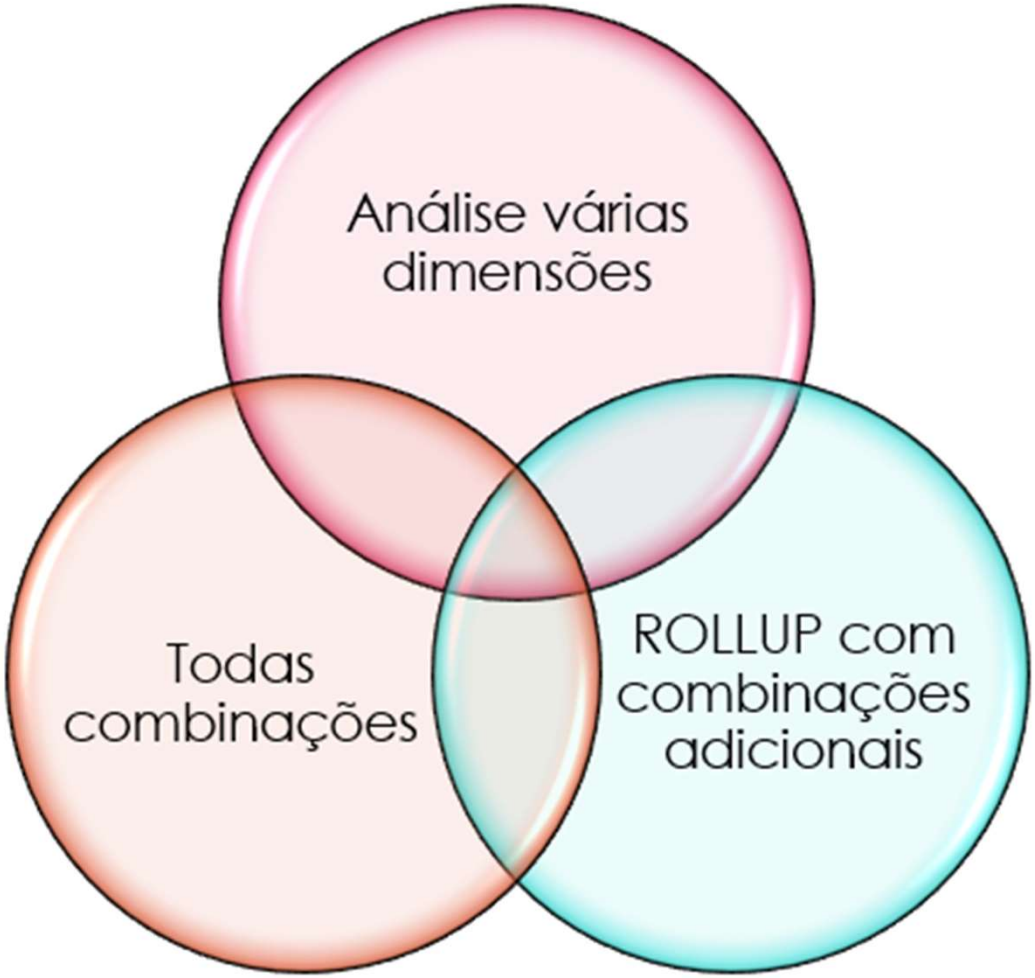
ROLLUP

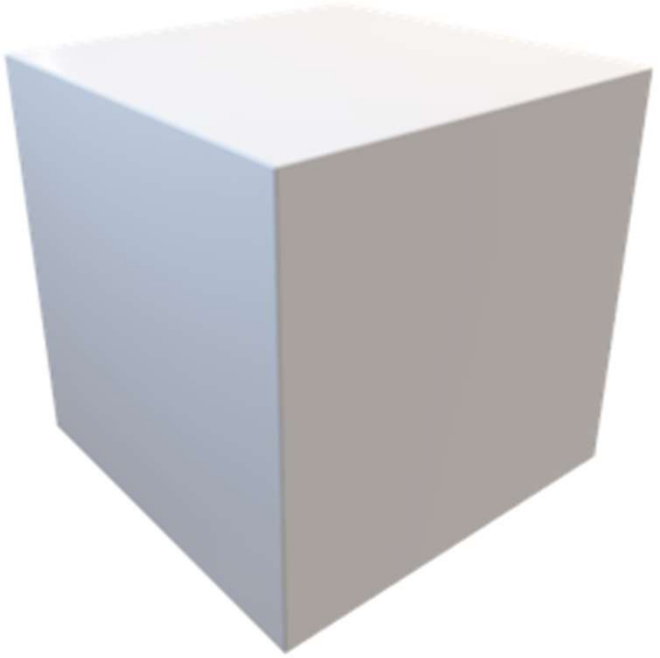
```
select   PRODUTO,
         CANAL_VENDAS,
         MEIO_PAGAMENTO,
         SUM(VALOR_VENDA) as soma_valor_vendas,
         SUM(QTD_VENDIDA) as soma_qtd
from     vendas f
group    by ROLLUP( PRODUTO, CANAL_VENDAS, MEIO_PAGAMENTO);
```

PRODUTO	CANAL_VENDAS	MEIO_PAGAM	SOMA_VALOR_VENDAS	SOMA_QTD
TV	ELO7	PIX	93266.81	349
TV	ELO7	BOLETO	99703.55	546
...				
...				
LIQUIDIFICADOR	MERCADOLIVRE	CREDITO	76428.97	437
LIQUIDIFICADOR	MERCADOLIVRE	DINHEIRO	91405.4	531
LIQUIDIFICADOR	MERCADOLIVRE		467435.28	2693
LIQUIDIFICADOR			2282093.5	12445
			22541983.6	119654

311 rows selected.

CUBE


Análise várias dimensões
Todas combinações
ROLLUP com combinações adicionais



CUBE

```
select   PRODUTO,
         CANAL_VENDAS,
         MEIO_PAGAMENTO,
         SUM(VALOR_VENDA) as soma_valor_vendas,
         SUM(QTD_VENDIDA) as soma_qtd
from     vendas f
group    by CUBE( PRODUTO, CANAL_VENDAS, MEIO_PAGAMENTO);
```

PRODUTO	CANAL_VENDAS	MEIO_PAGAM	SOMA_VALOR_VENDAS	SOMA_QTD
			22541983.6	119654
		PIX	4544871.92	23679
		BOLETO	4445754.56	23749
...				
	ELO7		4519689.78	23225
	ELO7	PIX	906723.21	4416
...				
TV		PIX	468715.42	2280
TV		BOLETO	449347.75	2523
...				
LIQUIDIFICADOR	MERCADOLIVRE	CREDITO	76428.97	437
LIQUIDIFICADOR	MERCADOLIVRE	DINHEIRO	91405.4	531

396 rows selected.

GROUPING SETS

GROUPING SETS

```
select  MES,
        CANAL_VENDAS,
        PRODUTO,
        SUM(VALOR_VENDA) as soma_valor_vendas,
        SUM(QTD_VENDIDA) as soma_qtd
from    vendas f
group   by GROUPING SETS( (MES, PRODUTO),
                          (MES, CANAL_VENDAS),
                          MES
                        )
order   by MES, PRODUTO nulls first, CANAL_VENDAS nulls first;
```

MES	CANAL_VENDAS	PRODUTO	SOMA_VALOR_VENDAS	SOMA_QTD
1			1897315.8	10275
1	AMAZON		387604.32	1836
1	ELO7		336148.8	1927
1	MERCADOLIVRE		400432.66	2130
1	PRESENCIAL		387624.57	2171
1	SITE		385505.45	2211
1		BICICLETA	198010.74	1120
1		CELULAR	171559.57	892
...				
2			1849561.05	9725
2	AMAZON		351547.87	2014
2	ELO7		344901.17	1834
2	MERCADOLIVRE		374228.41	1959
...				

192 rows selected.

VIEW

VISÃO

- “Um **apelido** para um consulta SELECT”
- Vantagens:
  - Simplificar repetição de consultas complexas
  - Controlar o acesso a DADOS específicos
  - Ocultar/mascarar dados
  - Produtividade e tempo

- `CREATE VIEW <nome_view> AS`
- `SELECT ... ;`

VIEW

- Consulta SELECT:

```
select   emp.first_name, emp.last_name, emp.salary,
         dep.department_name,
         loc.city
from     hr.employees emp
inner    join hr.departments dep
                  on (dep.department_id = emp.department_id)
inner    join hr.locations loc
                  on (loc.location_id = dep.location_id)
and      emp.salary > 10000;
```

VIEW

- CREATE VIEW …

```
CREATE   VIEW emp_dep_loc AS
select   emp.first_name, emp.last_name, emp.salary,
         dep.department_name,
         loc.city
from     hr.employees emp
inner    join hr.departments dep
                  on (dep.department_id = emp.department_id)
inner    join hr.locations loc
                  on (loc.location_id = dep.location_id)
and      emp.salary > 10000;
```

```
View created.
```

VIEW

```
1  desc emp_dep_loc ;
```

VIEW EMP_DEP_LOC

Column	Null?	Type
FIRST_NAME	-	VARCHAR2(20)
LAST_NAME	NOT NULL	VARCHAR2(25)
SALARY	-	NUMBER(8,2)
DEPARTMENT_NAME	NOT NULL	VARCHAR2(30)
CITY	NOT NULL	VARCHAR2(30)

```
select   *
from     emp_dep_loc;
```

```
select   first_name, last_name, salary,
         department_name, city
from     emp_dep_loc;
```

```
1  drop view   emp_dep_loc;
```

View dropped.

MATERIALIZED VIEW

VISÃO MATERIALIZADA

- "Um apelido para um consulta SELECT"
- Vantagens:
  - Simplificar repetição de consultas complexas
  - Controlar o acesso a DADOS específicos
  - Ocultar/mascarar dados
  - Produtividade e tempo
  - **Pré-executa e armazena o resultado da query**

- `CREATE MATERIALIZED VIEW <nome_view> AS`
- `SELECT ... ;`

MATERIALIZED VIEW

```
CREATE   MATERIALIZED VIEW emp_dep_loc AS
select   emp.first_name, emp.last_name, emp.salary,
         dep.department_name,
         loc.city
from     hr.employees emp
inner    join hr.departments dep
                  on (dep.department_id = emp.department_id)
inner    join hr.locations loc
                  on (loc.location_id = dep.location_id)
and      emp.salary > 10000;
```

```
Statement processed.
```

MATERIALIZED VIEW

- Dados já estão “guardados” em uma tabela
- Retorno é imediato, mesmo que a consulta fosse demorada

```
1  select * from emp_dep_loc;
```

FIRST_NAME	LAST_NAME	SALARY	DEPARTMENT_NAME	CITY
Den	Raphaely	11000	Purchasing	Seattle
Steven	King	24000	Executive	Seattle
Neena	Kochhar	17000	Executive	Seattle
Lex	De Haan	17000	Executive	Seattle

MATERIALIZED VIEW

- Existem:
  - a MView (select)
  - a Tabela (dados guardados)

```
1  select   object_name, object_type, created
2  from     user_objects
3  where    object_name = 'EMP_DEP_LOC';
```

OBJECT_NAME	OBJECT_TYPE	CREATED
EMP_DEP_LOC	TABLE	20/11/2024 19:45:37
EMP_DEP_LOC	MATERIALIZED VIEW	20/11/2024 19:45:37

MATERIALIZED VIEW

- Problema: dados são estáticos / atualizar os dados
- Procedure: DBMS_MVIEW.REFRESH
  - Executar manualmente

```
1  exec DBMS_MVIEW.REFRESH( 'EMP_DEP_LOC' );
```

```
Statement processed.
```

```
1  begin
2      DBMS_MVIEW.REFRESH( 'EMP_DEP_LOC' );
3  end;
4  /
```

```
Statement processed.
```

MATERIALIZED VIEW

- Problema: dados são estáticos / atualizar dos dados
- Procedure: DBMS_MVIEW.REFRESH
  - Agendar no scheduler

```
BEGIN
sys.dbms_scheduler.create_job(
        job_name          => 'HR.EMP_DEP_LOC_REFRESH',
        job_type          => 'PLSQL_BLOCK',
        job_action        => '
                    begin
                       DBMS_MVIEW.REFRESH( 'HR.EMP_DEP_LOC' );
                    end;
                    ',
        repeat_interval => 'FREQ=DAILY;BYHOUR=02;BYMINUTE=30;',
        enabled           => TRUE
        );
END;
/
```

ÍNDICE

- Estrutura auxiliar
- Salva os dados em uma forma ordenada
- Utiliza formato de Árvore
- Objetivo: acessos mais rápido

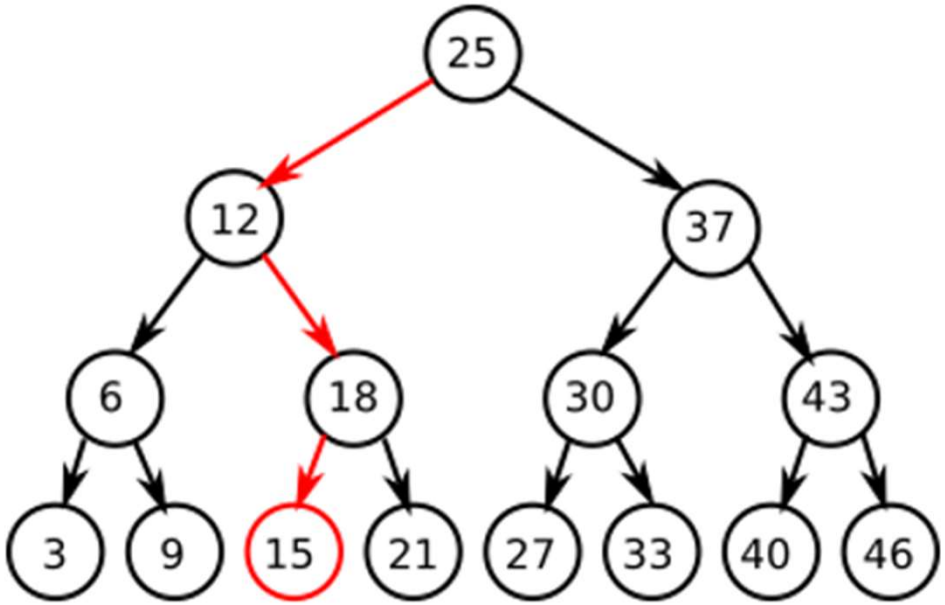
- Como é outro objeto (outra estrutura)
  - Ocupa um pouco mais de espaço
  - Torna DMLs um pouco mais lentos

ÍNDICE - ilustração

- Tabela:

18	12	33	6	27	40	30	25	46	9	3	21	**15**	37	43

- Índice:



ÍNDICE

- CREATE [UNIQUE] INDEX idx_name
- ON tabela (col1, col2);

- Tradicionais, UNIQUE, multi-columns:

```
create index HR.EMP_DEPARTMENT_IX    on HR.EMPLOYEES(DEPARTMENT_ID);
create index HR.EMP_JOB_IX           on HR.EMPLOYEES(JOB_ID);
create index HR.EMP_MANAGER_IX       on HR.EMPLOYEES(MANAGER_ID );

create index HR.EMP_NAME_IX          on HR.EMPLOYEES(LAST_NAME, FIRST_NAME);

create UNIQUE index HR.EMP_EMAIL_UK  on HR.EMPLOYEES(EMAIL);
create UNIQUE index HR.EMP_EMP_ID_PK on HR.EMPLOYEES(EMPLOYEE_ID);
```

Resumo e Dúvidas

- Dúvidas ou comentários ... ?