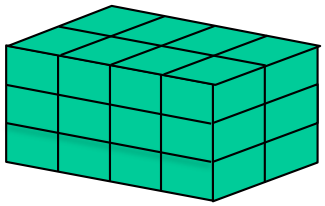
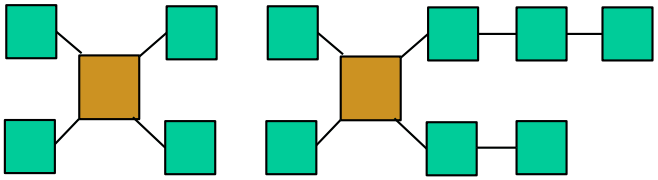
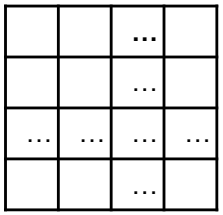
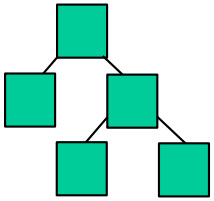
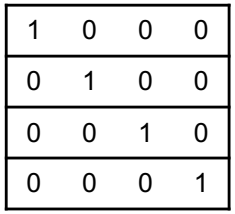
SQL

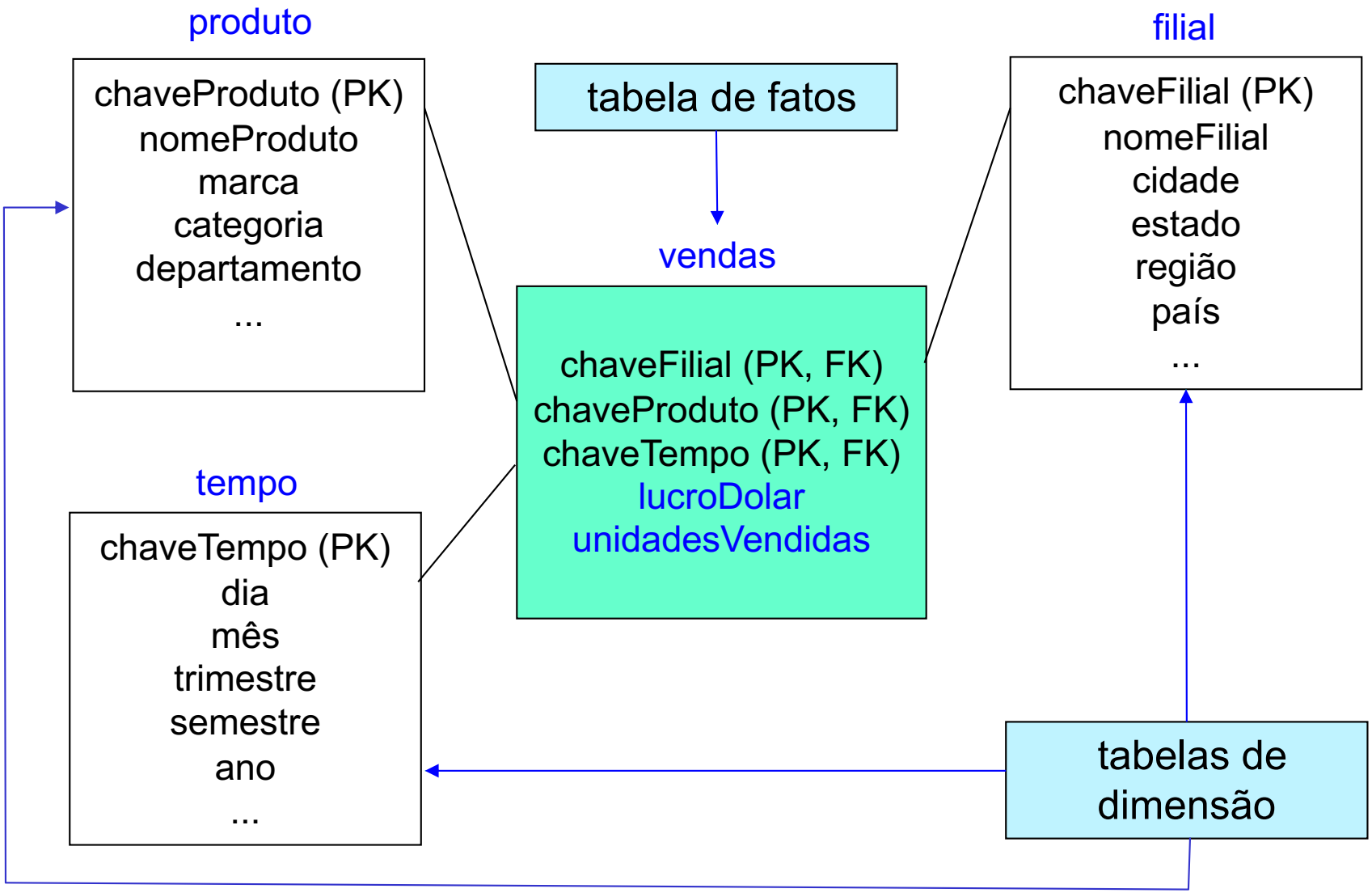
Processamento Analítico de Dados

Profa. Dra. Cristina Dutra de Aguiar Ciferri

Arquitetura de 3 Camadas

	esquema	operações
conceitual	 metáfora do cubo de dados	Cube Álgebra
lógico	 *esquemas estrela e floco de neve* ROLAP  *estruturas matriciais* MOLAP	SQL MDX ...
físico	 *índices: árvores*  *índices bitmap* ROLAP *estruturas e algoritmos proprietários* MOLAP	processamento e otimização de consultas

Esquema Estrela (SQL)



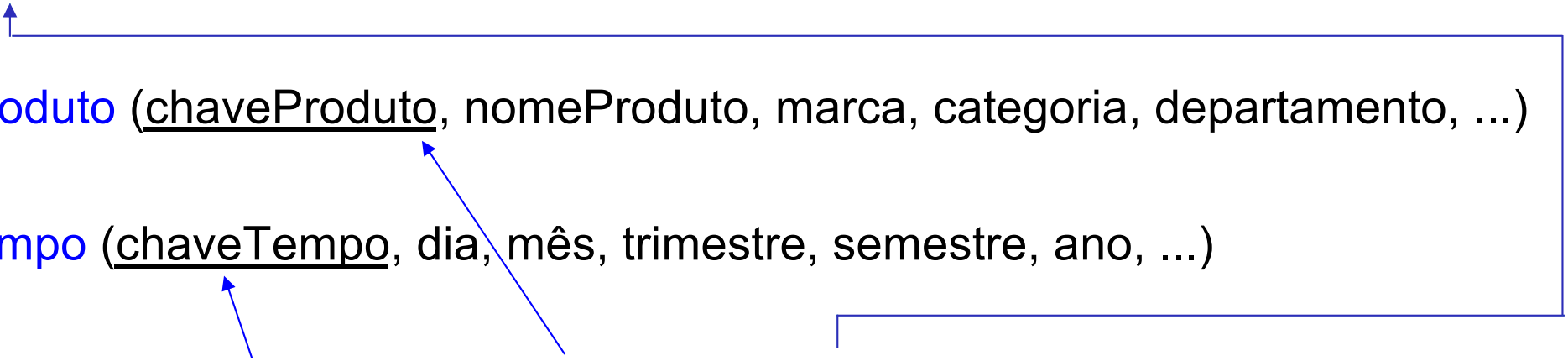
Esquema Estrela

filial (chaveFilial, nomeFilial, cidade, estado, região, país, ...)

produto (chaveProduto, nomeProduto, marca, categoria, departamento, ...)

tempo (chaveTempo, dia, mês, trimestre, semestre, ano, ...)

vendas (chaveTempo, chaveProduto, chaveFilial,
lucroDolar, unidadesVendidas)



Esquema Estrela: Criação

```
CREATE TABLE filial (
    chaveFilial NUMBER(4) NOT NULL,
    nomeFilial VARCHAR2(15) UNIQUE,
    cidade VARCHAR2(15),
    estado VARCHAR2(15),
    regiaoVARCHAR2(15),
    pais VARCHAR2(15),
    PRIMARY KEY (chaveFilial)
);

CREATE TABLE produto (
    chaveProduto NUMBER(4) NOT NULL,
    nomeProduto VARCHAR2(15) UNIQUE,
    marca VARCHAR2(15),
    categoria VARCHAR2(15),
    departamento VARCHAR2(15),
    PRIMARY KEY (chaveProduto)
);
```

```
CREATE TABLE tempo (
    chaveTempo NUMBER(4) NOT NULL,
    dia NUMBER(2),
    mes NUMBER(2),
    trimestre NUMBER(1),
    semestre NUMBER(1),
    ano NUMBER(4),
    PRIMARY KEY (chaveTempo)
);
```

tabelas de dimensão

Esquema Estrela: Criação

```
CREATE TABLE vendas (
   chaveFilial NUMBER(4) NOT NULL,
    chaveProduto NUMBER(4) NOT NULL,
    chaveTempo NUMBER(4) NOT NULL,
    lucroDolar NUMBER(10,2),
    unidadesVendidas NUMBER(4),
    PRIMARY KEY (chaveFilial, chaveProduto, chaveTempo),
    FOREIGN KEY (chaveFilial)
        REFERENCES filial(chaveFilial),
    FOREIGN KEY (chaveProduto)
        REFERENCES produto(chaveProduto),
    FOREIGN KEY (chaveTempo)
        REFERENCES tempo(chaveTempo)
);
```

tabela de fatos

Junção Estrela

- Necessária devido à organização dos dados segundo os esquemas estrela, floco de neve, estrela-floco
- Descrição
  - consiste em acessar todas as tabelas de fatos e de dimensão envolvidas em uma consulta OLAP e realizar todas as junções e agregações necessárias, além de resolver as condições de seleção aplicadas sobre os predicados convencionais

Pivot

reorienta a visão multidimensional dos dados, oferecendo diferentes perspectivas dos mesmos dados

- Liste o lucro em dólar e as unidades vendidas, por tempo, por produto, por filial

```
SELECT chaveTempo, chaveProduto, chaveFilial,
            lucroDolar, unidadesVendidas
FROM vendas
ORDER BY chaveTempo, chaveProduto, chaveFilial
```

semântica da resposta !

Mais Semântica

- Liste o lucro em dólar e as unidades vendidas, por tempo, por produto, por filial

```
SELECT dia || ' ' || mes || ' ' || ano AS data,
            nomeProduto, nomeFilial,
            lucroDolar, unidadesVendidas
FROM vendas V, tempo T, produto P, filial F
WHERE V.chaveTempo = T.chaveTempo
  AND V.chaveProduto = P.chaveProduto
  AND V.chaveFilial = F.chaveFilial
ORDER BY V.chaveTempo, V.chaveProduto, V.chaveFilial;
```

junção estrela

Slice

restringe os dados sendo analisados a um subconjunto destes dados, considerando um valor fixo

- Liste o lucro em dólar e as unidades vendidas, para chaveTempo = 1.

```
SELECT chaveFilial, chaveProduto, chaveTempo,
            lucroDolar, unidadesVendidas
FROM vendas
WHERE chaveTempo = 1;
```

semântica da resposta !

Dice

restringe os dados sendo analisados a um subconjunto destes dados, considerando uma faixa de valores

- Liste o lucro em dólar e as unidades vendidas, para chaveTempo entre 1 e 2.

```
SELECT chaveFilial, chaveProduto, chaveTempo,
            lucroDolar, unidadesVendidas
FROM vendas
WHERE chaveTempo BETWEEN 1 AND 2;
```

semântica da resposta !

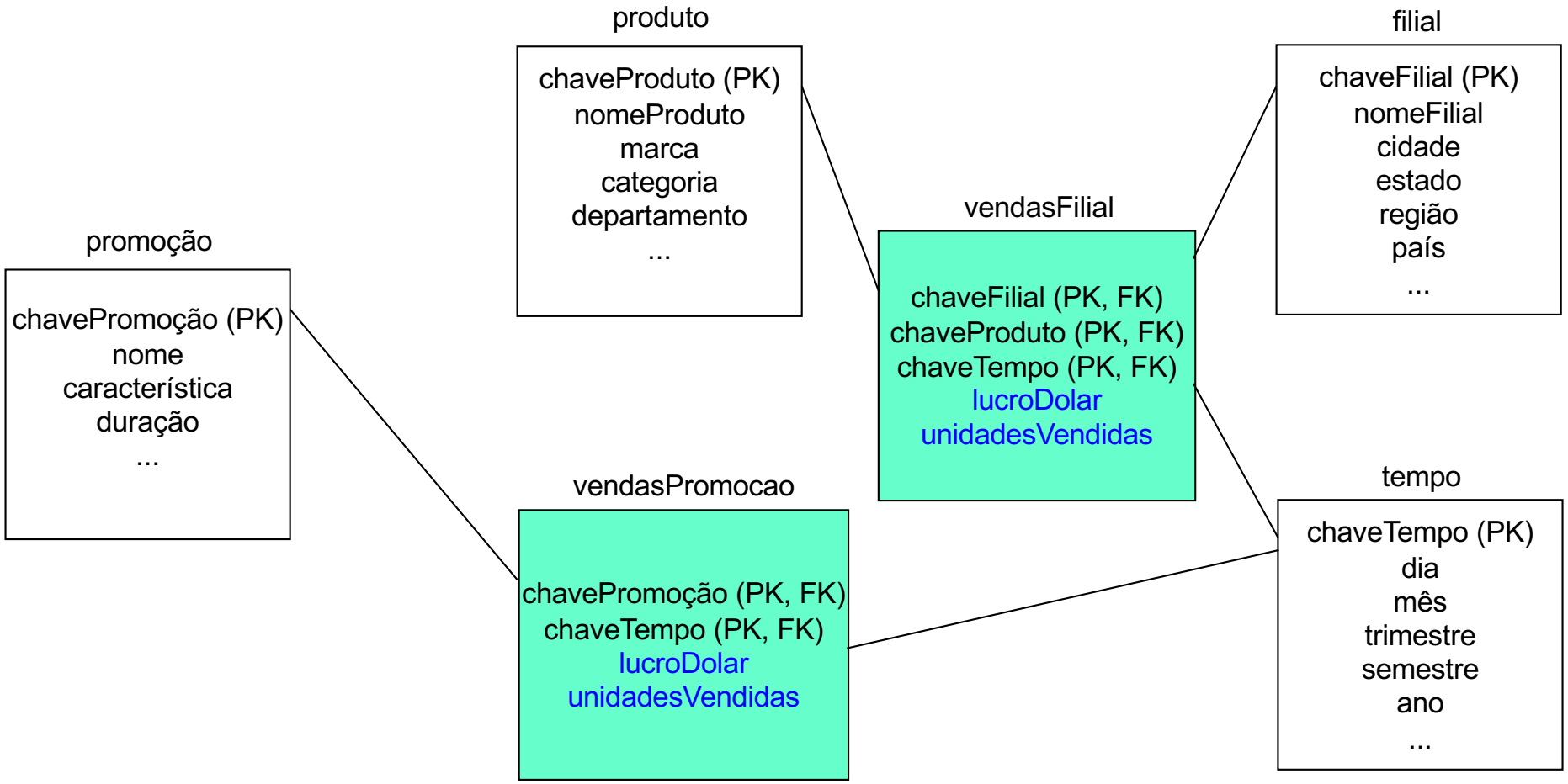
Roll-up

analisa os dados em níveis de agregação progressivamente menos detalhados, ou de maior granularidade

- Liste o lucro em dólar e as unidades vendidas, por ano, por estado, por produto.

```
SELECT ano, estado, nomeProduto, SUM(lucroDolar) AS "Lucro Anual",
            SUM(unidadesVendidas) AS "Unidades Vendidas"
FROM vendas V, filial F, tempo T, produto P
WHERE V.chaveFilial = F.chaveFilial
  AND V.chaveTempo = T.chaveTempo
  AND V.chaveProduto = P.chaveProduto
GROUP BY ano, estado, nomeProduto
ORDER BY ano, estado, nomeProduto;
```

Constelação de Fatos (SQL)



Demais Tabelas: Criação

```
CREATE TABLE promocao (
    chavePromocao NUMBER(4) NOT NULL,
    nomePromocao VARCHAR2(15) UNIQUE,
    caracteristica VARCHAR2(30),
    duracao NUMBER(2),
    PRIMARY KEY (chavePromocao)
);
```

```
CREATE TABLE vendasPromocao (
    chavePromocao NUMBER(4) NOT NULL,
    chaveTempo NUMBER(4) NOT NULL,
    unidadesVendidas NUMBER(4),
    PRIMARY KEY (chavePromocao, chaveTempo),
    FOREIGN KEY (chavePromocao) REFERENCES promocao(chavePromocao),
    FOREIGN KEY (chaveTempo) REFERENCES tempo(chaveTempo)
);
```

Drill-across

compara medidas numéricas distintas que são relacionadas entre si através de pelo menos uma dimensão em comum

- Liste as unidades vendidas por ano, considerando as tabelas de dimensão vendasFilial e vendasPromocao da constelação de fatos.

Drill-across

```
SELECT parte1.ano, vendaFilial, vendaPromocao
FROM
   (SELECT ano, SUM(unidadesVendidas) AS vendaFilial
    FROM vendas VF, tempo TF
    WHERE VF.chaveTempo = TF.chaveTempo
    GROUP BY ano) parte1,
   (SELECT ano, SUM(unidadesVendidas) AS vendaPromocao
    FROM vendasPromocao VP, tempo TP
    WHERE VP.chaveTempo = TP.chaveTempo
    GROUP BY ano) parte2
WHERE parte1.ano = parte2.ano
ORDER BY parte1.ano;
```

Novos Operadores SQL

- ROLLUP e CUBE
  - utilizadas para a construção de vários níveis de agregação
- GROUPING SETS
  - função mais genérica utilizada para a construção de vários níveis de agregação
- FUNÇÕES DE JANELAMENTO
  - permitem a comparação entre medidas numéricas agregadas e medidas numéricas não agregadas

ROLLUP

- Funcionalidade
  - criação de subtotais que envolvem desde o nível mais detalhado até um total geral, seguindo uma lista de agrupamento especificada na cláusula ROLLUP
- Argumento
  - lista ordenada de agrupamento de colunas

ROLLUP

- Processamento
  - cálculo dos valores agregados padrões especificados na cláusula GROUP BY
  - criação, de forma progressiva, de subtotais de nível mais alto, da esquerda para a direita na lista de agrupamento de colunas
  - criação de um total geral
- Resultado
  - n+1 níveis, sendo n o número de agrupamento de colunas

CUBE

- Funcionalidade
  - criação de subtotais para todas as combinações da lista de agrupamento especificada na cláusula CUBE
  - criação do total geral

- Resultado
  - 2^n níveis, sendo n o número de agrupamento de colunas

CUBE

- Processamento
  - cálculo dos valores agregados padrões especificados na cláusula GROUP BY
  - criação, de forma progressiva, de subtotais de nível mais alto, para todas as combinações de dimensões na lista de agrupamento de colunas
  - criação de um total geral

Exemplos

```
SELECT chaveFilial, chaveProduto, chaveTempo,
        SUM(lucroDolar) AS "Lucro Anual",
        SUM(unidadesVendidas) AS "Unidades Vendidas"
FROM vendas
GROUP BY ROLLUP (chaveFilial, chaveProduto, chaveTempo);
```

```
SELECT chaveFilial, chaveProduto, chaveTempo,
        SUM(lucroDolar) AS "Lucro Anual",
        SUM(unidadesVendidas) AS "Unidades Vendidas"
FROM vendas
GROUP BY CUBE (chaveFilial, chaveProduto, chaveTempo);
```

GROUPING SETS

- Características
  - permite definir qualquer conjunto (ou subconjunto) de agrupamentos a serem realizados
  - pode produzir subtotais de forma equivalente ao ROLLUP e ao CUBE, além de outros subtotais

Exemplos

```
SELECT chaveFilial, chaveProduto, chaveTempo,
        SUM(lucroDolar) AS "Lucro Anual",
        SUM(unidadesVendidas) AS "Unidades Vendidas"
FROM vendas
GROUP BY GROUPING SETS ((chaveFilial, chaveProduto, chaveTempo),
            (chaveFilial, chaveProduto), (chaveFilial), ());
```

```
SELECT chaveFilial, chaveProduto, chaveTempo,
        SUM(lucroDolar) AS "Lucro Anual",
        SUM(unidadesVendidas) AS "Unidades Vendidas"
FROM vendas
GROUP BY GROUPING SETS ((chaveFilial, chaveProduto, chaveTempo),
(chaveFilial, chaveProduto), (chaveProduto,chaveTempo), (chaveFilial,
chaveTempo), (chaveFilial), (chaveProduto), (chaveTempo), ());
```

FUNÇÕES DE JANELAMENTO

- WINDOW PARTITIONING
  - Cria uma partição, permitindo a comparação entre medidas numéricas agregadas e medidas numéricas não agregadas

WINDOW PARTITIONING

- Listar a quantidade de unidades vendidas por filial, por produto e por tempo individualmente, e ao mesmo tempo comparar com a soma das quantidades das unidades vendidas por filial

```
SELECT chaveFilial, chaveProduto, chaveTempo,
        unidadesVendidas AS "Vendas por Filial",
        SUM(unidadesVendidas) OVER (PARTITION BY chaveFilial)
        AS "Total Unidades Vendidas Filial"
FROM vendas
ORDER BY chaveFilial, chaveProduto, chaveTempo;
```