



Luiz Paulo de Oliveira Santos

lpaulo@fast-report.com

OTIMIZANDO BANCOS DE DADOS FIREBIRD PARA GERAÇÃO DE RELATÓRIOS E CUBS DE DECISÃO (FAST-REPORT)





- **Origem**
- **Finalidade**
- **Características**
- **Negócios**
- **Dispositivos de B.I.**
- **Cubos**
- **SQL para Cubos**





- Extrair e integrar dados de múltiplas fontes
- Fazer uso da experiência
- Analisar dados contextualizados
- Trabalhar com hipóteses
- Procurar relações de causa e efeito
- Transformar os registros obtidos em informação útil para o conhecimento empresarial





O que é O.L.A.P?

On-Line Analytical Processing

Onde OLAP se integra com B.I.?

Business Intelligence



OLAP - On-Line Analytical Processing



Processamento Analítico On-Line

OLAP provê apoio ao usuário final na tomada de decisão estratégica e suas tendências.

É uma ferramenta baseada em modelos matemáticos e contábeis mas que visa basicamente resumir a base estudada em um modelo que permita efetuar complexas consultas com poucos cliques.

O melhor cubo do mercado é o **Fast-Cube** da Fast-Report!!!



Velocidades do FastCube



NativeBD

Opening	Loading	Processing	Result
0,079	0,495	1,551	2,125
0,016	0,641	1,359	2,016
0,015	0,454	1,515	1,984

Load table 28000 strings

ADO

Opening	Loading	Processing	Result
0,093	1,472	5,029	6,594
0,078	1,359	5,079	6,516
0,078	1,446	5,023	6,547

26 diferentes tipos de campos

Três métodos de acesso à dados
(você também pode usar com *IBO*,
IBX and *FIBPlus* 😊)

BDE

Opening	Loading	Processing	Result
0,297	1,558	3,719	3,719
0,015	1,373	2,049	3,437
0,015	1,546	1,861	3,422

Tempo em segundos





Otimização do banco de dados

Criação de VIEWS no banco de dados

Criação de Índices no banco / tabelas

Conhecer o otimizador de consultas do banco de dados

Refazer a engenharia do seu cubo periodicamente



Como funciona?

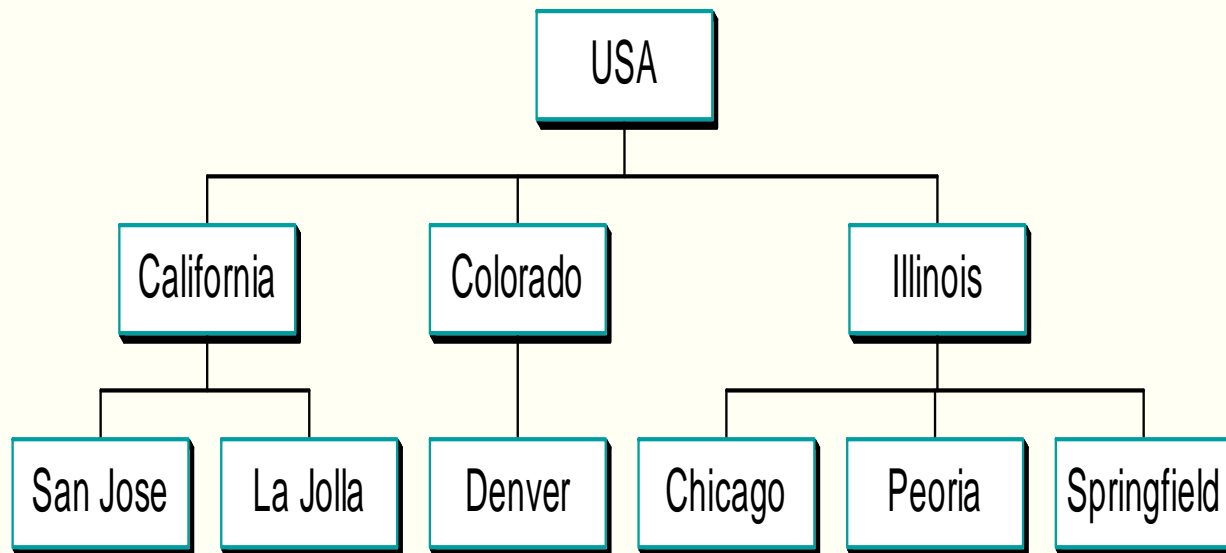


Dimensões

Sumários



Dimensões Comuns



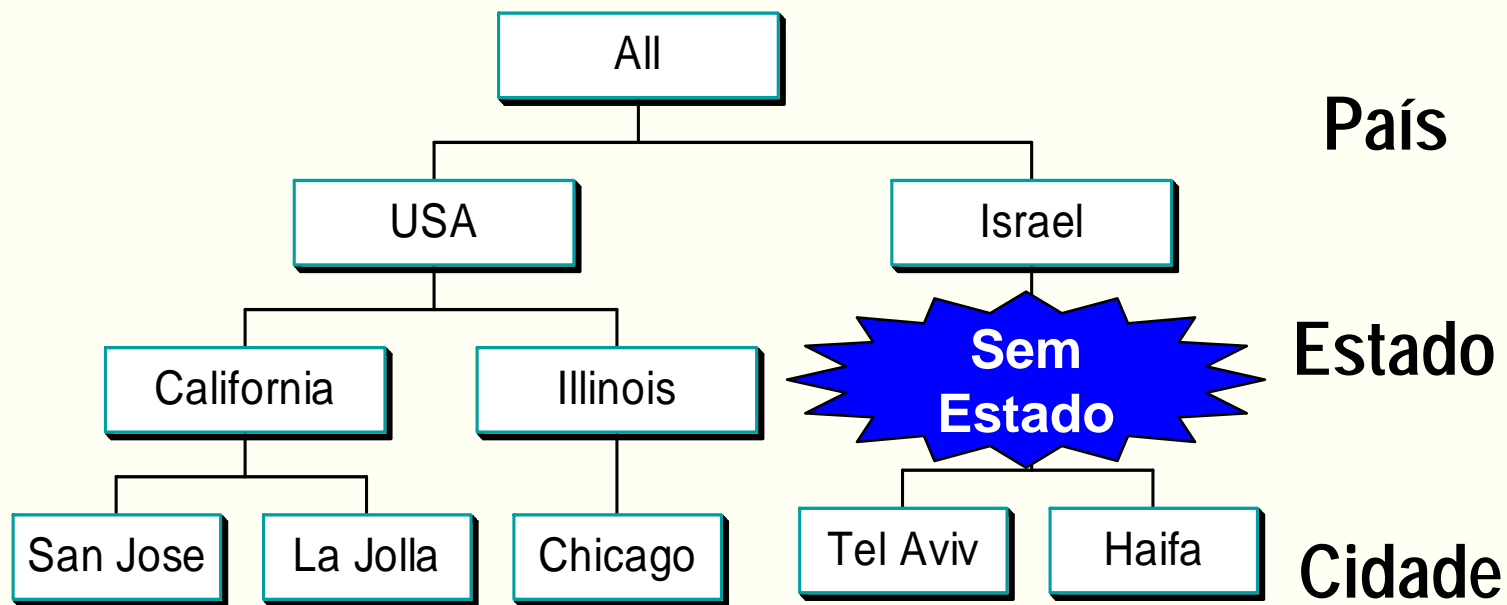
País

Estado

Cidade



Dimensões Especiais





```
select  
  case  
    when estado is NULL then 'SEM ESTADO'  
    else estado  
  end  
  
from LOCAIS  
  
group by PAIS, ESTADO
```





Campos a utilizar para extração de dados

Geralmente campos nque podem ser TOTALIZADOS

Dados não agrupáveis



Funções para sumários



<i>Aggregate function</i>	<i>Description</i>
AVG	Average of values in a numeric expression
COUNT	Number of values in an expression
COUNT (*)	Number of selected rows
MAX	Highest value in the expression
MIN	Lowest value in the expression
SUM	Total values in a numeric expression
STDEV	Statistical deviation of all values
STDEVP	Statistical deviation for the population
VAR	Statistical variance of all values
VARP	Statistical variance of all values for the population





Operações condicionais de sumário:

- Group by

- Having





- Próximas implementações:

– WITH ROLLUP

No Firebird em algum momento estará disponível!



Group by



```
USE northwind
SELECT productid,orderid,quantity
FROM orderhist
GO
```

productid	orderid	quantity
1	1	5
1	1	10
2	1	10
2	2	25
3	1	15
3	2	30

```
USE northwind
SELECT productid
      ,SUM(quantity) AS total_quantity
FROM orderhist
GROUP BY productid
GO
```

productid	total_quantity
1	15
2	35
3	45

Only rows that satisfy the WHERE clause are grouped

productid	total_quantity
2	35

```
USE northwind
SELECT productid
      ,SUM(quantity) AS total_quantity
FROM orderhist
WHERE productid = 2
GROUP BY productid
GO
```



Having



```
SELECT
    PROJ_ID, SUM(PROJECTED_BUDGET) AS TOTAL_BUDGET
FROM PROJ_DEPT_BUDGET
WHERE FISCAL_YEAR = 1994 AND PROJ_ID STARTING WITH 'M'
GROUP BY PROJ_ID
HAVING SUM(PROJECTED_BUDGET) >= 100000;
```



Principal Problema!



NULL



COALESCE()



```
select part.CODIGO,part.DESCRICAO
Produto,part.UNIDADE_REFERENCIA,part.VALOR_VISTA,
part.VALOR_PRAZO,part.COD_GRUPO,G.DESCRICAO,
COALESCE(extra.SALDO_INICIAL,0) + COALESCE(extra.ENTRADAS,0) +
COALESCE(extra.RMA_ENTRADAS,0) -
COALESCE(extra.SAIDAS,0) - COALESCE(extra.PERDAS,0) -
COALESCE(extra.CONSUMO_INTERNO,0) -
COALESCE(extra.DOACOES,0) - COALESCE(extra.RMA_Saidas,0) AS Qtde
from (PRODUTOS part
inner join GRUPOS Grupo on part.COD_GRUPO = Grupo.CODIGO)
inner join Estoque Extra on part.CODIGO = extra.COD_PRODUTO
where part.CODIGO BETWEEN info2 and info3
ORDER BY 7
```





**CREATE VIEW TELEFONES (NOME,
TELEFONE) AS SELECT NOME,
TELEFONE FROM CLIENTES ORDER BY
NOME DESC, TELEFONE**



Importância dos índices



Acelerar a obtenção de dados

Implementar chaves de integridade relacional

Ordenar os dados DIMENSIONADOS

Agrupar: Group by



Criação de índices



```
CREATE TRIGGER BIU_PERSONS
FOR PERSONS ACTIVE BEFORE INSERT OR UPDATE AS
BEGIN
    NEW.LAST_NAME_UPPER = UPPER (NEW.LAST_NAME);
    NEW.FIRST_NAME_UPPER = UPPER (NEW.FIRST_NAME);
END
```

```
CREATE INDEX IDX_LASTNAME ON PERSONS (LAST_NAME_UPPER);
CREATE INDEX IDX_FIRSTNAME ON PERSONS (FIRST_NAME_UPPER);
```



SELECT em STORED PROCEDURE



```
CREATE PROCEDURE CountTo10  
RETURNS (Cnt INTEGER)  
AS BEGIN  
  Cnt = 1;  
  WHILE (Cnt <= 10) DO BEGIN  
    SUSPEND; /* Return next line */  
    Cnt = Cnt + 1;  
  END;  
END
```



Agrupamento condicional



- Quando o campo sofre mudança em tempo de execução
- Quando implementa-se um SELECT em uma procedure



Agrupamento condicional



```
select
  case when username='ELENA' then 'HELENA'
        else username
  end as NOME
from log group by NOME;
```

Implementando algo como SOUNDEX()



Inverter colunas por linhas



Inverter colunas por linhas em cubos é uma operação corriqueira, logo teremos que escrever código para tal.

Nome	Qtd_Produzida
Database Turbo	5
Matematica Financeira	5
PMI	4

Database Turbo	Matematica Financeira	PMI
5	5	4



É isso aí. Muito obrigado!



Dúvidas?

