
FastBit e Índice Bitmap de Junção

AGENDA

- **Introdução**
- Instalação
- Índice Bitmap de Junção
- Construção do índice
- Consultas sobre o índice

Introdução

- FastBit é um software desenvolvido pelo Lawrence Berkeley National Laboratory
- Usado para a construção e o uso do Índice Bitmap
 - Este índice é amplamente empregado em DW para diminuir o tempo de resposta das consultas
- Codificado em C++
- <https://sdm.lbl.gov/fastbit/>

AGENDA

- Introdução
- **Instalação**
- Índice Bitmap de Junção
- Construção do índice
- Consultas sobre o índice

Requisitos

- Sistema operacional Windows
 - Compilador Visual C++ 2008 Express
- Sistema operacional Linux
 - Compilador GCC
- PostgreSQL
- FastBit 1.2.1

Instalação - Linux

Usando Ubuntu 9.04.

- **1° Passo:** em <https://codeforge.lbl.gov/projects/fastbit/> façam o download do “fastbit ibis1.2.1 2010-11-03”.
- **2° Passo:** descompactar o arquivo
- **3° Passo:** dentro da pasta criada pelo arquivo descompactado, executar o comando `./configure`

Instalação - Linux

- **4° Passo:** usar o comando *make all*
- **5° Passo:** usar o comando *make check*
- **6° Passo:** usar o comando *make install*

Instalação - Windows

É necessário compilar o código-fonte.

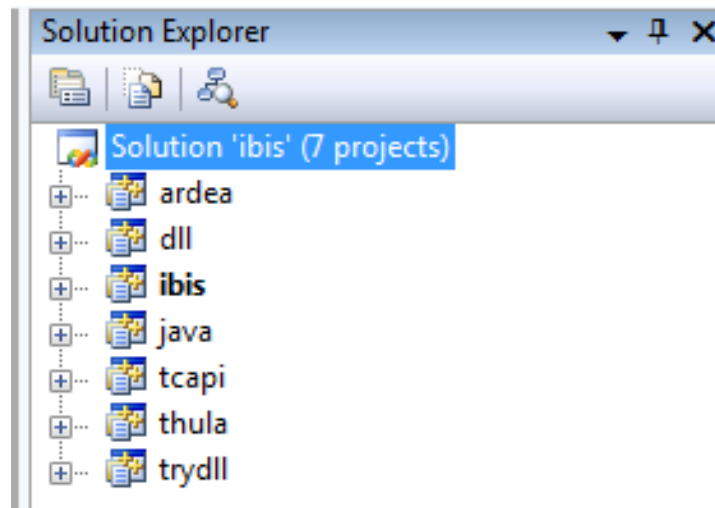
- **1º Passo:** instalar o Visual C++ Express Edition (caso não exista no pc uma outra versão, como o Visual Studio)
- **2º Passo:** em <https://codeforge.lbl.gov/projects/fastbit/> façam o download do “fastbit ibis1.2.1 2010-11-03”.
 - Descompactem esta pasta, preferencialmente, no C:\
- **3º Passo:** precisamos da biblioteca “POSIX Threads for Win32” para compilar o FastBit. Façam o download da versão 2.8.0: (<ftp://sourceware.org/pub/pthreads-win32/pthreads-w32-2-8-0-release.exe>) Instalem esta biblioteca, de preferência no C:\

Instalação - Windows

- **4° Passo:** Explore a pasta onde ficaram os arquivos da Pthreads e copie os arquivos conforme as imagens a seguir.
 - **Cuidado para não mover os arquivos. Criem apenas cópias deles e cole nestes destinos**
 - Pre2build\lib*.dll para C:\Windows\System32
 - Pre2build\include\ (os 3 arquivos) para C:\fastbit-ibis\1.0.8\src\
- **5° Passo:** No Visual C++ Express, abra o projeto ibis.sln (na pasta C:\fastbit-ibis\1.0.8\win\ibis.sln)

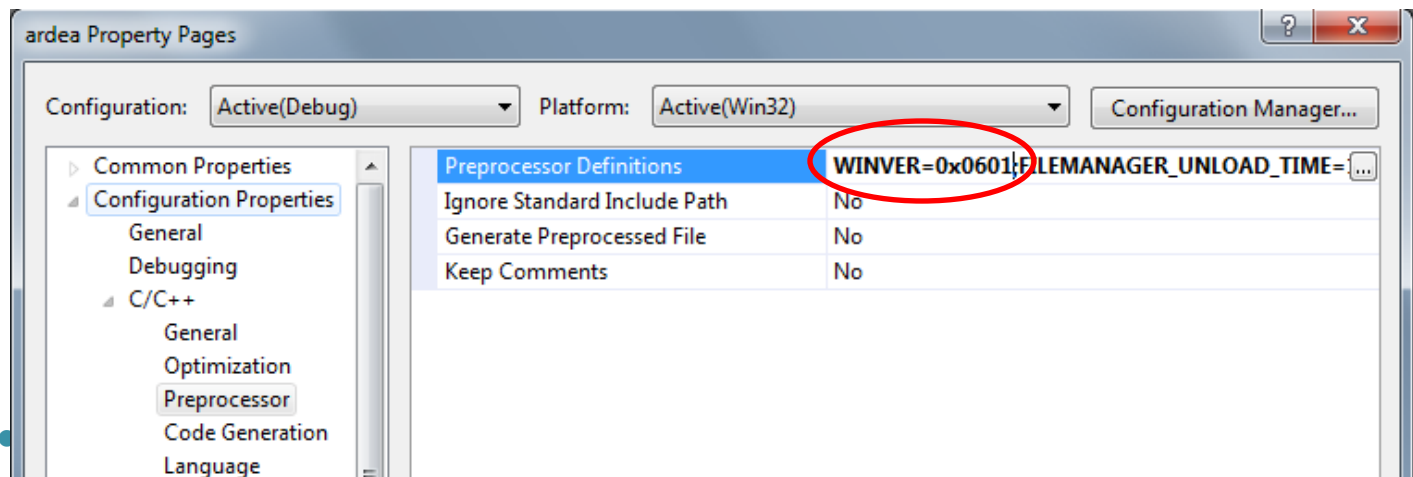
Instalação

- **6° Passo (opcional):** Alterar o winver de 0x0600 (Vista) para 0x0501 (XP) ou 0x0601 (Seven) nos arquivos **ardea**, **dll**, **ibis**, **java** e **thula** no Solution Explorer.



Instalação

- Clicar com o botão da direita no arquivo e depois em Properties.
- Ir na guia Configuration Properties -> C/C++ -> Preprocessador
- Alterar o WINVER



AGENDA

- Introdução
- Instalação
- Índice Bitmap de Junção
- Construção do índice
- Consultas sobre o índice

Índice Bitmap

- ❑ Conceitualmente, um índice *bitmap* é um vetor de bits para cada chave do índice, em que cada bit substitui um *rowid*; se o bit é 1, então o correspondente *rowid* contém o valor da chave.
- ❑ Uma função de mapeamento converte a posição de um bit para o *rowid* real.

Vantagens:

- ❑ Tempos de respostas reduzidos para uma larga gama de consultas *ad hoc*
- ❑ Espaço de armazenamento pequeno, quando comparado com outros tipos de índice

Índice Bitmap

Desvantagens:

- ❑ A atualização de índices *bitmap* não é tão eficiente quanto a de índices árvores-B, o que faz com que índices *bitmap* sejam indicados para BDs *read-only*, o que é o caso de DW.
- ❑ Para que haja ganho de espaço, é preciso que o número de valores da chave do índice seja pequeno.

Exemplo:

Suponha uma tabela Cliente, com índice bitmap por sexo (2 valores), estado civil (3 valores) e nível de renda (12 níveis).

Seja a consulta:

```
SELECT COUNT(*) FROM customers
WHERE cust_marital_status = 'married'
AND cust_income_level IN (8, 9);
```

Índice Bitmap

Índices bitmap podem processar eficientemente esta consulta, simplesmente contando o número de bits da composição lógica dos bits.



| Status = 'c' | | Nível = 8 | | Nível = 9 | | | | | | | |
|--------------|-----|-----------|----|-----------|---|---|-----|---|---|---|--|
| 0 | | 0 | | 0 | | 0 | | 0 | | 0 | |
| 1 | | 1 | | 0 | | 1 | | 1 | | 1 | |
| 1 | AND | 0 | OR | 1 | = | 1 | AND | 1 | = | 1 | |
| 0 | | 0 | | 1 | | 0 | | 1 | | 0 | |
| 0 | | 1 | | 0 | | 0 | | 1 | | 0 | |
| 1 | | 1 | | 0 | | 1 | | 1 | | 1 | |

Lidando com valores *Nulls*:

Diferentemente da maioria dos outros tipos de índice, índices *bitmap* incluem linhas que têm valores NULL.

```
SELECT COUNT(*) FROM customers WHERE cust_marital_status IS NULL;
```

Esta consulta usará um índice *bitmap* sobre *cust_marital_status*. Note que esta consulta não será capaz de usar um índice árvore-B.

Índice Bitmap de Junção

TABELA DE DIMENSÃO: Supplier

| <u>s_suppkey</u> | <u>s_address</u> | s_city | s_nation | s_region |
|------------------|------------------|-----------|----------|----------|
| 1 | A | VIETNAM 2 | VIETNAM | ASIA |
| 2 | B | FRANCE 5 | FRANCE | EUROPE |
| 3 | C | ROMANIA 2 | ROMANIA | EUROPE |
| 4 | D | ALGERIA 6 | ALGERIA | AFRICA |
| 5 | E | ALGERIA 0 | ALGERIA | AFRICA |

TABELA DE FATOS: lineorder

| <u>lo_suppkey</u> | <u>lo_custkey</u> | rev |
|-------------------|-------------------|-----|
| 1 | 235 | 20 |
| 1 | 512 | 16 |
| 2 | 512 | 22 |
| 3 | 235 | 19 |
| 3 | 512 | 15 |
| 3 | 106 | 21 |
| 4 | 235 | 20 |
| 5 | 106 | 18 |

s_address

?

Índice Bitmap de Junção

TABELA DE DIMENSÃO: Supplier

| <u>s_suppkey</u> | <u>s_address</u> | <u>s_city</u> | <u>s_nation</u> | <u>s_region</u> |
|------------------|------------------|---------------|-----------------|-----------------|
| 1 | A | VIETNAM 2 | VIETNAM | ASIA |
| 2 | B | FRANCE 5 | FRANCE | EUROPE |
| 3 | C | ROMANIA 2 | ROMANIA | EUROPE |
| 4 | D | ALGERIA 6 | ALGERIA | AFRICA |
| 5 | E | ALGERIA 0 | ALGERIA | AFRICA |

TABELA DE FATOS: lineorder

| <u>lo_suppkey</u> | <u>lo_custkey</u> | <u>rev</u> |
|-------------------|-------------------|------------|
| 1 | 235 | 20 |
| 1 | 512 | 16 |
| 2 | 512 | 22 |
| 3 | 235 | 19 |
| 3 | 512 | 15 |
| 3 | 106 | 21 |
| 4 | 235 | 20 |
| 5 | 106 | 18 |

s_address

| A |
|---|
| 1 |
| 1 |
| 0 |
| 0 |
| 0 |
| 0 |
| 0 |
| 0 |

Índice Bitmap de Junção

TABELA DE DIMENSÃO: Supplier

| <u>s_suppkey</u> | <u>s_address</u> | <u>s_city</u> | <u>s_nation</u> | <u>s_region</u> |
|------------------|------------------|---------------|-----------------|-----------------|
| 1 | A | VIETNAM 2 | VIETNAM | ASIA |
| 2 | B | FRANCE 5 | FRANCE | EUROPE |
| 3 | C | ROMANIA 2 | ROMANIA | EUROPE |
| 4 | D | ALGERIA 6 | ALGERIA | AFRICA |
| 5 | E | ALGERIA 0 | ALGERIA | AFRICA |

TABELA DE FATOS: lineorder

| <u>lo_suppkey</u> | <u>lo_custkey</u> | <u>rev</u> |
|-------------------|-------------------|------------|
| 1 | 235 | 20 |
| 1 | 512 | 16 |
| 2 | 512 | 22 |
| 3 | 235 | 19 |
| 3 | 512 | 15 |
| 3 | 106 | 21 |
| 4 | 235 | 20 |
| 5 | 106 | 18 |

s_address

| A | B | C | D | E |
|---|---|---|---|---|
| 1 | 0 | 0 | 0 | 0 |
| 1 | 0 | 0 | 0 | 0 |
| 0 | 1 | 0 | 0 | 0 |
| 0 | 0 | 1 | 0 | 0 |
| 0 | 0 | 1 | 0 | 0 |
| 0 | 0 | 0 | 1 | 0 |
| 0 | 0 | 0 | 0 | 1 |

Índice Bitmap de Junção

Region < Nation < City < Address

TABELA DE DIMENSÃO: Supplier

| s_suppkey | s_address | s_city | s_nation | s_region |
|-----------|-----------|-----------|----------|----------|
| 1 | A | VIETNAM 2 | VIETNAM | ASIA |
| 2 | B | FRANCE 5 | FRANCE | EUROPE |
| 3 | C | ROMANIA 2 | ROMANIA | EUROPE |
| 4 | D | ALGERIA 6 | ALGERIA | AFRICA |
| 5 | E | ALGERIA 0 | ALGERIA | AFRICA |

TABELA DE FATOS: lineorder

| lo_suppkey | lo_custkey | rev |
|------------|------------|-----|
| 1 | 235 | 20 |
| 1 | 512 | 16 |
| 2 | 512 | 22 |
| 3 | 235 | 19 |
| 3 | 512 | 15 |
| 3 | 106 | 21 |
| 4 | 235 | 20 |
| 5 | 106 | 18 |

s_address

| A | B | C | D | E |
|---|---|---|---|---|
| 1 | 0 | 0 | 0 | 0 |
| 1 | 0 | 0 | 0 | 0 |
| 0 | 1 | 0 | 0 | 0 |
| 0 | 0 | 1 | 0 | 0 |
| 0 | 0 | 1 | 0 | 0 |
| 0 | 0 | 0 | 1 | 0 |
| 0 | 0 | 0 | 0 | 1 |

s_region

'EUROPE'

| |
|--|
| |
| |
| |
| |
| |
| |
| |
| |
| |

?

Índice Bitmap de Junção

Region < Nation < City < Address

TABELA DE DIMENSÃO: Supplier

| <u>s_suppkey</u> | <u>s_address</u> | <u>s_city</u> | <u>s_nation</u> | <u>s_region</u> |
|------------------|------------------|---------------|-----------------|-----------------|
| 1 | A | VIETNAM 2 | VIETNAM | ASIA |
| 2 | B | FRANCE 5 | FRANCE | EUROPE |
| 3 | C | ROMANIA 2 | ROMANIA | EUROPE |
| 4 | D | ALGERIA 6 | ALGERIA | AFRICA |
| 5 | E | ALGERIA 0 | ALGERIA | AFRICA |

TABELA DE FATOS: lineorder

| <u>lo_suppkey</u> | <u>lo_custkey</u> | <u>rev</u> |
|-------------------|-------------------|------------|
| 1 | 235 | 20 |
| 1 | 512 | 16 |
| 2 | 512 | 22 |
| 3 | 235 | 19 |
| 3 | 512 | 15 |
| 3 | 106 | 21 |
| 4 | 235 | 20 |
| 5 | 106 | 18 |

s_address

| | A | B | C | D | E |
|---|---|---|---|---|---|
| 1 | 1 | 0 | 0 | 0 | 0 |
| 1 | 1 | 0 | 0 | 0 | 0 |
| 0 | 0 | 1 | 0 | 0 | 0 |
| 0 | 0 | 0 | 1 | 0 | 0 |
| 0 | 0 | 0 | 1 | 0 | 0 |
| 0 | 0 | 0 | 0 | 1 | 0 |
| 0 | 0 | 0 | 0 | 0 | 1 |

s_region

| 'EUROPE' |
|----------|
| |
| |
| |
| |
| |
| |
| |
| |
| |

?

Índice Bitmap de Junção

Region < Nation < City < Address

TABELA DE DIMENSÃO: Supplier

| s_suppkey | s_address | s_city | s_nation | s_region |
|-----------|-----------|-----------|----------|----------|
| 1 | A | VIETNAM 2 | VIETNAM | ASIA |
| 2 | B | FRANCE 5 | FRANCE | EUROPE |
| 3 | C | ROMANIA 2 | ROMANIA | EUROPE |
| 4 | D | ALGERIA 6 | ALGERIA | AFRICA |
| 5 | E | ALGERIA 0 | ALGERIA | AFRICA |

TABELA DE FATOS: lineorder

| lo_suppkey | lo_custkey | rev |
|------------|------------|-----|
| 1 | 235 | 20 |
| 1 | 512 | 16 |
| 2 | 512 | 22 |
| 3 | 235 | 19 |
| 3 | 512 | 15 |
| 3 | 106 | 21 |
| 4 | 235 | 20 |
| 5 | 106 | 18 |

s_address

| | A | B OR C | D | E |
|---|---|--------|---|---|
| 1 | 0 | 0 | 0 | 0 |
| 1 | 0 | 0 | 0 | 0 |
| 0 | 1 | 0 | 0 | 0 |
| 0 | 0 | 1 | 0 | 0 |
| 0 | 0 | 1 | 0 | 0 |
| 0 | 0 | 0 | 1 | 0 |
| 0 | 0 | 0 | 0 | 1 |

s_region

| 'EUROPE' |
|----------|
| 0 |
| 0 |
| 1 |
| 1 |
| 1 |
| 0 |
| 0 |

Índice Bitmap de Junção

Region < Nation < City < Address

TABELA DE DIMENSÃO: Supplier

| s_suppkey | s_address | s_city | s_nation | s_region |
|-----------|-----------|-----------|----------|----------|
| 1 | A | VIETNAM 2 | VIETNAM | ASIA |
| 2 | B | FRANCE 5 | FRANCE | EUROPE |
| 3 | C | ROMANIA 2 | ROMANIA | EUROPE |
| 4 | D | ALGERIA 6 | ALGERIA | AFRICA |
| 5 | E | ALGERIA 0 | ALGERIA | AFRICA |

TABELA DE FATOS: lineorder

| lo_suppkey | lo_custkey | rev |
|------------|------------|-----|
| 1 | 235 | 20 |
| 1 | 512 | 16 |
| 2 | 512 | 22 |
| 3 | 235 | 19 |
| 3 | 512 | 15 |
| 3 | 106 | 21 |
| 4 | 235 | 20 |
| 5 | 106 | 18 |

s_address

| A | B | C | D | E |
|---|---|---|---|---|
| 1 | 0 | 0 | 0 | 0 |
| 1 | 0 | 0 | 0 | 0 |
| 0 | 1 | 0 | 0 | 0 |
| 0 | 0 | 1 | 0 | 0 |
| 0 | 0 | 1 | 0 | 0 |
| 0 | 0 | 0 | 1 | 0 |
| 0 | 0 | 0 | 0 | 1 |

s_region

| 'ASIA' | 'EUROPE' | 'AFRICA' |
|--------|----------|----------|
| 1 | 0 | 0 |
| 1 | 0 | 0 |
| 0 | 1 | 0 |
| 0 | 1 | 0 |
| 0 | 1 | 0 |
| 0 | 0 | 1 |
| 0 | 0 | 1 |

Índice Bitmap de Junção

- **Processamento de consultas**
 - Velozes operações lógicas bit-a-bit
 - Pode evitar a junção estrela

TABELA DE FATOS: lineorder

| lo_suppkey | lo_custkey | rev |
|------------|------------|-----|
| 1 | 235 | 20 |
| 1 | 512 | 16 |
| 2 | 512 | 22 |
| 3 | 235 | 19 |
| 3 | 512 | 15 |
| 3 | 106 | 21 |
| 4 | 235 | 20 |
| 5 | 106 | 18 |

WHERE lo_custkey=512 *AND* s_region = 'ASIA'

lo_custkey=512

| |
|---|
| 0 |
| 1 |
| 1 |
| 0 |
| 1 |
| 0 |
| 0 |
| 0 |

AND

s_region='ASIA'

| |
|---|
| 1 |
| 1 |
| 0 |
| 0 |
| 0 |
| 0 |
| 0 |
| 0 |

=

| |
|---|
| 0 |
| 1 |
| 0 |
| 0 |
| 0 |
| 0 |
| 0 |
| 0 |

Considerações

□ Índice Bitmap em DW convencionais

- Junção estrela e hierarquias de atributos

□ Vantagens

- Multidimensionalidade não deteriora drasticamente o desempenho do processamento de consultas

□ Desvantagem

- Atributos com alta cardinalidade
 - *Binning*, Compressão, Codificação

AGENDA

- Introdução
- Instalação
- Índice Bitmap de Junção
- **Construção do índice**
- Consultas sobre o índice

Construção

1. Preparação dos dados: gerar arquivo csv separado por vírgula com informações das tabelas;
2. Criação das partições-ferramenta ardea:
 - `ardea -d /bjin/index0 -m "a:int, b:float, c:short" -t /bjin/test0.csv`
 - `-d` → diretório de saída;
 - `-m` → especificar nomes e tipos das colunas
 - `t` → arquivo com informações para geração das partições

Construção

Tipos de dados:

byte: 8-bit signed integer.

unsigned int: unsigned 32-bit signed integer.

short: 16-bit signed integer.

unsigned long: unsigned 64-bit signed integer.

int: 32-bit signed integer.

float: 32-bit IEEE floating-point values.

long: 64-bit signed integer.

double: 64-bit IEEE floating-point values.

unsigned byte: unsigned 8-bit signed integer.

key: String values with a small number of distinct choices.

unsigned short: unsigned 16-bit signed integer.

text: Arbitrary string values.

Construção

- I. Criação das partições-ferramenta ardea:
 - Trata dados como tabelas com linhas e colunas;
 - Divisão de tabelas em “partições”;
 - Cada coluna é armazenada em um arquivo separado na forma binária pura;
 - O nome do arquivo é o mesmo da coluna;
 - O arquivo de metadados –part.txt informações como: nome da partição, número de linhas e colunas de uma partição, nome das colunas, etc.

Construção

I. Criação das partições-arquivos de dados:

| Arquivo | Descrição |
|----------------|---|
| -part-txt | Arquivo de metadados. Descreve o conteúdo das partições. |
| colname | Nome do arquivo de dados para uma coluna. |
| colname.idx | Nome do arquivo de índice associado à coluna colname. |
| colname.msk | Vetor de bits da máscara para valores null da coluna. |
| colname.bin | Versão reordenada dos dados. Utilizado para aumentar velocidade |
| colname.sp | Posição inicial das strings nos arquivos de colunas do tipo string. |
| colname.tdlist | Lista usada para construir o índice . |
| colname.terms | Termos usados em colname.tdlist. |
| colname.dic | Dicionário que mapeia string em inteiros. |
| colname.int | Versão inteira dos valores categóricos (traduzidos no dicionário de dados). |

Construção

1. Criação do índice:

- `ibis -v -d /bjin/index0 -b "<binning none/><encoding equality/>"`
- `none` → gera um bitmap para cada valor distinto de atributo;

2. Execução da consulta:

- `ibis -d /bjin/index0 -q "select sum(extendedprice*discount) where year = 1993 and discount between 1 and 3 and quantity < 25" -o /bjin/index0/query.csv -l /bjin/index0/log.txt`
 - `-d` → diretório da partição
 - `-q` → consulta a ser executada
 - `-o` → arquivo de saída
 - `-l` → arquivo de log da consulta

Preparando o BD (usando o PostgreSQL)

- **1º Passo:** Copiar os arquivos TBL (do SSB) para /opt/postgres
- **2º Passo:** criar as tabelas e povoar com os arquivos TBL.

- Exemplo:

```
CREATE TABLE supplier (  
    s_suppkey integer NOT NULL,  
    s_name character(25),  
    s_address character varying(25),  
    s_city character(10),  
    s_nation character(15),  
    s_region character(12),  
    s_phone character(15),  
    atnull character(1),  
    CONSTRAINT supplier_pk PRIMARY KEY (s_suppkey)
```

```
);
```

```
copy supplier from '/opt/postgressupplier.tbl'
```

```
with delimiter '|';
```

```
alter table supplier drop column atnull;
```

Construção do Índice

ÍNDICE BITMAP DE JUNÇÃO

Usar o conteúdo do arquivo `construcao_consulta_fastbit.txt`

Segue um exemplo.

Consulta Q1 do SSB:

```
SELECT sum(lo_extendedprice*lo_discount) AS revenue
FROM lineorder, date
WHERE lo_orderdate = d_datekey
      AND d_year = [YEAR] -- Specific values below
      AND lo_discount BETWEEN [DISCOUNT] - I AND
      [DISCOUNT] + I
      AND lo_quantity < [QUANTITY];
```


Construção do Índice

ÍNDICE BITMAP DE JUNÇÃO

No SGBD:

- Computação da junção para a criação do Índice Bitmap de Junção.

```
CREATE TABLE qI AS
```

```
  SELECT lo_extendedprice, lo_discount, lo_quantity, d_year, d_yearmonthnum,  
         d_weeknuminyear
```

```
  FROM lineorder, date
```

```
  WHERE lo_orderdate = d_datekey;
```

- Exportar a junção computada para um arquivo CSV (usando um comando do PostgreSQL):

```
COPY qI TO '/bjin/SSB/qI.csv'
```

```
WITH DELIMITER ',' CSV QUOTE '\"';
```

(se for exportar uma string, antes do “;” deve-se acrescentar FORCE QUOTE seguido do nome da coluna (s_region, por exemplo))

OBS: A pasta deve ter permissão de escrita para o usuário Postgres

Construção do Índice

ÍNDICE BITMAP DE JUNÇÃO

No SGBD:

- Apagar a junção:
`DROP TABLE q1;`
- Copiar o CSV para /bjin/ (por exemplo)

No *terminal* de comando:

- Ir em `/opt/fastbit/`

Construção do Índice

ÍNDICE BITMAP DE JUNÇÃO

No Fastbit:

- Criar a partição:

```
./ardea -d /bjin/ssb/index0 -m "col0:double, col1:double, col2:double, col3:double,  
col4:double, col5:key, col6:key, col7:key, col8:key, col9:key, col10:key, col11:key,  
col12:key, col13:key, col14:int, col15:key, col16:key"
```

- Criar o índice bitmap:

```
ibis -v -d /bjin/ssb/index0 -b "<binning none/><encoding equality/>"
```

Construção do Índice

ÍNDICE BITMAP DE JUNÇÃO

Tabela de Mapeamento das colunas

| FastBit | SGBD |
|----------------|------------------|
| col0 | lo_revenue |
| col1 | lo_extendedprice |
| col2 | lo_discount |
| col3 | lo_quantity |
| col4 | lo_supplycost |
| col5 | c_region |
| col6 | c_nation |
| col7 | c_city |
| col8 | s_region |
| col9 | s_nation |

| FastBit | SGBD |
|----------------|-------------|
| col10 | s_city |
| col11 | p_brand1 |
| col12 | p_category |
| col13 | p_mfgr |
| col14 | d_year |
| col15 | d_month |
| col16 | d_dayofweek |

AGENDA

- Introdução
- Instalação
- Índice Bitmap de Junção
- Construção do índice
- Consultas sobre o índice

Consulta Q1.1

- **No SGBD:**

```
select sum(lo_extendedprice*lo_discount) as revenue
  from lineorder, date
 where lo_orderdate = d_datekey
       and d_year = 1993
       and lo_discount between 1 and 3
       and lo_quantity < 25;
```

- **No Fastbit :**

```
ibis -d /bjin/ssb/index0 -q "select sum(col1*col2) where
col14 = 1993 and col2
between 1 and 3 and col3 < 25" -o /bjin/ssb/resultq11.csv
```

Consulta Q1.2

- No SGBD:

```
select sum(lo_extendedprice*lo_discount) as revenue
from lineorder, date
where lo_orderdate = d_datekey
and d_yearmonthnum = 199401
and lo_discount between 4 and 6
and lo_quantity between 26 and 35;
```

- No Fastbit :

```
./ibis -d /bjin/ssb/index0 -v -l q12_fastbit.log -o q12_fastbit.txt -q "select
sum (lo_extendedprice * lo_discount) where d_yearmonthnum =
199401 and lo_discount >= 4 and lo_discount <=6 and lo_quantity >=
26 and lo_quantity <= 35"
```

Consulta Q1.3

- No SGBD:

```
select sum(lo_extendedprice*lo_discount) as revenue
from lineorder, date
where lo_orderdate = d_datekey
and d_weeknuminyear = 6
and d_year = 1994
and lo_discount between 5 and 7
and lo_quantity between 26 and 35;
```

- No Fastbit :

```
./ibis -d /bjin/ssb/index0 -v -l q13_fastbit.log -o q13_fastbit.txt -q "select
sum (lo_extendedprice * lo_discount) where d_weeknuminyear = 6 and
d_year = 1994 and lo_discount >= 5 and lo_discount <=7 and
lo_quantity >= 26 and lo_quantity <= 35"
```


Consulta Q2.1

- No SGBD:

```
select sum(lo_revenue), d_year, p_brandl
from lineorder, date, part, supplier
where lo_orderdate = d_datekey
and lo_partkey = p_partkey
and lo_suppkey = s_suppkey
and p_category = 'MFGR#12'
and s_region = 'AMERICA'
group by d_year, p_brandl
order by d_year, p_brandl;
```

- No Fastbit :

```
./ibis -d /bjin/ssb/index0 -v -l q21_fastbit.log -o q21_fastbit.txt -q "select
sum(lo_revenue), d_year, p_brandl where p_category = 'MFGR#12' and
s_region = 'AMERICA    ' ;
```

Consulta Q2.2

- No SGBD:

```
select sum(lo_revenue), d_year, p_brandl
from lineorder, date, part, supplier
where lo_orderdate = d_datekey
and lo_partkey = p_partkey
and lo_suppkey = s_suppkey
and p_brandl between 'MFGR#2221' and 'MFGR#2228'
and s_region = 'ASIA'
group by d_year, p_brandl
order by d_year, p_brandl;
```

- No Fastbit :

```
./ibis -d /bjin/ssb/index0 -v -l q22_fastbit.log -o q22_fastbit.txt -q "select
sum(lo_revenue), d_year, p_brandl where (p_brandl = 'MFGR#2221' or
p_brandl = 'MFGR#2222' or p_brandl = 'MFGR#2223' or p_brandl =
'MFGR#2224' or p_brandl = 'MFGR#2225' or p_brandl = 'MFGR#2226' or
p_brandl = 'MFGR#2227' or p_brandl = 'MFGR#2228') and s_region =
'ASIA'"
```

Consulta Q2.3

- No SGBD:

```
select sum(lo_revenue), d_year, p_brandl
from lineorder, date, part, supplier
where lo_orderdate = d_datekey
and lo_partkey = p_partkey
and lo_suppkey = s_suppkey
and p_brandl = 'MFGR#222I'
and s_region = 'EUROPE'
group by d_year, p_brandl
order by d_year, p_brandl;
```

- No Fastbit :

```
./ibis -d /bjin/ssb/index0 -v -l q23_fastbit.log -o q23_fastbit.txt -q "select
sum(lo_revenue), d_year, p_brandl where p_brandl = 'MFGR#222I'
and s_region = 'EUROPE'"
```

Consulta Q3.0

- No SGBD:

```
select c_nation, s_nation, d_year, sum(lo_revenue) as revenue
  from customer, lineorder, supplier, date
 where lo_custkey = c_custkey
 and lo_suppkey = s_suppkey
 and lo_orderdate = d_datekey
 and c_region = 'ASIA' and s_region = 'ASIA'
 and d_year >= 1992 and d_year <= 1997
 group by c_nation, s_nation, d_year
 order by d_year asc, revenue desc;
```

- No FastBit?

Consulta Q3.2

- No SGBD:

```
select c_city, s_city, d_year, sum(lo_revenue) as reve-nue from customer,  
       lineorder, supplier, date  
where lo_custkey = c_custkey  
and lo_suppkey = s_suppkey  
and lo_orderdate = d_datekey  
and c_nation = 'UNITED STATES'  
and s_nation = 'UNITED STATES'  
and d_year >= 1992 and d_year <= 1997  
group by c_city, s_city, d_year  
order by d_year asc, revenue desc;
```

- No Fastbit?

Consulta Q3.3

- No SGBD:

```
select c_city, s_city, d_year, sum(lo_revenue) as revenue from
customer, lineorder, supplier, date
where lo_custkey = c_custkey
and lo_suppkey = s_suppkey
and lo_orderdate = d_datekey
and (c_city='UNITED K11'
or c_city='UNITED K15')
and (s_city='UNITED K11'
or s_city='UNITED K15')
and d_year >= 1992 and d_year <= 1997
group by c_city, s_city, d_year
order by d_year asc, revenue desc;
```

Consulta Q3.4

- No SGBD:

```
select c_city, s_city, d_year, sum(lo_revenue) as revenue from
customer, lineorder, supplier, date
where lo_custkey = c_custkey
and lo_suppkey = s_suppkey
and lo_orderdate = d_datekey
and (c_city='UNITED K11' or
c_city='UNITED K15')
and (s_city='UNITED K11' or
s_city='UNITED K15')
and d_yearmonth = 'Dec1997'
group by c_city, s_city, d_year
order by d_year asc, revenue desc;
```

Consulta Q4.0

- No SGBD:

```
select d_year, c_nation, sum(lo_revenue - lo_supplycost) as  
profit from date, customer, supplier, part, lineorder where  
lo_custkey = c_custkey and lo_suppkey = s_suppkey and  
lo_partkey = p_partkey and lo_orderdate = d_datekey and  
c_region = 'AMERICA' and s_region = 'AMERICA' and  
(p_mfgr = 'MFGR#1' or p_mfgr = 'MFGR#2') group by  
d_year, c_nation order by d_year, c_nation
```


Consulta Q4.2

```
select d_year, s_nation, p_category, sum(lo_revenue -
    lo_supplycost) as profit
from date, customer, supplier, part, lineorder
where lo_custkey = c_custkey
and lo_suppkey = s_suppkey
and lo_partkey = p_partkey
and lo_orderdate = d_datekey
and c_region = 'AMERICA'
and s_region = 'AMERICA'
and (d_year = 1997 or d_year = 1998)
and (p_mfgr = 'MFGR#1'
or p_mfgr = 'MFGR#2')
group by d_year, s_nation, p_category order by d_year, s_nation,
    p_category
```

Consultas OLAP

- Roll-up?
- Drill-down?

Considere as hierarquias de atributos:

$p_mfgr < p_category < p_brand1 < p_partkey$

$c_region < c_nation < c_city < c_address$

$s_region < s_nation < s_city < s_address$

...

→ Modifique as colunas de agrupamento ou aquelas presentes na cláusula WHERE