



Bases de Dados

Linguagem SQL - DDL

Profa. Elaine Parros Machado de Sousa



Conteúdo

- Conceitos introdutórios
- Componentes do SGBD
- SQL
 - operadores
 - tipos de dado
 - DDL

Conceitos introdutórios:

- ***Data Definition Language*** (DDL)
 - conjunto de comandos para **definição do esquema** da base de dados
 - Exemplos em linguagem SQL
 - *create table*
 - *alter table*
 - *drop table*
- Compilador/Interpretador DDL



Conceitos introdutórios:

- **Dicionário de Dados:**
 - banco de dados do sistema
 - armazena descrição do esquema
 - armazena metadados
 - armazena restrições de segurança e integridade
 - outras denominações: **catálogo de dados, diretório de dados**

Conceitos introdutórios:

- ***Data Manipulation Language*** (DML)
 - Conjunto de comandos para manipulação dos dados (instância da base)
 - recuperação (consulta)
 - inserção
 - remoção
 - modificação
 - Viabiliza manipulação dos dados de maneira compatível com o modelo de dados

Conceitos introdutórios:

- ***Data Manipulation Language*** (DML)
 - Exemplos em linguagem SQL
 - *select*
 - *insert*
 - *delete*
 - *update*

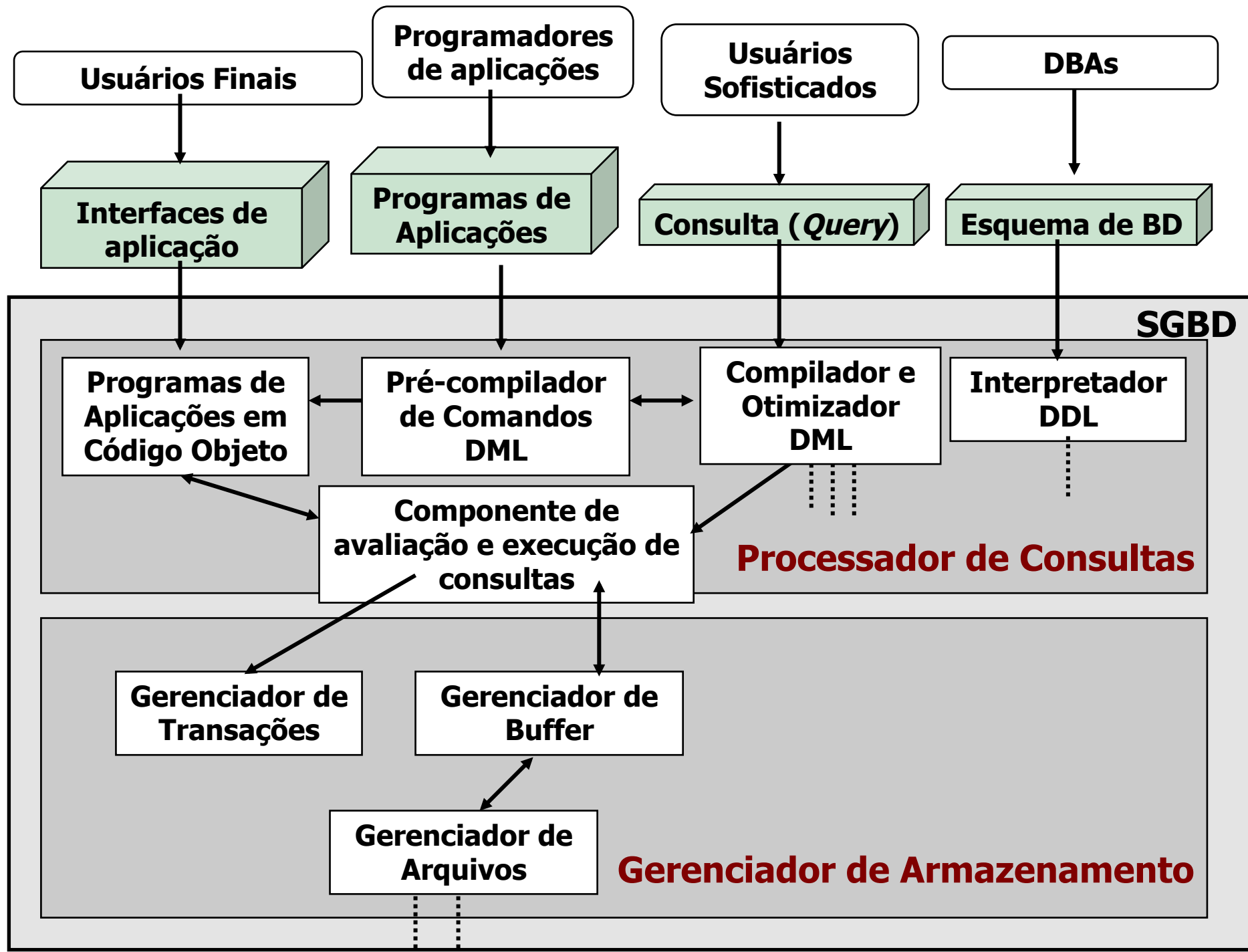
Conceitos introdutórios:

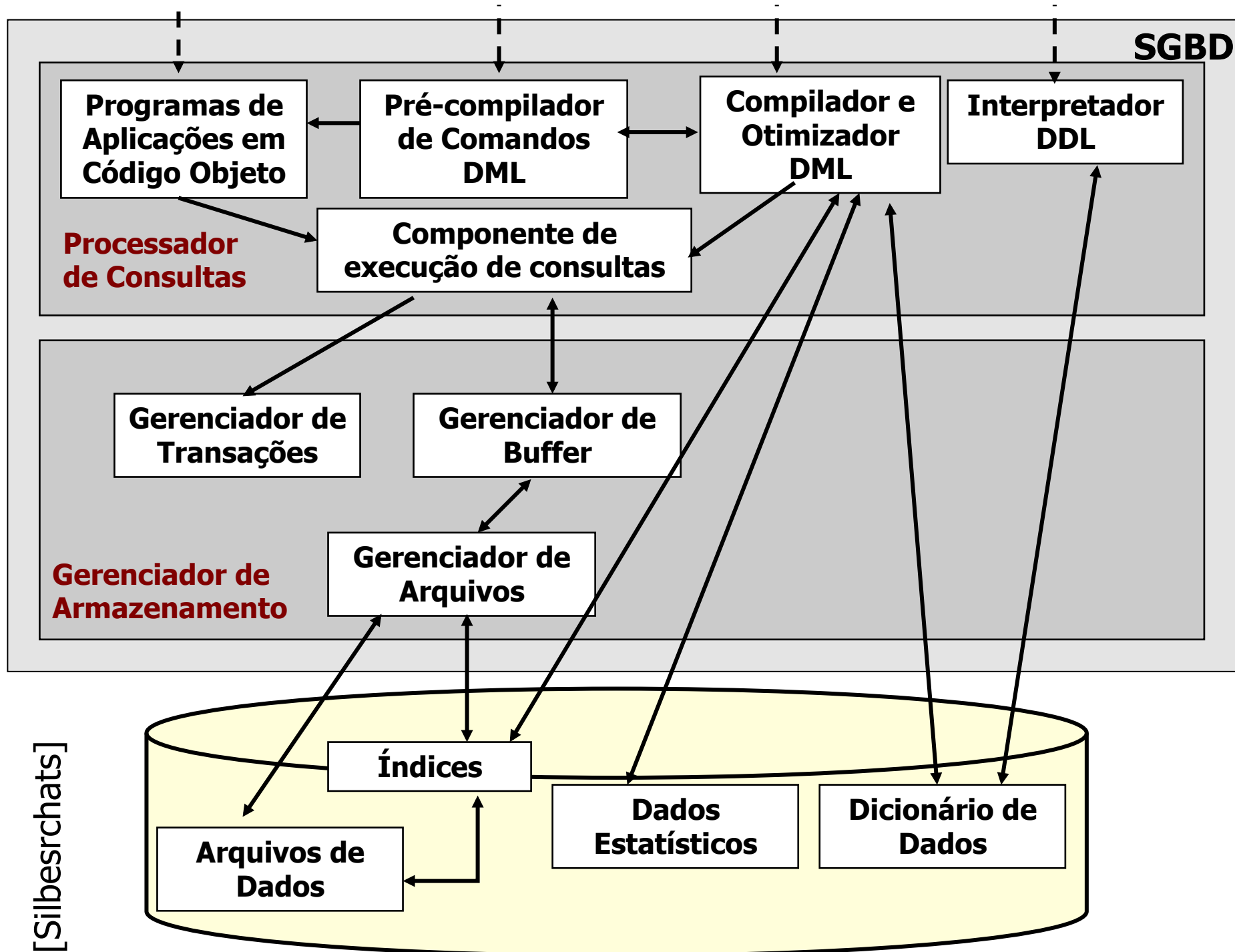
- Dois tipos de DML
 - **Procedural:** exige especificação de quais dados são necessários, e como obtê-los
 - requer uma sequência específica de operações a serem executadas
 - ex: *álgebra relacional*
 - **Não-Procedural (Declarativa):** exige apenas especificação de quais dados são necessários, e não de como obtê-los
 - ex: *SQL*

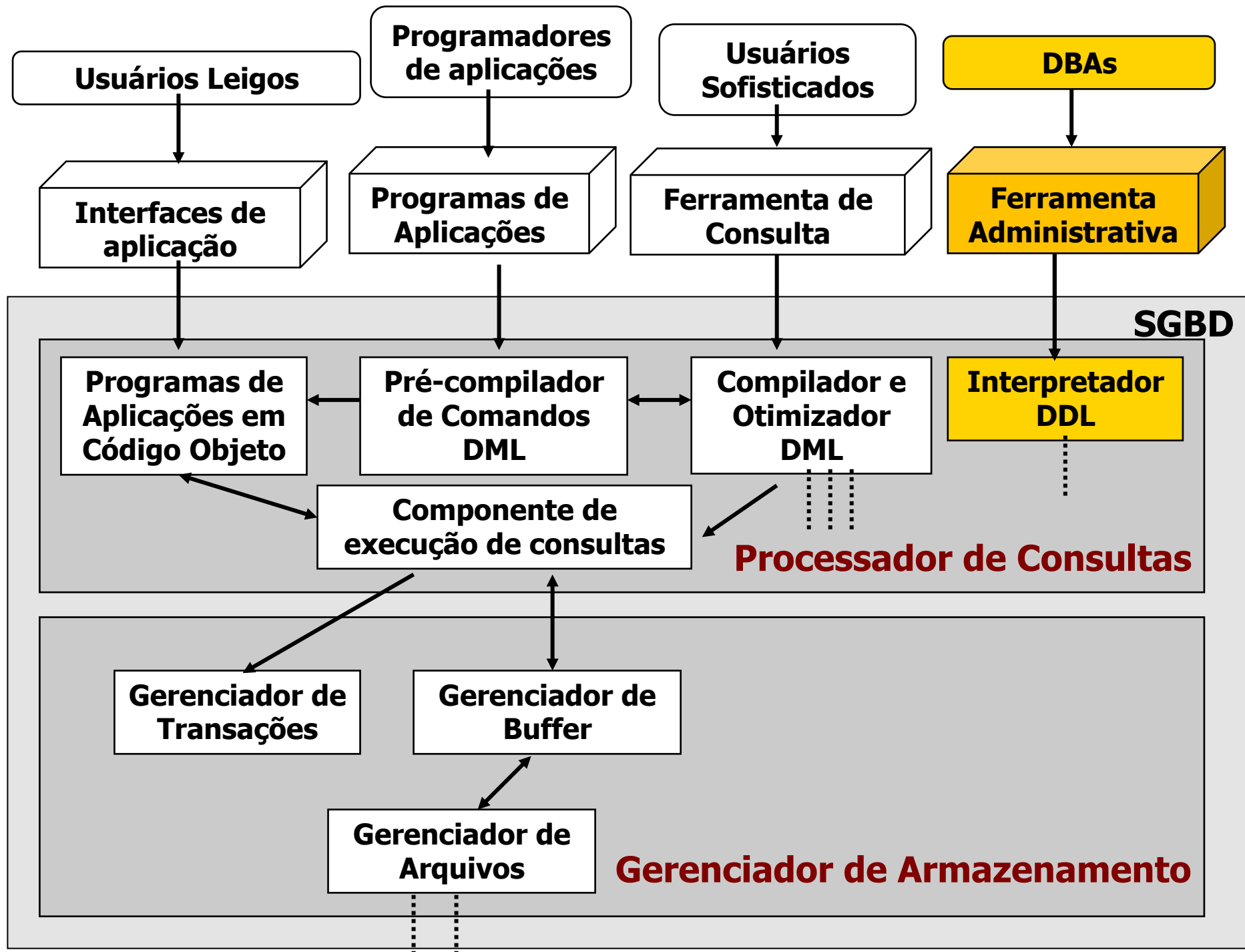
Componentes de um SGBD (Sistema de Gerenciamento de Bases de Dados)

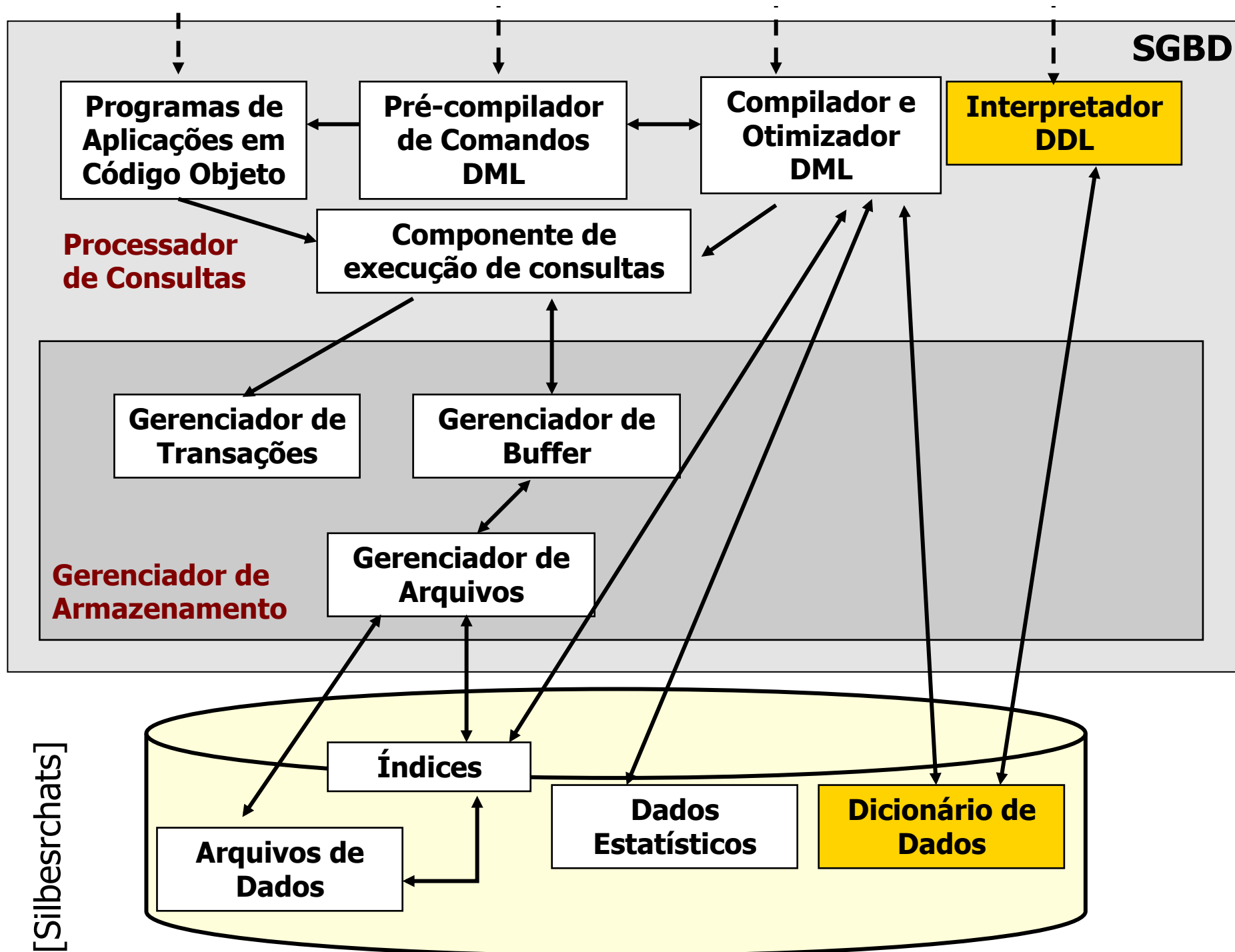


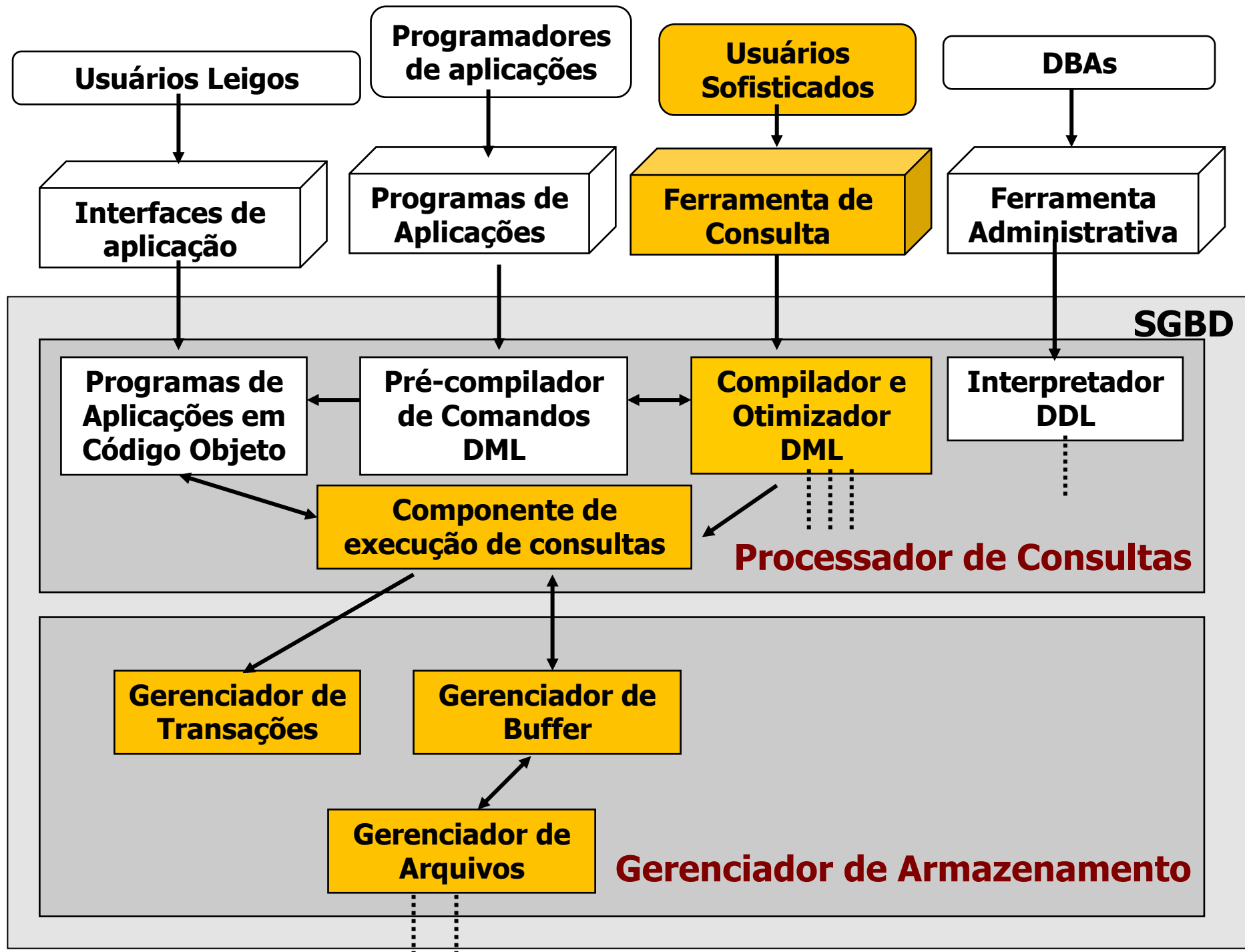
- Os componentes funcionais do SGBD:
 - componentes de processamento de consultas
 - componentes de gerenciamento de armazenamento

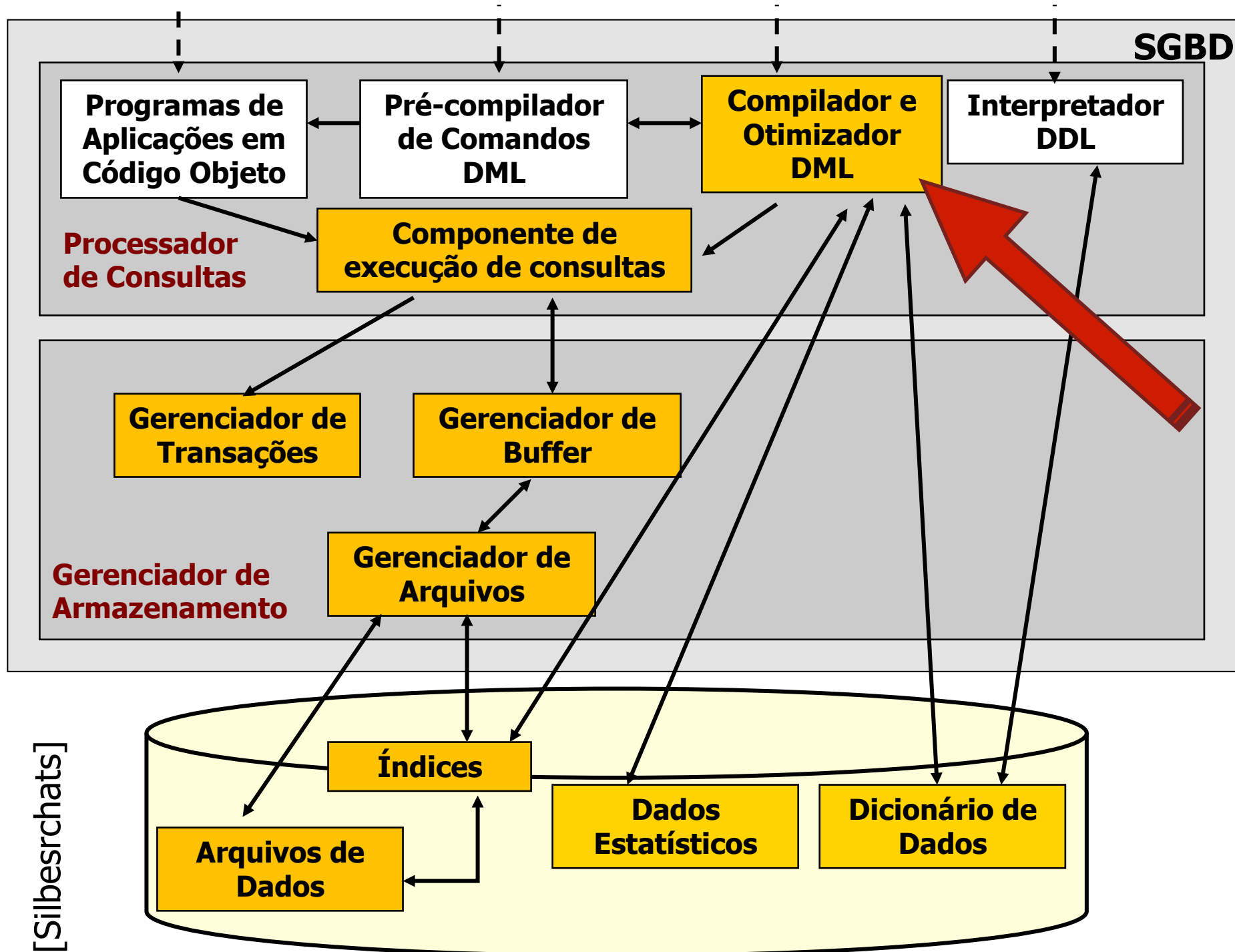


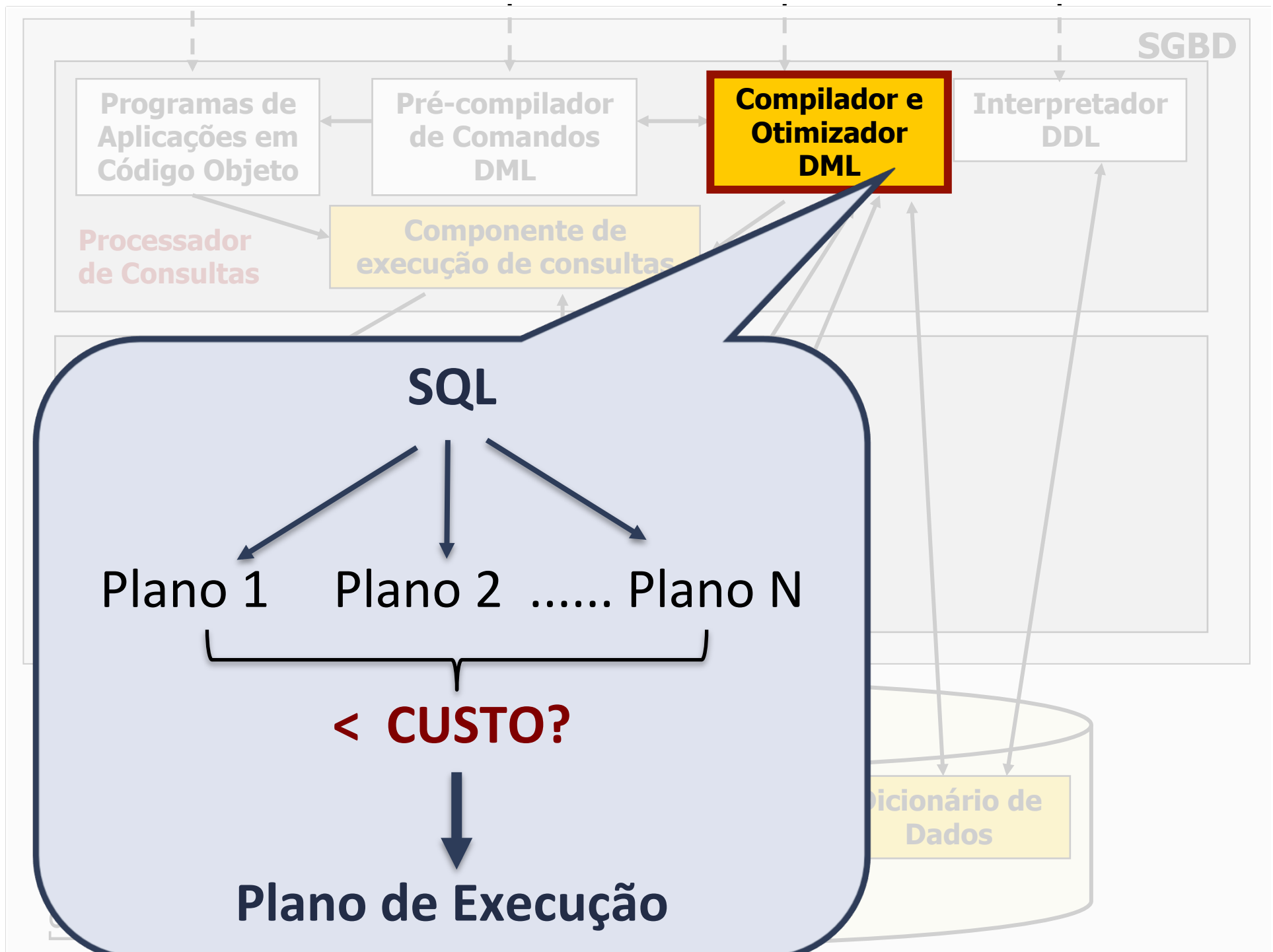


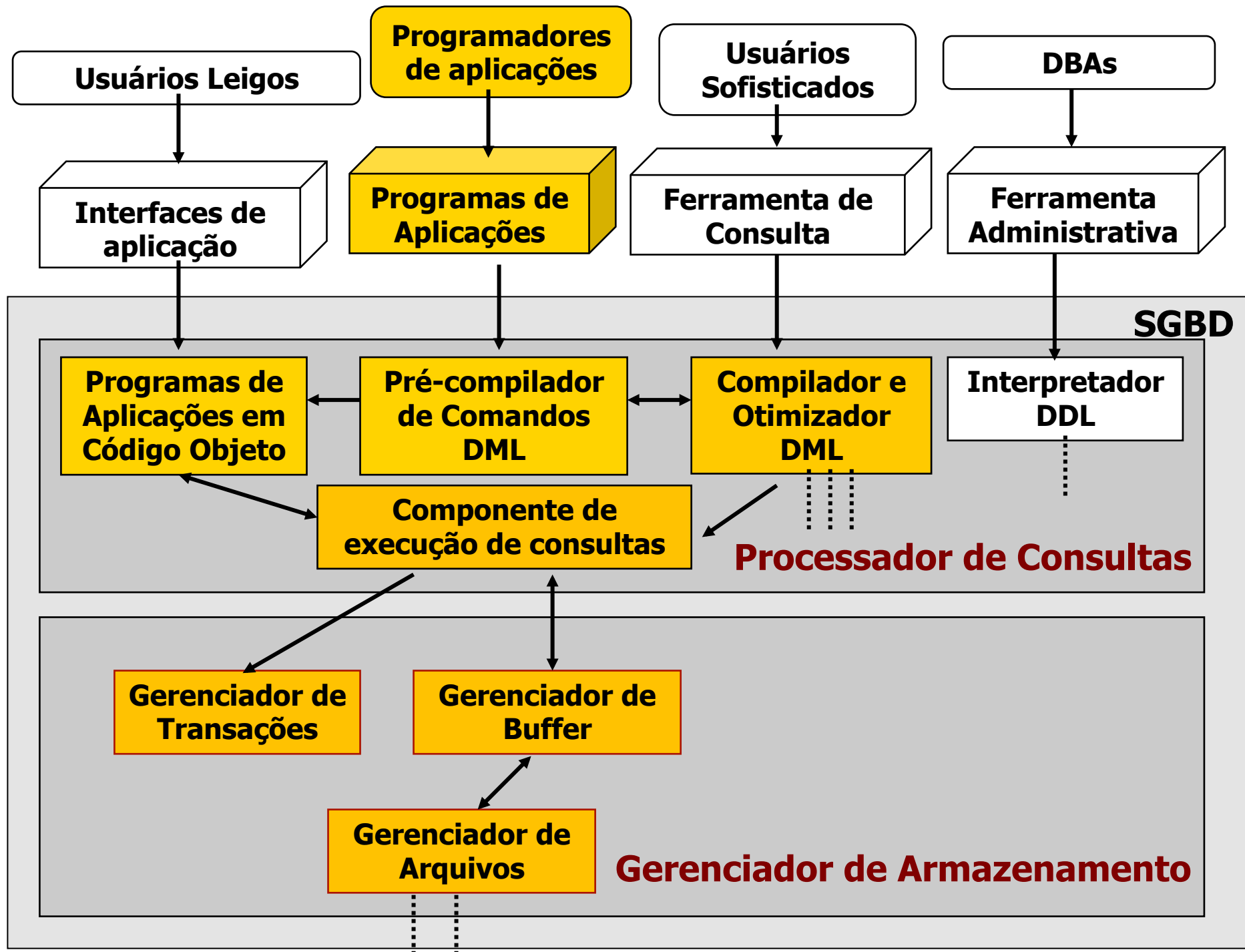


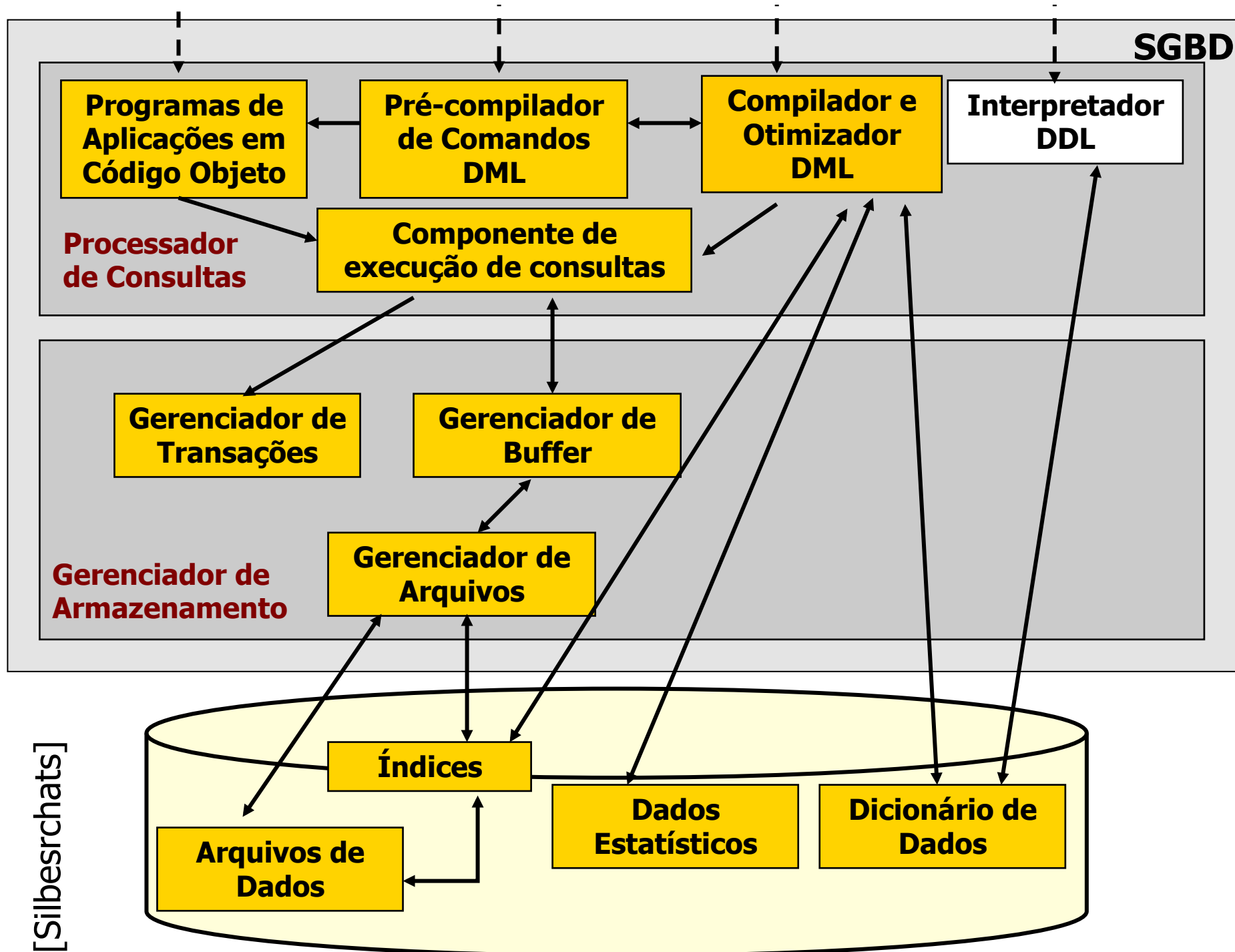


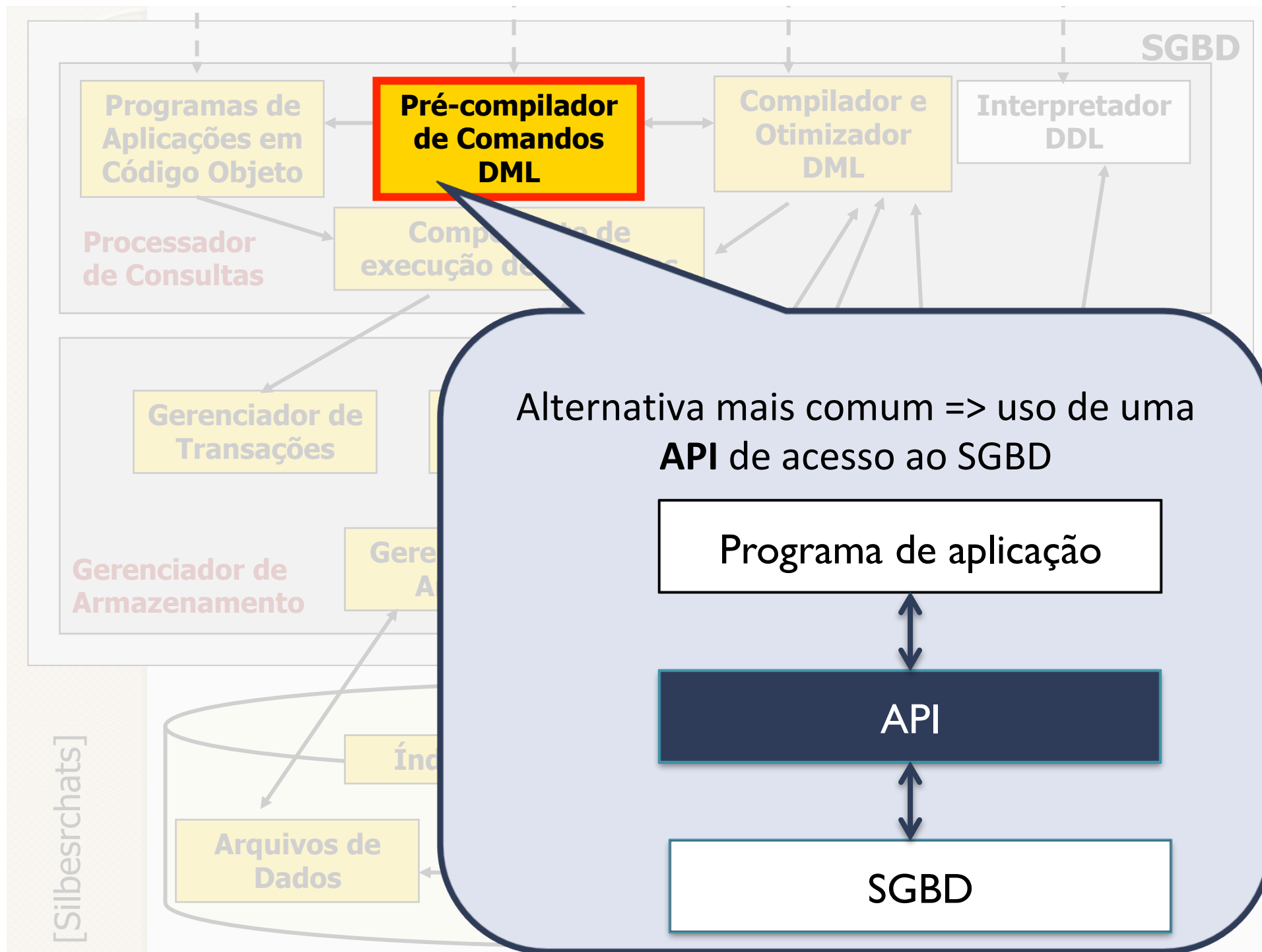


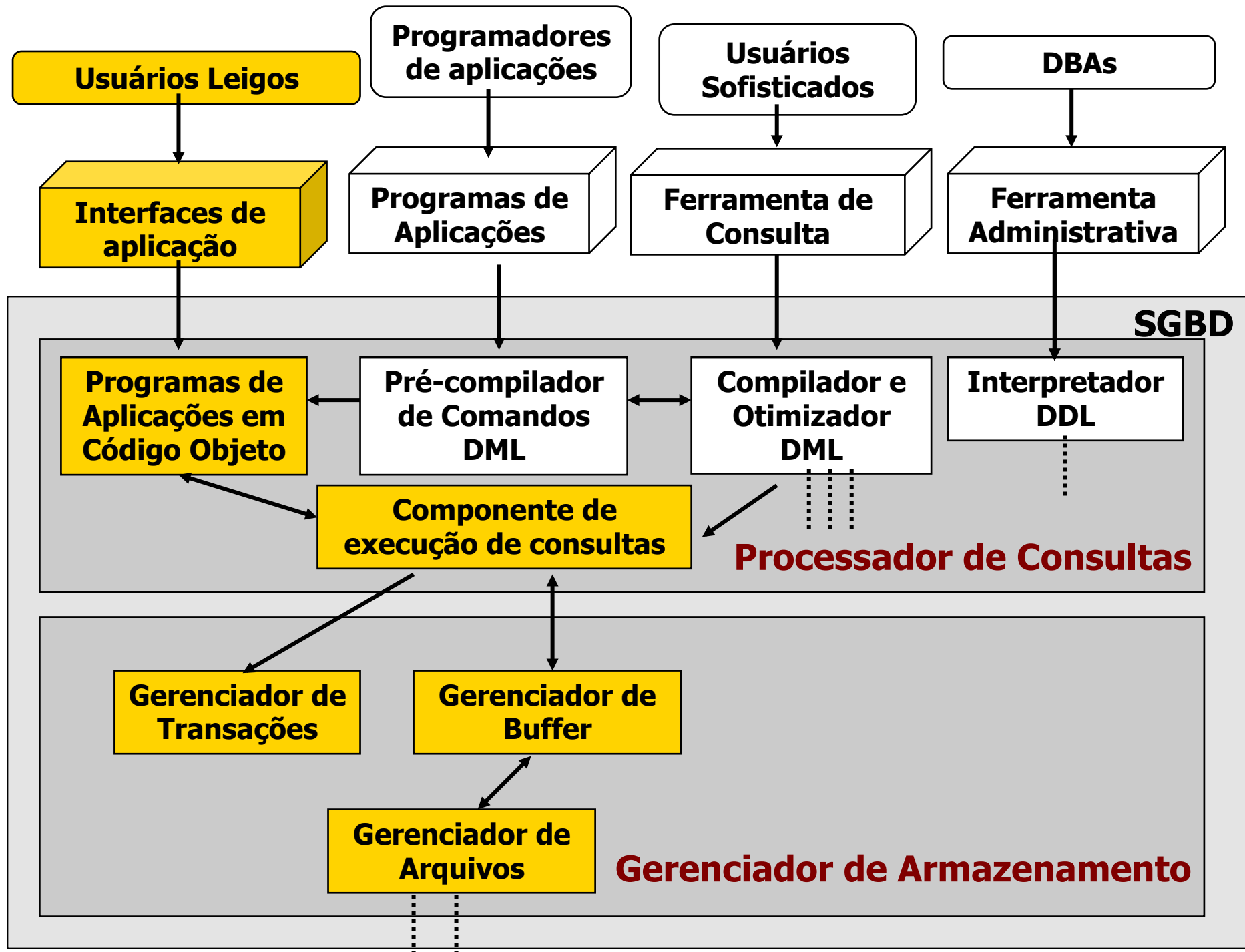


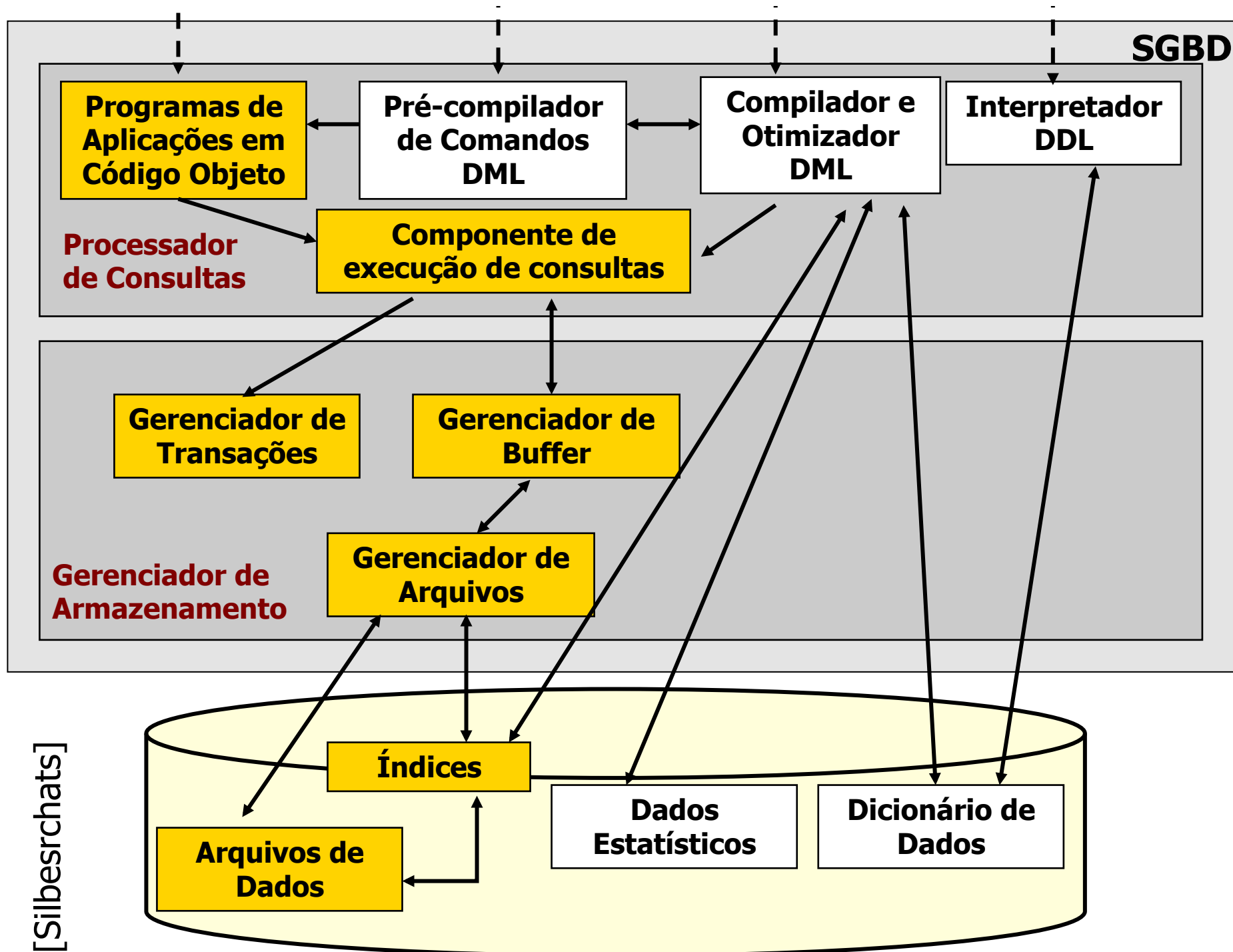














SQL

SQL - Introdução

- **Linguagem declarativa** \Rightarrow não procedural
- Desenvolvida e implementada pelo laboratório de pesquisa da **IBM** em San Jose – década de 70
- Inicialmente chamada SEQUEL (*Structured English QUery Language*)
- Criada como interface entre usuários e o primeiro SGBDR (**SYSTEM R**)

SQL - Introdução

- Padrão industrial que atinge grande parte do mercado de SGBDs
 - atrativos:
 - simplicidade
 - grande poder de consulta
 - padrão facilita migração

SQL – Introdução

O padrão SQL (ISO/IEC)

- **SQL 2016**

- *Polymorphic Table Functions*, suporte a JSON, reconhecimento de padrões em tuplas

- **SQL 2011**

- recursos para suporte a bases de dados temporais

- **SQL 2008**

- novos recursos de orientação a objetos

- **SQL 2006**

- SQL/XML

- **SQL 2003**

- SQL/XML

- **SQL 99 (SQL3)**

- conceitos de orientação a objetos

- **padrões anteriores (ANSI)**

- SQL92 – SQL2
- SQL86

SQL

- Recursos:
 - **DDL** – *Data Definition Language*
 - **DML** – *Data Manipulation Language*
 - definição de restrições de integridade
 - criação de visões (**views**)
 - especificações de segurança e autorizações (**DCL** – *Data Control Language*)
 - controle de transação (**DTL** – *Data Transaction Language*)
 - regras para integração com linguagens de programação
 -

SQL – Alguns Operadores...

- =, <, >, <=, >=, <>
- AND, OR, NOT
- *<atributo ou expressão>* BETWEEN *valor1* AND *valor2*
 - determina se o valor de um atributo ou de uma expressão se encontra dentro de um intervalo especificado de valores
- *<atributo ou expressão>* IS NULL
 - determina se o valor de um atributo ou de uma expressão é nulo

SQL – Alguns Operadores...

LIKE

- compara partes de uma sequência de caracteres

atributo **LIKE** '*%string%*'

% compara qualquer *substring* (número qualquer de caracteres)

atributo **LIKE** '*_string_ _*'

_ compara qualquer caractere

- comparação *case-sensitive*

SQL – Alguns Operadores...

- **<atributo ou expressão> IN <conjunto valores>**
 - verifica se o valor de um atributo ou de uma expressão é igual a algum dos valores em uma lista
- **<atributo ou expressão> NOT IN <conjunto valores>**
- ...

SQL – Alguns tipos de dado

- **INTEGER | SMALLINT | DOUBLE | FLOAT | REAL ...**
- **DECIMAL [(precision, scale)]**
 - *precision* - número total de dígitos
 - *scale* - número de dígitos depois do ponto
- **NUMBER (precision, scale)** ← tipo numérico ORACLE
- **CHAR(n)** - tamanho fixo - n caracteres
- **VARCHAR(n)** - tamanho variável ← ORACLE:
VARCHAR2
 - máximo de n caracteres
- **BLOB** – *Binary Large Object*
- **DATE | TIME | TIMESTAMP** ← ORACLE:
DATE
armazena data e hora
- ...



DDL

DDL - Introdução

- **Alguns comandos da DDL:**

- **CREATE TABLE**
- **ALTER TABLE**
- **DROP TABLE**

- **CREATE VIEW**
- **ALTER VIEW**
- **DROP VIEW**

- **CREATE PROCEDURE**
- **ALTER PROCEDURE**
- **DROP PROCEDURE**

- **CREATE INDEX**
- **ALTER INDEX**
- **DROP INDEX**

- **CREATE TRIGGER**
- **ALTER TRIGGER**
- **DROP TRIGGER**

- **CREATE USER**
- **ALTER USER**
- **DROP USER**

- ...

Comandos DDL

- **CREATE TABLE** - criar uma tabela, definir colunas e restrições

```
CREATE TABLE tabela (  
    atrib1 tipo [<restrições da coluna 1>],  
    atrib2 tipo [<restrições da coluna 2>],  
    ....  
    atribn tipo [<restrições da coluna n>],  
  
    [<restrições da tabela>]  
) ;
```


CREATE TABLE

- Restrições de colunas
 - NOT NULL
 - DEFAULT *valor*
 - CHECK (*condição*)

```
CREATE TABLE tabela (  
    atrib1 tipo [(tamanho)] [NOT NULL | DEFAULT valor]  
        [CHECK (condição)],  
    atrib2 tipo [(tamanho)] [NOT NULL | DEFAULT valor]  
        [CHECK (condição)],  
    ...  
);
```

CREATE TABLE

- Restrições de tabela

- CHECK (*condição*)

- PRIMARY KEY (*<atributos chave primária>*)

- UNIQUE (*<atributos chave secundária
(terciária,) >*)

- FOREIGN KEY (*<atributos chave estrangeira>*
REFERENCES *tabelaRef* [(*<chave primária>*)]
[*<ações>*]

<instrução> *<efeito>*

ON DELETE | ON UPDATE

CASCADE | SET NULL | SET DEFAULT


- CREATE TABLE

```
CREATE TABLE tabela (  
    atrib1 tipo [(tamanho)] [NOT NULL | DEFAULT valor]  
        [CHECK (condição)],  
    atrib2 tipo [(tamanho)] [NOT NULL | DEFAULT valor]  
        [CHECK (condição)],  
    ...  
    [CONSTRAINT nome da restrição]  
        PRIMARY KEY (<atributos chave primária>),  
    [CONSTRAINT nome da restrição]  
        UNIQUE (<atributos chave candidata>),  
    [CONSTRAINT nome da restrição]  
        FOREIGN KEY (<atributos chave estrangeira>)  
        REFERENCES tabelaRef [(chave primária>)]  
            [ON DELETE CASCADE | SET NULL | SET DEFAULT]  
            [ON UPDATE CASCADE | SET NULL | SET DEFAULT],  
    [CONSTRAINT nome da restrição]  
        CHECK (condição)  
);
```

ORACLE não
implementa
ON UPDATE



ORACLE não implementa
SET DEFAULT



Exemplo

- Criar as tabelas para o seguinte esquema:

Aluno = {Nusp, CPF, RG, UF, Nome, DataNasc}

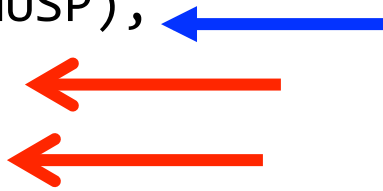
Professor = {Nome, NFunc, Titulação}

Disciplina = {Sigla, Nome, NCred, Prof}


Turma = {Sigla, Numero, NAlunos}

Matrícula = {Sigla, Numero, Aluno, Ano, Nota}

```
CREATE TABLE ALUNO (  
    NUSP NUMBER NOT NULL,  
    CPF CHAR(14),  
    RG CHAR(8),  
    UF CHAR(2),  
    NOME VARCHAR2(30) NOT NULL,  
    DATANASC DATE NOT NULL,  
    PRIMARY KEY (NUSP),  
    UNIQUE(CPF),  
    UNIQUE(RG, UF)  
);
```



```
CREATE TABLE PROFESSOR (  
    NFUNC NUMBER NOT NULL PRIMARY KEY,  
    NOME VARCHAR2(30) NOT NULL,  
    TITULACAO CHAR(7) NOT NULL,  
    CHECK (UPPER(TITULACAO) IN ('MESTRE', 'DOUTOR', 'TITULAR'))  
);
```



Professor = {Nome, NFunc, Titulação}

Disciplina = {Sigla, Nome, NCred, Prof}

```
CREATE TABLE DISCIPLINA (  
    SIGLA CHAR(6) NOT NULL,  
    NOME VARCHAR2(30) NOT NULL,  
    NCRED NUMBER NOT NULL,  
    PROF NUMBER ,  
    CONSTRAINT PK_DISCIPLINA PRIMARY KEY (SIGLA),  
    CONSTRAINT FK_DISCIPLINA FOREIGN KEY (PROF)  
        REFERENCES PROFESSOR(NFUNC)  
        ON DELETE SET NULL,  
    CONSTRAINT CK_NCREDITOS CHECK (NCRED > 0)  
);
```

Melhor colocar CONSTRAINT sempre!

Disciplina = {Sigla, Nome, NCred, Prof}

Turma = {Sigla, Numero, NAlunos}

```
CREATE TABLE TURMA (  
    SIGLA CHAR(6) NOT NULL,  
    NUMERO NUMBER NOT NULL,  
    NALUNOS NUMBER NOT NULL  
    CONSTRAINT CK_NALUNOS CHECK(NAlunos <= 70),  
    CONSTRAINT PK_TURMA PRIMARY KEY (SIGLA, NUMERO),  
    CONSTRAINT FK_TURMA FOREIGN KEY (SIGLA)  
        REFERENCES DISCIPLINA(SIGLA)  
        ON DELETE CASCADE  
);
```



E se colocarmos
ON DELETE SET NULL???

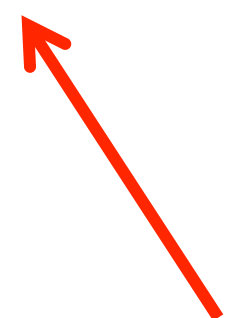
Aluno = {Nusp, ...}

Turma = {Sigla, Numero, NAlunos}

Matrícula = {Sigla, Numero, Aluno, Ano, Nota}



```
CREATE TABLE MATRICULA (  
    SIGLA CHAR(6) NOT NULL,  
    NUMERO NUMBER NOT NULL,  
    ALUNO NUMBER NOT NULL,  
    ANO NUMBER(4) NOT NULL,  
    NOTA NUMBER,  
    CONSTRAINT PK_MATRICULA  
        PRIMARY KEY (SIGLA, NUMERO, ALUNO, ANO),  
    CONSTRAINT FK1_MATRICULA FOREIGN KEY (SIGLA, NUMERO)  
        REFERENCES TURMA(SIGLA, NUMERO)  
        ON DELETE CASCADE,  
    CONSTRAINT FK2_MATRICULA FOREIGN KEY (ALUNO)  
        REFERENCES ALUNO(NUSP)  
        ON DELETE CASCADE  
);
```



Comandos DDL

- **ALTER TABLE** – incluir/alterar/remover definições de colunas e restrições

ALTER TABLE *tabela* <ação>;

■ <ação>:

- ADD *novoAtrib tipo* [<restrições de coluna>]
- ADD [CONSTRAINT *nome*] <restrição de tabela>
- DROP *atributo* [CASCADE | RESTRICT]
- DROP CONSTRAINT *nome*
- ALTER *atributo* DROP DEFAULT
- ALTER *atributo* SET DEFAULT <valor>

... .

ALTER TABLE

- **ADD** *novoStrib* *tipo* [*<restrições de coluna>*]
 - e os valores do novo atributo nas tuplas já existentes?
- **DROP** *atributo* [**CASCADE** | **RESTRICT**]
 - **CASCADE** – todas as visões e restrições (*constraints*) que referenciam o atributo são removidas automaticamente
 - **RESTRICT** – atributo só é removido se não houver nenhuma visão ou restrição que o referencie



ORACLE remove apenas as restrições (além do atributo)

Aluno = {Nusp, CPF, RG, UF, Nome, DataNasc}

Professor = {Nome, NFunc, Titulação}

Disciplina = {Sigla, Nome, NCred, Prof}

Turma = {Sigla, Numero, NAlunos}

Matrícula = {Sigla, Numero, Aluno, Ano, Nota}

- Modificar as tabelas para:
 - 1) acrescentar o atributo Cidade de Origem para a tabela Aluno com default 'Sao Carlos';
 - 2) remover o atributo Numero da tabela Turma
 - como fica a tabela matrícula?
 - 3) acrescentar uma restrição para que a nota de um aluno seja sempre maior ou igual a zero
 - 4) remover uma restrição da tabela Matricula
 - 5) definir o valor *default* 'Sanca' para a cidade de origem dos alunos

ORACLE:

- DROP COLUMN
- CASCADE CONSTRAINTS
- não tem **RESTRICT** – é o *default*

1) alter table Aluno add CidadeOrigem varchar2(30)
default 'Sao Carlos';

2) alter table Turma drop Numero restrict;

Turma = {Sigla, Numero, NAlunos}

Matrícula = {Sigla, Numero, Aluno, Ano, Nota}



alter table Turma drop Numero cascade;

Turma = {Sigla, NAlunos}

Matrícula = {Sigla, **Numero**, Aluno, Ano, Nota}

3) alter table Matricula **add constraint**
ck_nota **check** (nota > 0);

4) alter table Matricula **drop constraint** ck_nota;

5) alter table Aluno **alter** CidadeOrigem
set default 'Sanca';



ORACLE:

```
- alter table Aluno  
    modify (CidadeOrigem default 'Sanca');
```

Comandos DDL

- **DROP TABLE** - exclui uma tabela da base de dados

DROP TABLE *tabela* [**CASCADE** | **RESTRICT**];

- **CASCADE**: todas as visões e restrições que referenciam a tabela são removidas automaticamente
- **RESTRICT**: a tabela é removida somente se não for referenciada em nenhuma restrição ou visão



ORACLE remove apenas as restrições (além da tabela)

Exercício

- Modificar o esquema para:
 - remover a tabela Turma
 - como fica a tabela matrícula?

Aluno = {Nusp, CPF, RG, UF, Nome, DataNasc}

Professor = {Nome, NFunc, Titulação}

Disciplina = {Sigla, Nome, NCred, Prof}

Turma = {Sigla, Numero, NAlunos}

Matrícula = {Sigla, Numero, Aluno, Ano, Nota}

Leitura recomendada

- R. Elmasri, S. Navathe: *Sistemas de Banco de Dados*
 - 4ª Edição
 - Capítulos 8 e 9
 - 6ª Edição
 - Capítulos 4 e 5
- A. Silberschatz, H. F. Korth, s. Sudarshan: *Sistema de Banco de Dados*
 - Capítulo 4