

# Aplicações - SQL

## Banco de Dados: Teoria e Prática

André Santanchè  
Instituto de Computação - UNICAMP  
Agosto 2011



Estes *slides* são concedidos sob uma Licença Creative Commons.  
Veja detalhes e restrições adicionais no último slide.

# SQL

- SQL - Structured Query Language
- Originalmente: SEQUEL - Structured English QUERy Language
- Criada pela IBM Research
  - Interface BD Relacional → SYSTEM R

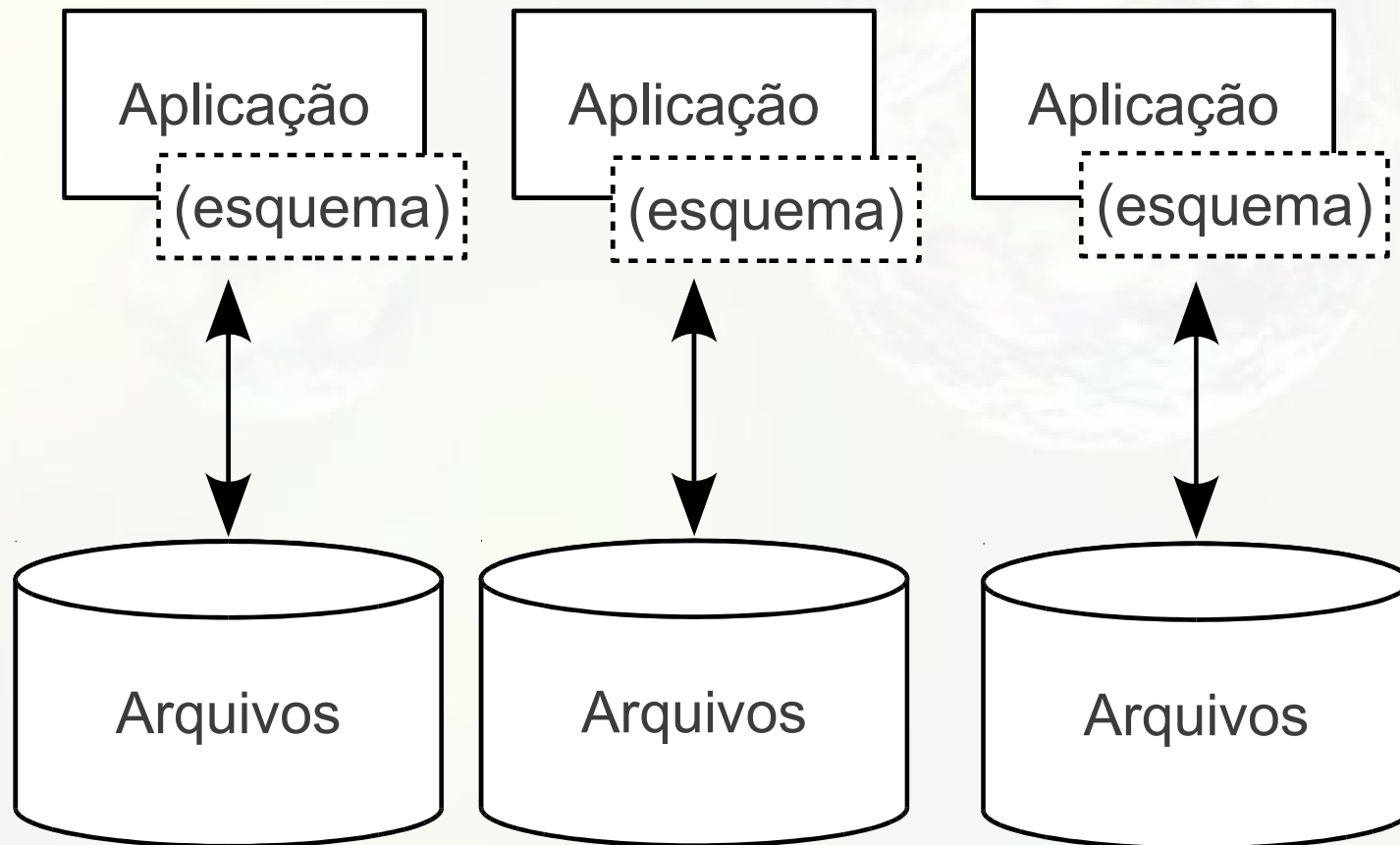
# SQL

## Padronização

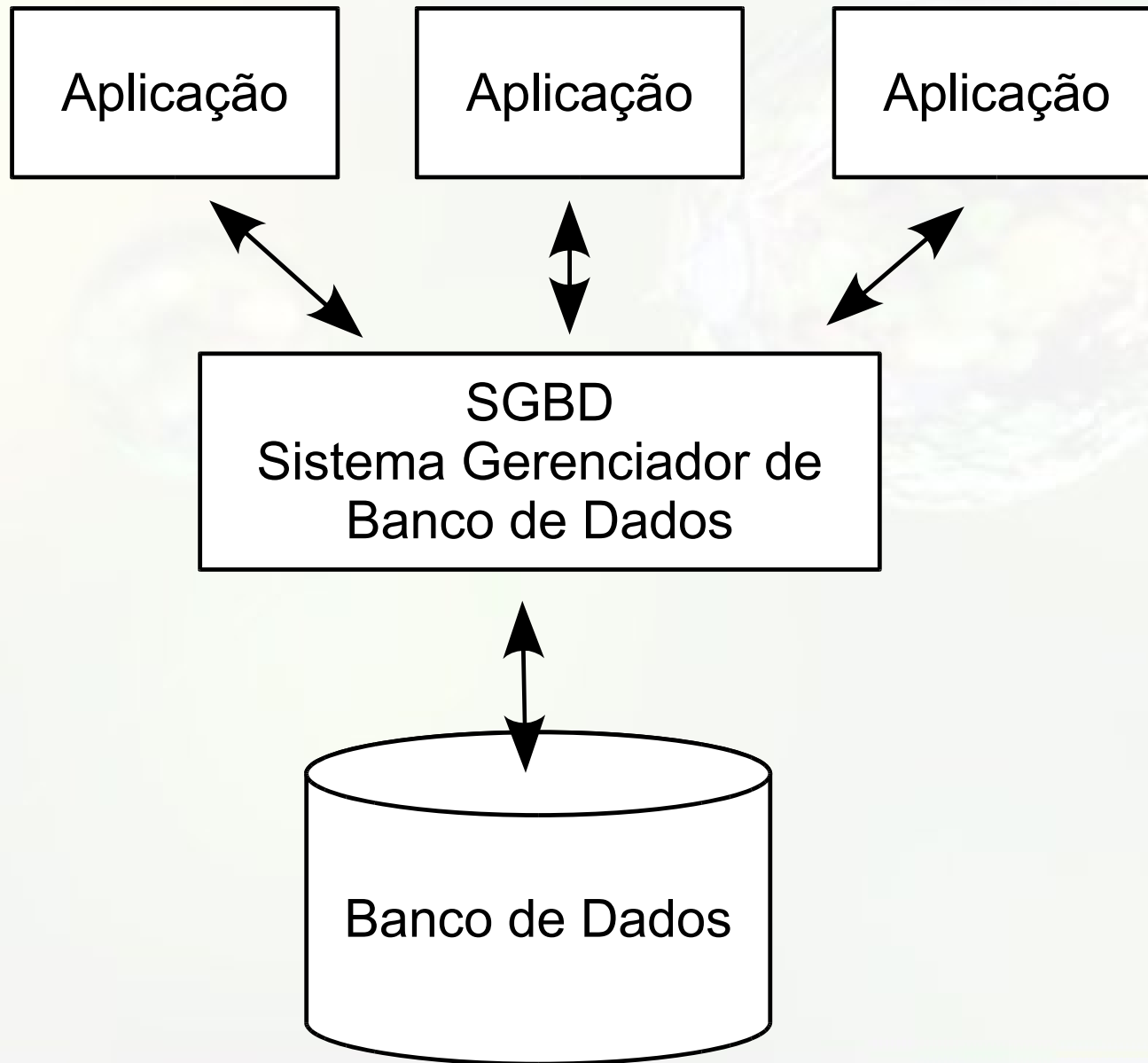
- ANSI + ISO
- SQL-86 ou SQL1
- SQL-92 ou SQL2
- SQL:1999 ou SQL3
- SQL:2003
- SQL:2006

# Aplicações e Armazenamento

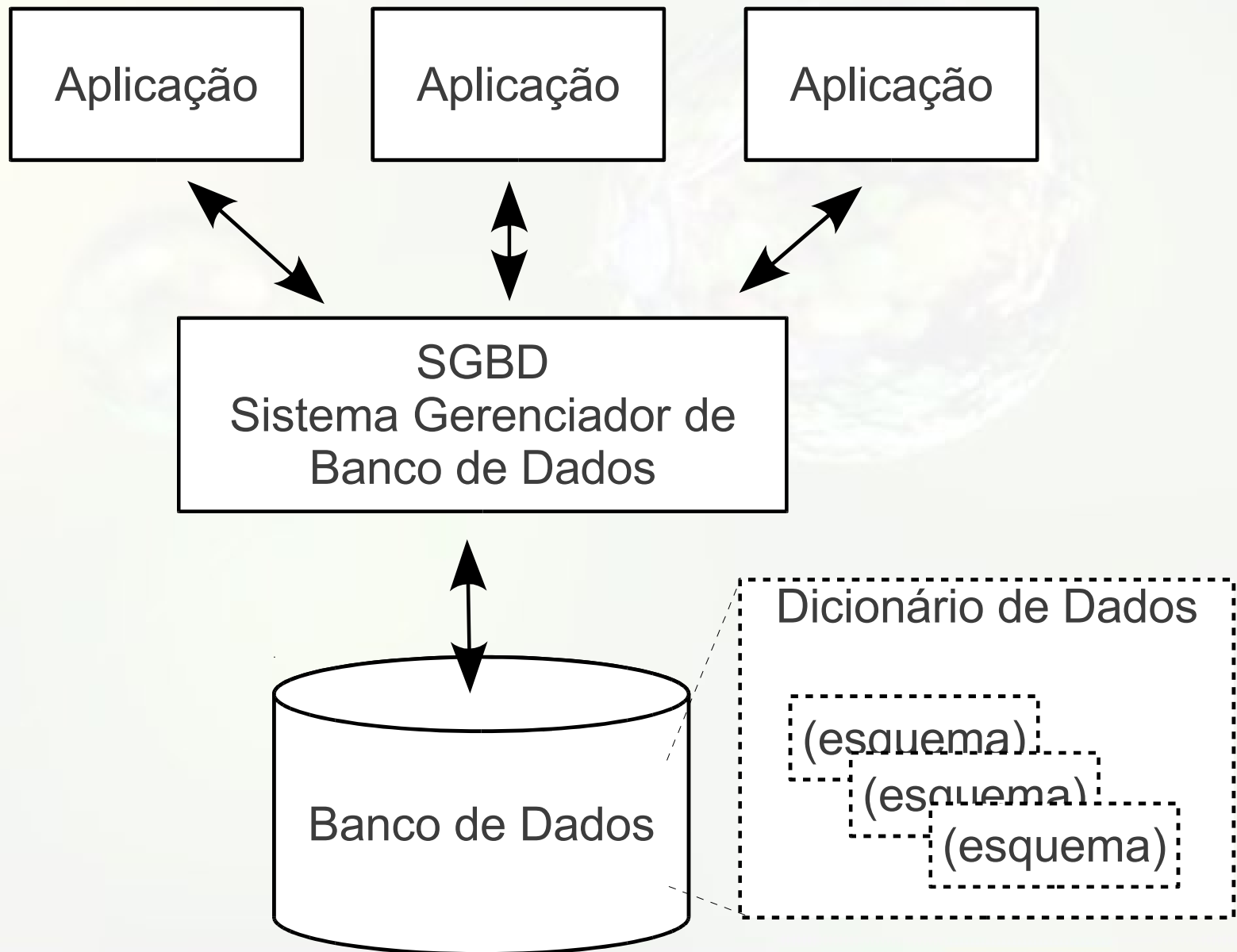
## Arquivos



# Aplicações e Armazenamento SGBD



# Dicionário de Dados



# CREATE SCHEMA

- CREATE SCHEMA <esquema>  
AUTHORIZATION <id\_autorizado>
- executeUpdate(...)

# CREATE TABLE

- CREATE TABLE <tabela>  
    (<campo<sub>1</sub>> <tipo> [NULL|NOT NULL] [restrição],  
    [... ,  
    <campo<sub>n</sub>> <tipo> [NULL|NOT NULL] [restrição],  
    PRIMARY KEY <chave\_primaria>])
- executeUpdate(...)



# CREATE TABLE

- CREATE TABLE Taxi2 (  
    Placa VARCHAR(7) NOT NULL,  
    Marca VARCHAR(30) NOT NULL,  
    Modelo VARCHAR(30) NOT NULL,  
    AnoFab INTEGER,  
    Licenca VARCHAR(9),  
    PRIMARY KEY(Placa)  
)

# INSERT

- INSERT INTO <tabela>  
[(<campo<sub>1</sub>>[,..., <campo<sub>n</sub>>])]  
VALUES ( <valor<sub>1</sub>>[,..., <valor<sub>n</sub>>])
- executeUpdate(...)

# SELECT

- `SELECT * | <campo1>[,..., <campon>]  
FROM <tabela1>[,..., <tabelan>]  
WHERE <condição/junção>`
- `executeQuery(...)`

# SELECT LIKE

- SELECT ...  
FROM <tabela<sub>1</sub>>[,..., <tabela<sub>n</sub>>]  
WHERE <condição/junção>
- % → qualquer cadeia com 0 a n caracteres
- \_ → exatamente um caractere (qualquer)
- = → caractere de escape
  - e.g., serve para encontrar um caractere \_

# AS (alias)

- SELECT <campo<sub>1</sub>> [AS] <alias<sub>1</sub>>  
[,..., <campo<sub>n</sub>> [AS] <alias<sub>n</sub>>]  
...
- SELECT ...  
FROM <tabela<sub>1</sub>> [AS] <alias<sub>1</sub>>  
[,..., <tabela<sub>n</sub>> [AS] <alias<sub>n</sub>>]  
...

# SELECT DISTINCT e ALL

- SELECT DISTINCT ...
- SELECT ALL ...
- A cláusula ALL é implícita se não especificada

# SELECT ORDER BY

- SELECT ...  
ORDER BY <campo<sub>1</sub>>[, ..., <campo<sub>n</sub>>]

# DELETE

- DELETE FROM <tabela<sub>1</sub>>  
WHERE <condição>
- executeUpdate(...))



# UPDATE

- UPDATE <tabela>  
    SET <campo<sub>1</sub>>=<valor<sub>1</sub>>  
        [,..., <campo<sub>n</sub>>=<valor<sub>n</sub>>]  
    WHERE <condição>
- executeUpdate(...)

# Estudo de Caso SQL

- UPDATE Marcadores  
SET Categoria = <nova>  
WHERE Categoria = <antiga>
- UPDATE Taxonomia  
SET Categoria = <nova>  
WHERE Categoria = <antiga>
- UPDATE Taxonomia  
SET Superior = <nova>  
WHERE Superior = <antiga>

# Prepared Statement

# Utilizando o PreparedStatement

- `SELECT FROM Marcadores  
WHERE Titulo = ?`
- `<comando>.setString(<numero>, <valor>)`

# Utilizando o PreparedStatement

- INSERT INTO Marcadores  
VALUES ( ? , ? , ? , ? )
- <comando>.setString(<numero>, <valor>)
- <comando>.setInt(<numero>, <valor>)

# Utilizando o PreparedStatement

- UPDATE Marcadores  
    SET Categoria = ?  
    WHERE Categoria = ?
- <comando>.setString(<numero>, <valor>)
- <comando>.setInt(<numero>, <valor>)

# Agrupamento

# GROUP BY

- **SELECT \* | <campo<sub>1</sub>>[,..., <campo<sub>n</sub>>]  
FROM <tabela<sub>1</sub>>[,..., <tabela<sub>n</sub>>]  
WHERE <condição/junção>  
GROUP BY <coluna\_agrupar>  
HAVING <condição\_grupo>**



# Funções de Agregação

- COUNT(\*) ⇒ contagem
- SUM(<coluna>) ⇒ soma
- AVG(<coluna>) ⇒ média
- MAX(<coluna>) ⇒ maior valor
- MIN(<coluna>) ⇒ menor valor

# Visões

# VIEW

- CREATE VIEW <nome> AS  
SELECT ...

# Consultas Aninhadas

# SELECT IN e NOT IN

- SELECT ...  
WHERE <campo> IN  
(SELECT <campo> ...)
- SELECT ...  
WHERE <campo> NOT IN  
(SELECT <campo> ...)

# SELECT EXISTS e NOT EXISTS

- SELECT ...  
WHERE EXISTS  
(SELECT <campo> ...)
- SELECT ...  
WHERE NOT EXISTS  
(SELECT <campo> ...)

# SELECT Comparação

- SELECT ...  
WHERE <campo> <comparação>  
(SELECT <campo> ...)

# Outer Join



# Outer Join

- SELECT ...  
FROM <tabela> <join> <tabela>  
ON <condição> ...
- <join>
  - JOIN
  - NATURAL JOIN
  - LEFT JOIN
  - RIGHT JOIN
  - FULL JOIN

# União, Interseção e Diferença

- SELECT ...  
    <operador>  
    SELECT ...
- <operador>
  - UNION
  - INTERSECT
  - EXCEPT

[santanche@ic.unicamp.br](mailto:santanche@ic.unicamp.br)

# Licença

- Estes slides são concedidos sob uma Licença Creative Commons. Sob as seguintes condições: Atribuição, Uso Não-Comercial e Compartilhamento pela mesma Licença, com restrições adicionais:
  - Se você é estudante, você não está autorizado a utilizar estes slides (total ou parcialmente) em uma apresentação na qual você esteja sendo avaliado, a não ser que o professor que está lhe avaliando:
    - lhe peça explicitamente para utilizar estes slides;
    - ou seja informado explicitamente da origem destes slides e concorde com o seu uso.
- Mais detalhes sobre a referida licença Creative Commons veja no link:  
<http://creativecommons.org/licenses/by-nc-sa/2.5/br/>

# Aplicações - SQL

## Banco de Dados: Teoria e Prática

André Santanchè  
Instituto de Computação - UNICAMP  
Agosto 2011



Estes *slides* são concedidos sob uma [Licença Creative Commons](#).  
Veja detalhes e restrições adicionais no último slide.

# SQL

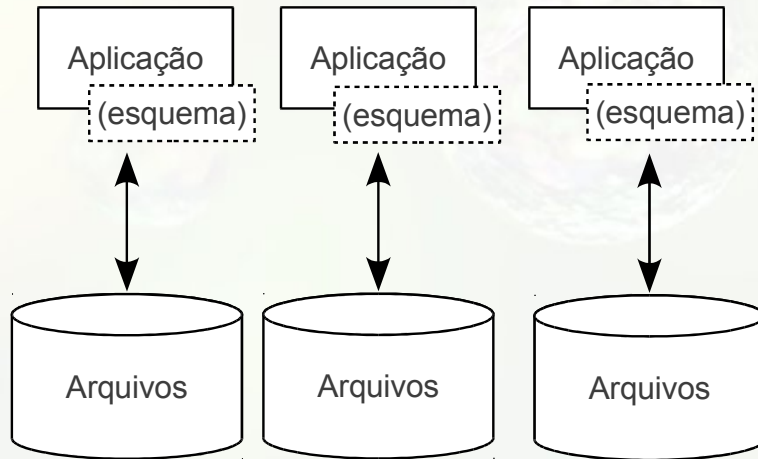
- SQL - Structured Query Language
- Originalmente: SEQUEL - Structured English QUery Language
- Criada pela IBM Research
  - Interface BD Relacional → SYSTEM R

# SQL

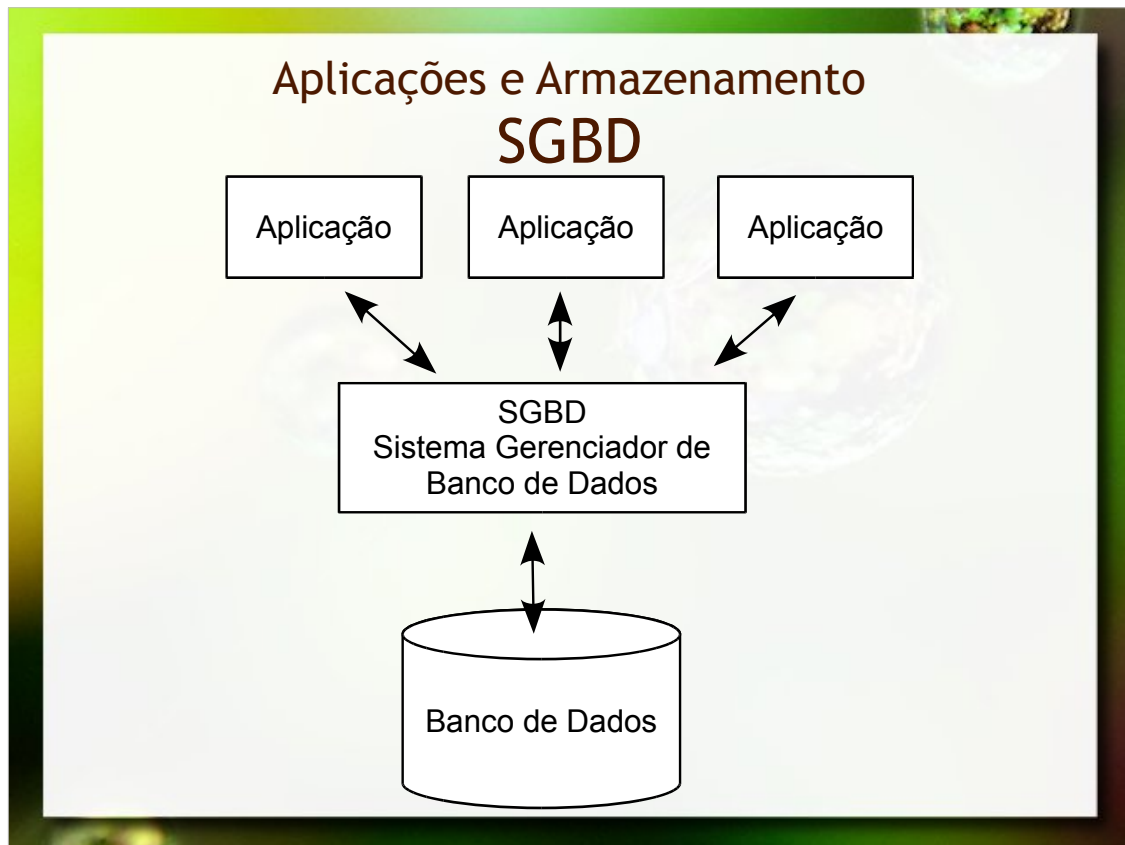
## Padronização

- ANSI + ISO
- SQL-86 ou SQL1
- SQL-92 ou SQL2
- SQL:1999 ou SQL3
- SQL:2003
- SQL:2006

## Aplicações e Armazenamento Arquivos



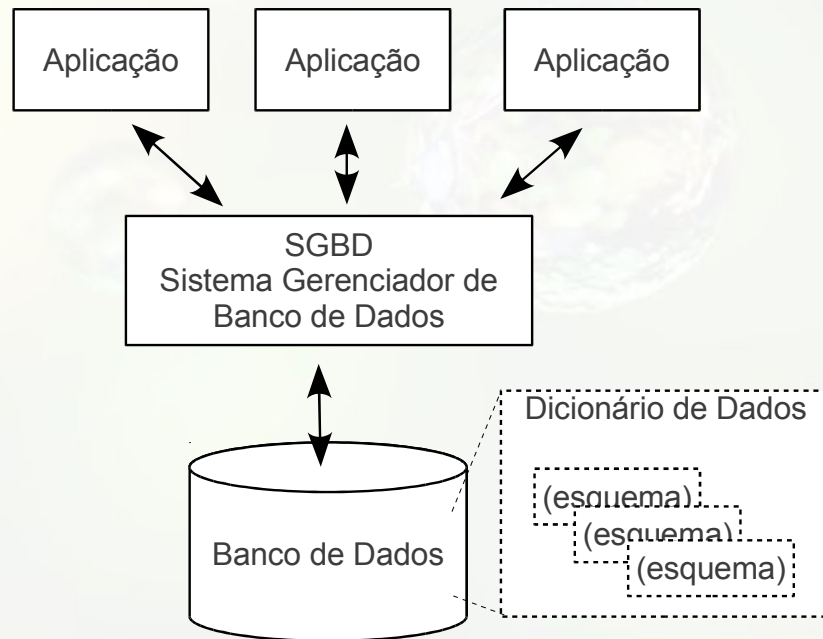




“Database System:

- The DBMS software together with the data itself. Sometimes, the applications are also included” (Elmasri, 2007, c. 1, s. 5)

# Dicionário de Dados



## CREATE SCHEMA

- CREATE SCHEMA <esquema>  
AUTHORIZATION <id\_autorizado>
- executeUpdate(...)

## CREATE TABLE

- CREATE TABLE <tabela>  
(<campo<sub>1</sub>> <tipo> [NULL|NOT NULL] [restrição],  
[...,  
<campo<sub>n</sub>> <tipo> [NULL|NOT NULL] [restrição],  
PRIMARY KEY <chave\_primaria>])
- executeUpdate(...)

## CREATE TABLE

```
▫ CREATE TABLE Taxi2 (  
    Placa VARCHAR(7) NOT NULL,  
    Marca VARCHAR(30) NOT NULL,  
    Modelo VARCHAR(30) NOT NULL,  
    AnoFab INTEGER,  
    Licenca VARCHAR(9),  
    PRIMARY KEY(Placa)  
)
```

# INSERT

- INSERT INTO <tabela>  
[(<campo<sub>1</sub>>[,..., <campo<sub>n</sub>>])]  
VALUES ( <valor<sub>1</sub>>[,..., <valor<sub>n</sub>>])
- executeUpdate(...)

# SELECT

- `SELECT * | <campo1>[,..., <campon>]  
FROM <tabela1>[,..., <tabelan>]  
WHERE <condição/junção>`
- `executeQuery(...)`

## SELECT LIKE

- SELECT ...  
FROM <tabela<sub>1</sub>>[,..., <tabela<sub>n</sub>>]  
WHERE <condição/junção>
- % → qualquer cadeia com 0 a n caracteres
- \_ → exatamente um caractere (qualquer)
- = → caractere de escape
  - e.g., serve para encontrar um caractere \_



## AS (alias)

- SELECT <campo<sub>1</sub>> [AS] <alias<sub>1</sub>>  
    [,..., <campo<sub>n</sub>> [AS] <alias<sub>n</sub>>]  
    ...
- SELECT ...  
    FROM <tabela<sub>1</sub>> [AS] <alias<sub>1</sub>>  
        [,..., <tabela<sub>n</sub>> [AS] <alias<sub>n</sub>>]  
    ...

## SELECT DISTINCT e ALL

- SELECT DISTINCT ...
- SELECT ALL ...
- A cláusula ALL é implícita se não especificada

## SELECT ORDER BY

- SELECT ...  
ORDER BY <campo<sub>1</sub>>[,..., <campo<sub>n</sub>>]

# DELETE

- DELETE FROM <tabela<sub>1</sub>>  
WHERE <condição>
- executeUpdate(...))

# UPDATE

- UPDATE <tabela>  
    SET <campo<sub>1</sub>>=<valor<sub>1</sub>>  
        [,..., <campo<sub>n</sub>>=<valor<sub>n</sub>>]  
    WHERE <condição>
- executeUpdate(...)

## Estudo de Caso SQL

- UPDATE Marcadores  
    SET Categoria = <nova>  
    WHERE Categoria = <antiga>
- UPDATE Taxonomia  
    SET Categoria = <nova>  
    WHERE Categoria = <antiga>
- UPDATE Taxonomia  
    SET Superior = <nova>  
    WHERE Superior = <antiga>



## Prepared Statement

## Utilizando o PreparedStatement

- `SELECT FROM Marcadores  
WHERE Titulo = ?`
- `<comando>.setString(<numero>, <valor>)`



## Utilizando o PreparedStatement

- INSERT INTO Marcadores  
VALUES ( ? , ? , ? , ? )
- <comando>.setString(<numero>, <valor>)
- <comando>.setInt(<numero>, <valor>)

## Utilizando o PreparedStatement

- UPDATE Marcadores  
SET Categoria = ?  
WHERE Categoria = ?
- <comando>.setString(<numero>, <valor>)
- <comando>.setInt(<numero>, <valor>)



# Agrupamento

## GROUP BY

```
□ SELECT * | <campo1>[,..., <campon>]  
  FROM <tabela1>[,..., <tabelan>]  
  WHERE <condição/junção>  
  GROUP BY <coluna_agrupar>  
  HAVING <condição_grupo>
```

## Funções de Agregação

- COUNT(\*) ⇒ contagem
- SUM(<coluna>) ⇒ soma
- AVG(<coluna>) ⇒ média
- MAX(<coluna>) ⇒ maior valor
- MIN(<coluna>) ⇒ menor valor



# Visões

## VIEW

- CREATE VIEW <nome> AS  
SELECT ...



## Consultas Aninhadas



## SELECT IN e NOT IN

- SELECT ...  
WHERE <campo> IN  
(SELECT <campo> ...)
- SELECT ...  
WHERE <campo> NOT IN  
(SELECT <campo> ...)

## SELECT EXISTS e NOT EXISTS

- SELECT ...  
WHERE EXISTS  
(SELECT <campo> ...)
- SELECT ...  
WHERE NOT EXISTS  
(SELECT <campo> ...)

## SELECT Comparação

- SELECT ...  
WHERE <campo> <comparação>  
(SELECT <campo> ...)



Outer Join

## Outer Join

- SELECT ...  
FROM <tabela> <join> <tabela>  
ON <condição> ...
- <join>
  - JOIN
  - NATURAL JOIN
  - LEFT JOIN
  - RIGHT JOIN
  - FULL JOIN

## União, Interseção e Diferença

- SELECT ...  
    <operador>  
    SELECT ...
- <operador>
  - UNION
  - INTERSECT
  - EXCEPT

[santanche@ic.unicamp.br](mailto:santanche@ic.unicamp.br)

# Licença

- Estes slides são concedidos sob uma Licença Creative Commons. Sob as seguintes condições: Atribuição, Uso Não-Comercial e Compartilhamento pela mesma Licença, com restrições adicionais:
  - Se você é estudante, você não está autorizado a utilizar estes slides (total ou parcialmente) em uma apresentação na qual você esteja sendo avaliado, a não ser que o professor que está lhe avaliando:
    - lhe peça explicitamente para utilizar estes slides;
    - ou seja informado explicitamente da origem destes slides e concorde com o seu uso.
- Mais detalhes sobre a referida licença Creative Commons veja no link:  
<http://creativecommons.org/licenses/by-nc-sa/2.5/br/>