

# Aula 1

## Linguagem SQL



# Linguagem SQL

## **Introdução à linguagem SQL**

- Arquitetar um banco de dados capaz de atender às necessidades especificadas.
- Desenvolver rotinas de definição, manipulação e recuperação de dados.
- Garantir a integridade dos dados armazenados utilizando restrições estruturais e funcionais.
- Criar relatórios para análise e consolidação das informações armazenadas.

- Conhecimento das regras de mapeamento dos modelos lógico/conceitual para o físico.
- Conhecimento da sublinguagem SQL de definição de dados (DDL): Estruturas e regras de armazenamento.
- Conhecimento da sublinguagem SQL de manipulação de dados (DML): Manipulação de dados.
- Conhecimento da sublinguagem SQL de pesquisa de dados (DQL): Consulta de dados.

# Bibliografia

## Bibliografia Básica (até 3 livros, padrão ABNT)

CORONEL, C.; MORRIS, S. **Database Systems: design, implementation and management**. 11.ed. São Paulo: Cengage do Brasil, 2014.

DATE, C.J. **SQL e Teoria Relacional: como escrever códigos SQL precisos**. 1.ed. São Paulo: Novatec, 2015.

PRATT, P.; LAST, M. **Concepts of Database Management**. 8. Ed. Boston: Cengage, 2014.

## Bibliografia Complementar (demais obras utilizadas)

ELMASRI, R. E.; NAVATHE, S. B. **Sistemas de Banco de Dados**. 6. Ed. São Paulo: Pearson, 2011.

SILBERCHATZ, A.; KORTH, H. F. **Sistema de Banco de Dados**. 6.ed. Rio de Janeiro: Elsevier, 2012.

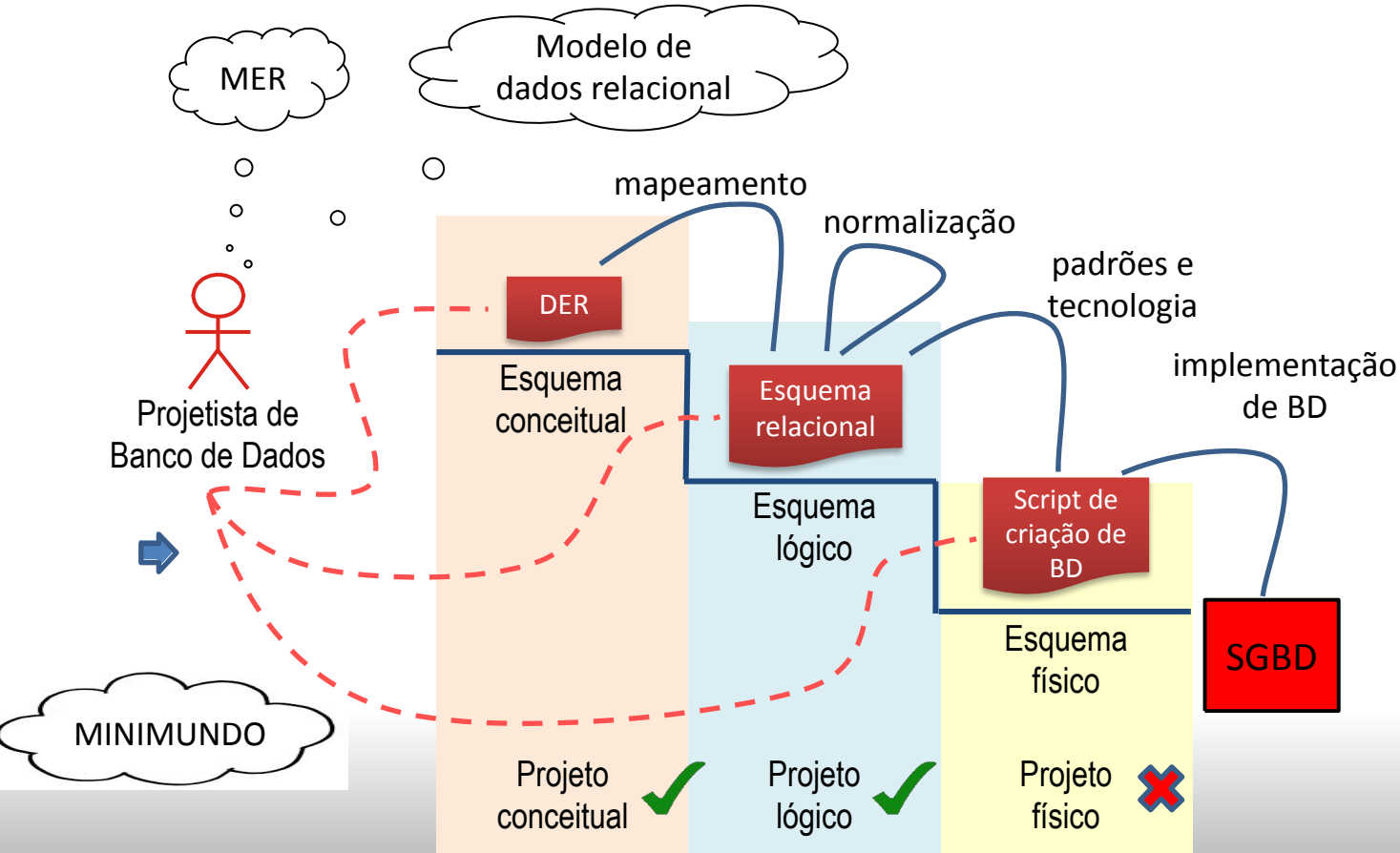
TEOREY, T. J.; LIGHSTONE, S.; NADEAU, T. **Projeto e Modelagem de Banco de Dados**. 2.ed. Rio de Janeiro: Elsevier, 2014.

VIESCAS, J. L.; HERNANDEZ, M.J. **SQL Queries for mere mortals: A hands-on guide to data manipulation in SQL**. 3rd. Ed. Upper Sadle River: Addison-Wesley, 2014.

DATE, C. J.; FERNANDES, A. **Projeto de Banco de Dados e Teoria Relacional**. 1. ed. São Paulo: Novatec, 2015.

- Modelo físico
  - Tipagem
  - Nulidade
  - Constraints
  - Atributos autoincrementais
- Sintaxe da sublinguagem DDL
- Exemplos

# Projeto de banco de dados



- Bancos de dados associam tipos de dados a colunas, expressões, variáveis e parâmetros.
- Tipos de dados determinam quais tipos de valores serão permitidos no armazenamento.



Todos os dados são armazenados nos bancos de dados em formato de Bytes. Essa é a forma como os computadores trabalham, ou seja, ao armazenar a letra A, armazenam o código binário “01000001”, que a representa.

**A** → **01000001**

# Tipos de dados

Lower Table

| Dec | Hex | Char | Dec | Hex | Char | Dec | Hex | Char | Dec | Hex | Char |
|-----|-----|------|-----|-----|------|-----|-----|------|-----|-----|------|
| 0   | 00  |      | 32  | 20  |      | 64  | 40  | @    | 96  | 60  | `    |
| 1   | 01  | ☉    | 33  | 21  | !    | 65  | 41  | A    | 97  | 61  | a    |
| 2   | 02  | ☼    | 34  | 22  | "    | 66  | 42  | B    | 98  | 62  | b    |
| 3   | 03  | ▼    | 35  | 23  | #    | 67  | 43  | C    | 99  | 63  | c    |
| 4   | 04  | +    | 36  | 24  | \$   | 68  | 44  | D    | 100 | 64  | d    |
| 5   | 05  | ♣    | 37  | 25  | %    | 69  | 45  | E    | 101 | 65  | e    |
| 6   | 06  | ♠    | 38  | 26  | &    | 70  | 46  | F    | 102 | 66  | f    |
| 7   | 07  | •    | 39  | 27  | '    | 71  | 47  | G    | 103 | 67  | g    |
| 8   | 08  | ▣    | 40  | 28  | (    | 72  | 48  | H    | 104 | 68  | h    |
| 9   | 09  | ○    | 41  | 29  | )    | 73  | 49  | I    | 105 | 69  | i    |
| 10  | 0A  | ⊗    | 42  | 2A  | *    | 74  | 4A  | J    | 106 | 6A  | j    |
| 11  | 0B  | ♠    | 43  | 2B  | +    | 75  | 4B  | K    | 107 | 6B  | k    |
| 12  | 0C  | ♣    | 44  | 2C  | ,    | 76  | 4C  | L    | 108 | 6C  | l    |
| 13  | 0D  | ♠    | 45  | 2D  | -    | 77  | 4D  | K    | 109 | 6D  | m    |
| 14  | 0E  | ♠    | 46  | 2E  | .    | 78  | 4E  | N    | 110 | 6E  | n    |
| 15  | 0F  | ☉    | 47  | 2F  | /    | 79  | 4F  | O    | 111 | 6F  | o    |
| 16  | 10  | ▶    | 48  | 30  | 0    | 80  | 50  | P    | 112 | 70  | p    |
| 17  | 11  | ◀    | 49  | 31  | 1    | 81  | 51  | Q    | 113 | 71  | q    |
| 18  | 12  | !    | 50  | 32  | 2    | 82  | 52  | R    | 114 | 72  | r    |
| 19  | 13  | !!   | 51  | 33  | 3    | 83  | 53  | S    | 115 | 73  | s    |
| 20  | 14  | ⌘    | 52  | 34  | 4    | 84  | 54  | T    | 116 | 74  | t    |
| 21  | 15  | \$   | 53  | 35  | 5    | 85  | 55  | U    | 117 | 75  | u    |
| 22  | 16  | -    | 54  | 36  | 6    | 86  | 56  | V    | 118 | 76  | v    |
| 23  | 17  | 1    | 55  | 37  | 7    | 87  | 57  | W    | 119 | 77  | w    |
| 24  | 18  | 1    | 56  | 38  | 8    | 88  | 58  | X    | 120 | 78  | x    |
| 25  | 19  | 1    | 57  | 39  | 9    | 89  | 59  | Y    | 121 | 79  | y    |
| 26  | 1A  | 1    | 58  | 3A  | :    | 90  | 5A  | Z    | 122 | 7A  | z    |
| 27  | 1B  | 1    | 59  | 3B  | ;    | 91  | 5B  | [    | 123 | 7B  | {    |
| 28  | 1C  | L    | 60  | 3C  | <    | 92  | 5C  | \    | 124 | 7C  |      |
| 29  | 1D  | 1    | 61  | 3D  | =    | 93  | 5D  | ]    | 125 | 7D  | }    |
| 30  | 1E  | ▲    | 62  | 3E  | >    | 94  | 5E  | ^    | 126 | 7E  | ~    |
| 31  | 1F  | ▼    | 63  | 3F  | ?    | 95  | 5F  | _    | 127 | 7F  | ◊    |

Upper Table

| Dec | Hex | Char | Dec | Hex | Char | Dec | Hex | Char | Dec | Hex | Char |
|-----|-----|------|-----|-----|------|-----|-----|------|-----|-----|------|
| 128 | 80  | Ç    | 160 | A0  | à    | 192 | C0  | Ì    | 224 | E0  | α    |
| 129 | 81  | ü    | 161 | A1  | á    | 193 | C1  | Í    | 225 | E1  | β    |
| 130 | 82  | é    | 162 | A2  | â    | 194 | C2  | Î    | 226 | E2  | Γ    |
| 131 | 83  | â    | 163 | A3  | û    | 195 | C3  | Ï    | 227 | E3  | π    |
| 132 | 84  | ä    | 164 | A4  | ü    | 196 | C4  | —    | 228 | E4  | Σ    |
| 133 | 85  | ä    | 165 | A5  | ÿ    | 197 | C5  | þ    | 229 | E5  | σ    |
| 134 | 86  | Å    | 166 | A6  | •    | 198 | C6  | ÿ    | 230 | E6  | μ    |
| 135 | 87  | ç    | 167 | A7  | °    | 199 | C7  | ÿ    | 231 | E7  | τ    |
| 136 | 88  | è    | 168 | A8  | ¿    | 200 | C8  | ÿ    | 232 | E8  | φ    |
| 137 | 89  | é    | 169 | A9  | ¸    | 201 | C9  | ÿ    | 233 | E9  | θ    |
| 138 | 8A  | ê    | 170 | AA  | ¸    | 202 | CA  | ÿ    | 234 | EA  | Ω    |
| 139 | 8B  | ÿ    | 171 | AB  | ¸    | 203 | CB  | ÿ    | 235 | EB  | δ    |
| 140 | 8C  | ÿ    | 172 | AC  | ¸    | 204 | CC  | ÿ    | 236 | EC  | ω    |
| 141 | 8D  | ÿ    | 173 | AD  | ÿ    | 205 | CD  | ÿ    | 237 | ED  | φ    |
| 142 | 8E  | ÿ    | 174 | AE  | «    | 206 | CE  | ÿ    | 238 | EE  | e    |
| 143 | 8F  | ÿ    | 175 | AF  | »    | 207 | CF  | ÿ    | 239 | EF  | ñ    |
| 144 | 90  | ÿ    | 176 | BO  | ÿ    | 208 | DO  | ÿ    | 240 | FO  | ≡    |
| 145 | 91  | æ    | 177 | B1  | ÿ    | 209 | D1  | ÿ    | 241 | F1  | ±    |
| 146 | 92  | æ    | 178 | B2  | ÿ    | 210 | D2  | ÿ    | 242 | F2  | ≥    |
| 147 | 93  | ó    | 179 | B3  | ÿ    | 211 | D3  | ÿ    | 243 | F3  | ≤    |
| 148 | 94  | ó    | 180 | B4  | ÿ    | 212 | D4  | ÿ    | 244 | F4  | [    |
| 149 | 95  | ó    | 181 | B5  | ÿ    | 213 | D5  | ÿ    | 245 | F5  | ]    |
| 150 | 96  | ü    | 182 | B6  | ÿ    | 214 | D6  | ÿ    | 246 | F6  | ÷    |
| 151 | 97  | ü    | 183 | B7  | ÿ    | 215 | D7  | ÿ    | 247 | F7  | ≈    |
| 152 | 98  | ÿ    | 184 | B8  | ÿ    | 216 | D8  | ÿ    | 248 | F8  | °    |
| 153 | 99  | ÿ    | 185 | B9  | ÿ    | 217 | D9  | ÿ    | 249 | F9  | •    |
| 154 | 9A  | ÿ    | 186 | BA  | ÿ    | 218 | DA  | ÿ    | 250 | FA  | •    |
| 155 | 9B  | ó    | 187 | BB  | ÿ    | 219 | DB  | ÿ    | 251 | FB  | √    |
| 156 | 9C  | é    | 188 | BC  | ÿ    | 220 | DC  | ÿ    | 252 | FC  | ≈    |
| 157 | 9D  | ÿ    | 189 | BD  | ÿ    | 221 | DD  | ÿ    | 253 | FD  | •    |
| 158 | 9E  | ÿ    | 190 | BE  | ÿ    | 222 | DE  | ÿ    | 254 | FE  | ■    |
| 159 | 9F  | ÿ    | 191 | BF  | ÿ    | 223 | DF  | ÿ    | 255 | FF  | ■    |

Exemplo de armazenamento em Bytes - Tabela ASCII

Tipos de dados determinam quais tipos de valores serão permitidos no armazenamento e os principais tipos são agrupados em categorias, conforme mostrado abaixo:

|                       |                   |
|-----------------------|-------------------|
| Numéricos exatos      | Caractere Unicode |
| Numéricos aproximados | Binários          |
| Data e Hora           | Outros tipos      |
| Strings de caractere  |                   |

- Numéricos exatos

| Tipo de dado    | Intervalo   | Armazenamento<br>o<br>(bytes) |
|-----------------|---|-------------------------------|
| tinyint         | 0 to 255  | 1                             |
| smallint        | -32,768 to 32,767   | 2                             |
| int             | $2^{31}$ (-2,147,483,648) to<br>$2^{31}-1$ (2,147,483,647)              | 4                             |
| Bigint          | $-2^{63}$ - $2^{63}-1$<br>(+/- 9 quintillion)                           | 8                             |
| bit             | 1, 0 or NULL  | 1                             |
| decimal/numeric | - $10^{38} + 1$ through $10^{38} - 1$ when<br>maximum precision is used | 5-17                          |
| money           | -922,337,203,685,477.5808 to<br>922,337,203,685,477.5807                | 8                             |
| smallmoney      | - 214,748.3648 to 214,748.3647  | 4                             |

- Dados Caractere Non-Unicode

| Tipo de dado | Intervalo                       | Armazenamento     |
|--------------|---------------------------------|-------------------|
| CHAR(n)      | 1-8000 characters               | n bytes, padded   |
| VARCHAR(n)   | 1-8000 characters               | n+2 bytes         |
| VARCHAR(MAX) | 1-2 <sup>31</sup> -1 characters | Actual length + 2 |

- Data e Hora

| Tipo de dado   | Armazenamento (bytes) | Intervalo de data                    | Precisão           | Formato de entrada recomendado         |
|----------------|-----------------------|--------------------------------------|--------------------|--|
| DATETIME       | 8                     | January 1, 1753 to December 31, 9999 | 3-1/3 milliseconds | 'YYMMDD hh:mm:ss:nnn'                  |
| SMALLDATETIME  | 4                     | January 1, 1900 to June 6, 2079      | 1 minute           | 'YYMMDD hh:mm:ss:nnn'                  |
| DATETIME2      | 6 to 8                | January 1, 0001 to December 31, 9999 | 100 nanoseconds    | 'YYMMDD hh:mm:ss.nnnnnn'               |
| DATE           | 3                     | January 1, 0001 to December 31, 9999 | 1 day              | 'YYYY-MM-DD'                           |
| TIME           | 3 to 5                |                                      | 100 nanoseconds    | 'hh:mm:ss:nnnnnnn'                     |
| DATETIMEOFFSET | 8 to 10               | January 1, 0001 to December          | 100 nanoseconds    | 'YY-MM-DD hh:mm:ss:nnnnnnn [+ -]hh:mm' |

- Data e Hora

| Tipo de dado   | Formatos de linguagem neutros   | Exemplos  |
|----------------|---|---|
| DATETIME       | 'YYYYMMDD hh:mm:ss.nnn'<br>'YYYY-MM-DDThh:mm:ss.nnn'<br>'YYYYMMDD'  | '20120212 12:30:15.123'<br>'2012-02-12T12:30:15.123'<br>'20120212'  |
| SMALLDATETIME  | 'YYYYMMDD hh:mm'<br>'YYYY-MM-DDThh:mm'<br>'YYYYMMDD'  | '20120212 12:30'<br>'2012-02-12T12:30'<br>'20120212'  |
| DATETIME2      | 'YYYY-MM-DD'<br>'YYYYMMDD hh:mm:ss.nnnnnnn'<br>'YYYY-MM-DD hh:mm:ss.nnnnnnn'<br>'YYYY-MM-DDThh:mm:ss.nnnnnnn'<br>'YYYYMMDD'<br>'YYYY-MM-DD' | '20120212 12:30:15.1234567'<br>'2012-02-12 12:30:15.1234567'<br>'2012-02-12T12:30:15.1234567'<br>'20120212'<br>'2012-02-12' |
| DATE           | 'YYYYMMDD'<br>'YYYY-MM-DD'  | '20120212'<br>'2012-02-12'  |
| TIME           | 'hh:mm:ss.nnnnnnn'  | '12:30:15.1234567'  |
| DATETIMEOFFSET | 'YYYYMMDD hh:mm:ss.nnnnnnn [+ -]hh:mm'<br>'YYYY-MM-DD hh:mm:ss.nnnnnnn [+ -]hh:mm'<br>'YYYYMMDD'<br>'YYYY-MM-DD'                            | '20120212 12:30:15.1234567 +02:00'<br>'2012-02-12 12:30:15.1234567 +02:00'<br>'20120212'<br>'2012-02-12'                    |

- Numéricos aproximados

| Tipo de dado | Intervalo   | Armazenamento (bytes)               |
|--------------|---|-------------------------------------|
| float(n)     | - 1.79E+308 to -2.23E-308, 0 and 2.23E-308 to 1.79E+308     | Depends on value of n, 4 or 8 bytes |
| real         | - 3.40E + 38 to -1.18E - 38, 0 and 1.18E - 38 to 3.40E + 38 | 4                                   |



- Binários

| Tipo de dado   | Intervalo                       | Armazenamento<br>(bytes) |
|----------------|---------------------------------|--------------------------|
| binary(n)      | 1-8000 bytes                    | n bytes                  |
| varbinary(n)   | 1-8000 bytes                    | n bytes + 2              |
| varbinary(MAX) | 1-2.1 billion (approx)<br>bytes | actual length + 2        |

- Outros tipos

| Tipo de dado     | Intervalo      | Armazenamento<br>(bytes) | Observações   |
|------------------|----------------|--------------------------|---|
| rowversion       | Auto-generated | 8                        | Successor type to timestamp   |
| uniqueidentifier | Auto-generated | 16                       | Globally unique identifier (GUID)                                   |
| xml              | 0-2 GB         | 0-2 GB                   | Stores XML in native hierarchical structure                         |
| cursor           | N/A            | N/A                      | Not a storage data type   |
| hierarchyid      | N/A            | Depends on content       | Represents position in a hierarchy                                  |
| sql_variant      | 0-8000 bytes   | Depends on content       | Can store data of various data types                                |
| table            | N/A            | N/A                      | Not a storage data type, used for query and programmatic operations |

- Leitura do arquivo PDF disponibilizado na plataforma