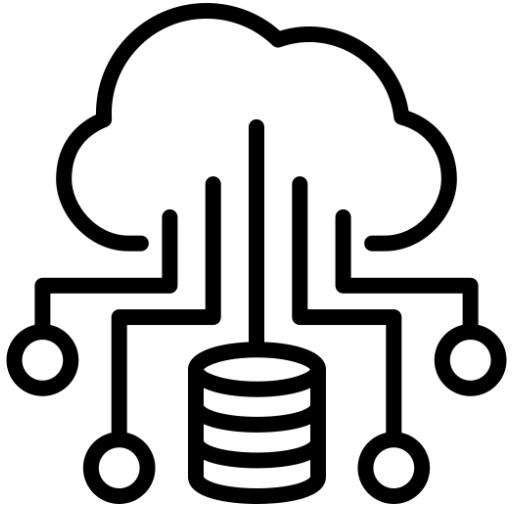
Bases de Dados

Aula Prática 7 – Povoamento e Atualização de Bases de Dados



Docente: Regina Sousa

Horário de Atendimento:
Quarta Feira 10h às 11h
Sala: 1.15

Versão 1

Aula Prática 7

- Rever a implementação física da base de dados
- Rever a aplicação das instruções CREATE TABLE, ALTER TABLE, DROP TABLE
- Povoamento de bases de dados
- Demonstração da aplicação das instruções INSERT, UPDATE, DELETE
- Realização de algumas consultas simples de dados: Instrução SELECT
- Devolver alguns exemplos de aplicação que envolvam projeção, filtragem, ordenação e cálculo de dados

Bibliografia

- Connolly, T., Begg, C., Database Systems, A Practical Approach to Design, Implementation, and Management , Addison-Wesley, 4a Edição, 2004.

Capítulo 4 (Relational Model), 6(Data Manipulation), 7 (Data Definition) e 18 (Methodology — Physical Database Design for Relational Databases)

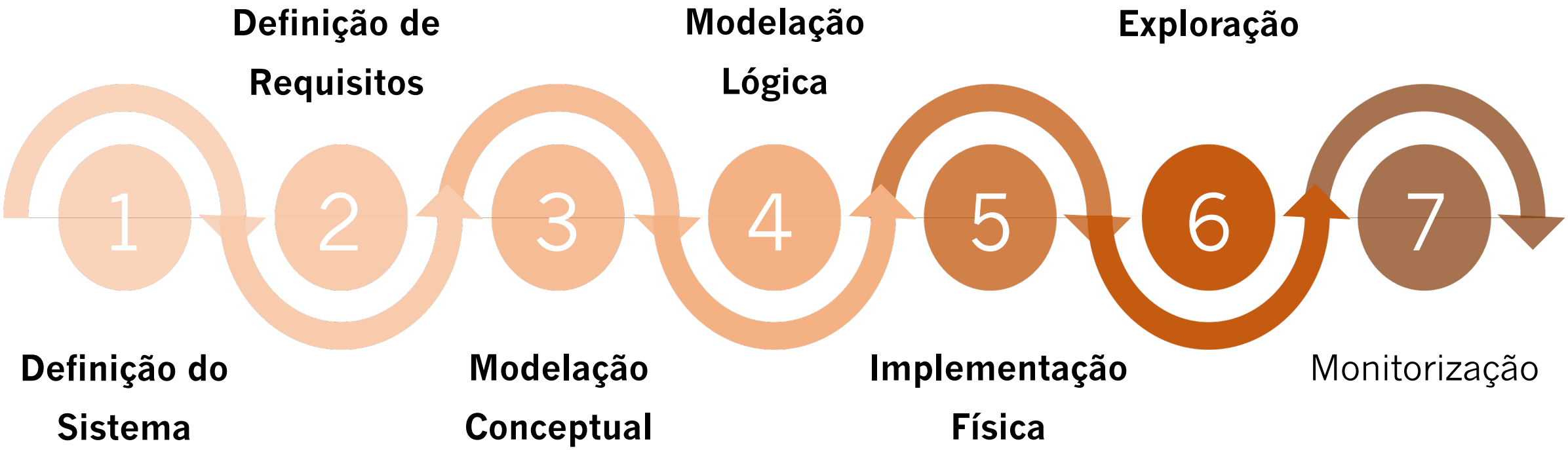
- Belo, O., "Bases de Dados Relacionais: Implementação com MySQL", FCA – Editora de Informática, 376p, Set 2021. ISBN: 978-972-722-921-5.

Capítulos: 1 (Introdução) e 2 (Implementação de Bases de Dados), 3(Povoamento e Atualização de Dados) e 4(Exploração Básica de Dados)

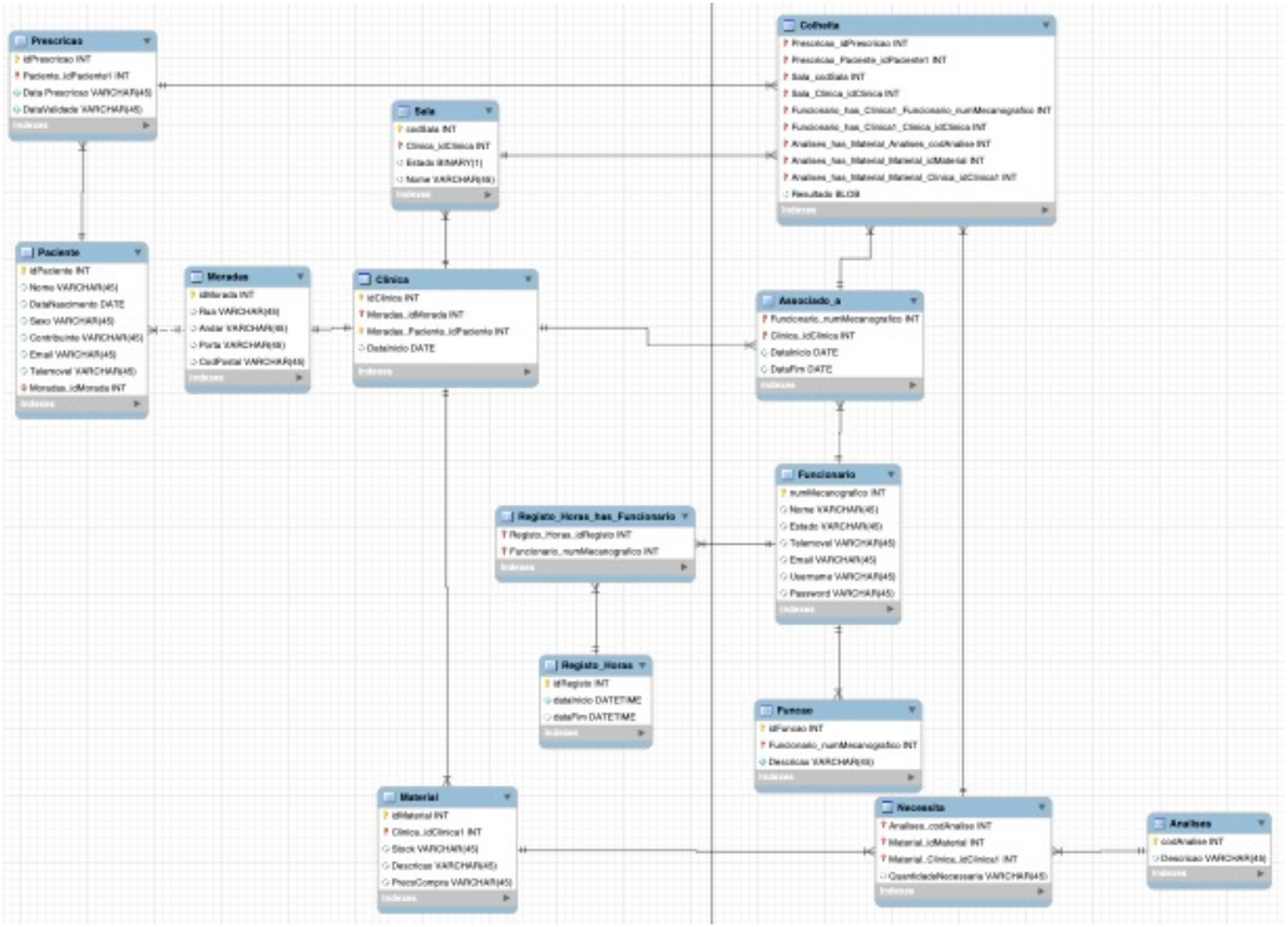
Webgrafia

- https://towardsdatascience.com/coding-and-implementing-a-relational-database-using-mysql-d9bc69be90f5
- https://dev.mysql.com/doc/refman/8.0/en/tutorial.html
- https://www.mysqltutorial.org/
- https://www.youtube.com/watch?v=7S_tz1z_5bA
- https://www.tutorialspoint.com/mysql/index.htm

Ciclo de Vida de Desenvolvimento



Modelo Lógico



Para relembrar ....

SCHEMA

CREATE SCHEMA [Name | **AUTHORIZATION** CreatorIdentifier]
DROP SCHEMA Name [**RESTRICT** | **CASCADE**]

TABLE

CREATE TABLE TableName
{(columnName dataType [**NOT NULL**] [**UNIQUE**]
[**DEFAULT** defaultOption] [**CHECK** (searchCondition)] [, . . .]} [**PRIMARY KEY** (listOfColumns),]
{[**UNIQUE** (listOfColumns)] [, . . .]}
{[**FOREIGN KEY** (listOfForeignKeyColumns)
REFERENCES ParentTableName [(listOfCandidateKeyColumns)]
[**MATCH** {**PARTIAL** | **FULL**}
[**ON UPDATE** referentialAction]
[**ON DELETE** referentialAction]] [, . . .]}
{[**CHECK** (searchCondition)] [, . . .]})

ALTER TABLE TableName
[**ADD** [**COLUMN**] columnName dataType [**NOT NULL**] [**UNIQUE**] [**DEFAULT** defaultOption] [**CHECK** (searchCondition)]]
[**DROP** [**COLUMN**] columnName [**RESTRICT** | **CASCADE**]] [**ADD** [**CONSTRAINT** [ConstraintName]]
tableConstraintDefinition] [**DROP CONSTRAINT** ConstraintName [**RESTRICT** | **CASCADE**]] [**ALTER** [**COLUMN**] **SET DEFAULT** defaultOption]
[**ALTER** [**COLUMN**] **DROP DEFAULT**]

DROP TABLE TableName [**RESTRICT** | **CASCADE**]

INDEX

CREATE [**UNIQUE**] **INDEX** IndexName
ON TableName (columnName [**ASC** | **DESC**] [, . . .

DROP INDEX IndexName

Data Manipulation

Existem 4 instruções básicas para a manipulação dos dados_

SELECT - para consultar dados na base de dados;
INSERT - para inserir dados numa tabela;
UPDATE - para atualizar dados numa tabela;
DELETE - para apagar dados de uma tabela.

INSERT

INSERT INTO TableName [(columnList)] **VALUES**
(dataValueList)

REGRAS PARA A INSTRUÇÃO FUNCIONAR:

- O número de parâmetros em cada lista deve ser o mesmo;
- Deve haver uma correspondência direta na posição dos itens das duas listas, de modo a que o primeiro item na dataValueList se aplique ao primeiro item na columnList, o segundo item na dataValueList se aplique ao segundo item na columnList, e assim por diante;
- O tipo de dados de cada item na dataValueList deve ser compatível com o tipo de dados da coluna correspondente.

SELECT

Questão 1: Produza uma lista de todos os detalhes dos pacientes de todas as clinicas;
Questão 2: Produza uma lista de todos os pacientes listando apenas o nome, data de nascimento e telemóvel;
Questão 3: Liste o código de análise de todas as análises que já foram pedidas;
Questão 4: Verifique quais os Materiais que tem preço de custo superior a 5;
Questão 5: Verificar a descrição e código dos Materiais que custam mais de 5 ou tem stock inferior a 10;
Questão 6: Liste todos os materiais que tem stock entre 0 e 5;
Questão 7: Liste todos os funcionários que se chamam João ou Maria;
Questão 8: Liste o nome e data de nascimento dos pacientes cujo contribuinte tenha 245 na sua constituição;
Questão 9: Liste de forma detalhada todas as colheitas que não tem resultado associado;
Questão 10: Liste as requisições ordenando por data de validade crescente;
Questão 11: Liste a contagem de materiais que tem um custo superior a 20, ordenando de forma decrescente;

SELECT

1. SELECT * FROM pacientes;
2. SELECT nome, dataNascimento, Telemovel FROM pacientes
3. SELECT DISTINCT codAnalise FROM Colheita;
   SELECT codAnalise FROM Colheita GROUP BY codAnalise;
4. SELECT * FROM Material WHERE precoCusto > 5;
5. SELECT idMaterial, Descricao FROM Material where precoCusto>5 or stock<10;
6. SELECT * FROM Material where stock BETWEEN 0 AND 5;
7. SELECT nome FROM funcionário WHERE nome IN (‘Joao’, ‘Maria’);
8. SELECT nome, dataNascimento FROM Pacientes WHERE Contribuinte LIKE(‘%245%’)
9. SELECT * FROM colheitas WHERE resultado IS NOT NULL;
10. SELECT * FROM prescrições ORDER BY dataValidade ASC;
11. SELECT count(*) from Material WHERE precoCusto> 20 ORDER BY precoCusto DESC;

SELECT

SELECT FROM [**WHERE** [**GROUP BY** [**ORDER BY**
[**DISTINCT** | **ALL**] {* | [columnExpression [**AS** newName]] [, . . .]} TableName [alias] [, . . .]
condition]
columnList] [**HAVING** condition]
columnList]