* DDL - Definição: Criar ou alterar um banco de dados ou uma tabela
  + CREATE DATABASE
  + CREATE TABLE
  + ALTER TABLE
  + DROP TABLE
* DML - Manipulação: Incluir e excluir dados, alterar a composição
  + INSERT INTO
  + UPDATE
  + DELETE
  + TRUNCATE
* DQL - Solicitações:
  + SELECT
* DCL - Controle: Definir que usuários podem acessar o banco e que tipo de acesso ele vai ter, que tipo de comandos ele vai poder executar
* DTL - Transações: Qualquer solicitação que pode ser feita por um banco de dados e ela será atendida da melhor maneira possível, seguindo a ***DICA (Durabilidade, Isolamento, Consistência e Atomicidade)***

1. Durabilidade: Todo dado colocado ou alterado tem que permanecer durável enquanto eu quiser que ele esteja lá
2. Isolamento: Se há duas transações sendo feitas ao mesmo tempo, uma **não** pode interferir a outra
3. Consistência: Toda transação tem que levar o banco de dados de um estado consistente a outro consistente
4. Atomicidade: Ou toda a transação acontece ou nada acontece, ou ela dá certo ou ele da um “Ctrl Z” e volta a um estado onde tudo era consistente

* Banco de dados → tabelas → registros → campos
* Parâmetros em SQL se chamam ***constraints***
* ***date →*** *para datas de nascimento*
* ***enum(‘M’, ‘F’) →*** *para casos de registro de sexo*
* ***decimal(x,y) →*** *para casos de peso, altura, etc*
* *Para casos de nacionalidade, usar um* ***default*** *‘Brasil’ ou outra nacionalidade entre ‘’*
* ***auto\_increment →*** *para fazer com que cada pessoa cadastrada tenha um ‘identificador numérico’ definido automaticamente*
* ***If not exists*** *ou* ***if exists →*** *a tabela só vai ser criada se uma tabela não existir ou já existir*
* ***unique →*** *para não possibilitar mais de um registro com o mesmo nome*
* ***unsigned →*** *significa sem sinal, economiza 1 byte a cada registro*
* ***limit ‘x’*** *→ para indicar quantas linhas podem ser afetadas*

1. Criar o banco de dados: ***create database xxx default character set utf8mb4 default collate utf8mb4\_general\_ci;***
2. Criar tabelas: ***create table xxx(***
3. Criar campos:

xxx tipo primitivo (exemplo: varchar(30)) ***not null***,

yyy tipo primitivo,

zzz tipo primitivo,

www tipo primitivo,

***primary key (xxx)***

***) default charset = utf8mb4;***

* Tipos primitivos:

1. Numérico:

* Inteiros → TinyInt, SmallInt, Int, MediumInt, BigInt
* Reais → Decimal, Float, Double, Real
* Lógicos → Bit, Boolean *(sim e não; true e false; 0 e 1)*

1. Data/Tempo → Date, DateTime, TimeStamp, Time, Year
2. Literal:

* Caractere → Char, VarChar
* Texto → TinyText, Text, MediumText, LongText
* Binário → TinyBlob, Blob, MediumBlob, LongBlob
* Coleção → Enum, Set *(permite configurar quais são os valores permitidos)*

1. Espacial → Geometry, Point, Polygon, MultiPolygon

* Usar um banco de dados: USE xxx; **(ou apenas clicar duas vezes no banco de dados desejado)**
* Apagar um banco de dados: DROP DATABASE xxx;
* Inserir dados em uma tabela:

INSERT INTO xxx

(xxx, yyy, zzz, www)

VALUES

(‘dados’, ‘dados’, ‘dados’, ‘dados’)

* Para ver informações da tabela: DESCRIBE xxx;
* Para alterar a tabela: ALTER TABLE xxx  
  ADD COLUMN yyy tipo primitivo;
* Para eliminar uma coluna: ALTER TABLE xxx

DROP COLUMN yyy;

* Para mudar a posição de uma coluna: ALTER TABLE xxx

DROP COLUMN yyy tipo primitivo AFTER zzz;

* Para modificar definições da coluna: ALTER TABLE xxx

MODIFY COLUMN yyy tipo primitivo(com outra especificação)

* Para renomear uma coluna: ALTER TABLE xxx

CHANGE COLUMN yyy zzz tipo primitivo;

* Para renomear a table: ALTER TABLE xxx

RENAME TO yyy;

* Adicionar chave primária: ALTER TABLE xxx

ADD PRIMARY KEY(registro);

Para modificar linhas: UPDATE xxx

SET ‘registro’ = “novo nome do registro”

WHERE ‘chave primária’ = ‘número da chave’;

* Remover uma linha: DELETE FROM xxx

WHERE ‘chave primária’ = ‘número da chave’;

* Remover TODAS as linhas: TRUNCATE TABLE xxx;
* Para fazer uma cópia de segurança (exportar o banco): Server → Data Export → Selecionar o banco de dados → Gerar o ‘dump’ → Escolher qual tipo de exportação (do folder inteiro ou em um arquivo único) → Marcar include create schema → start export
* Para ver os passos de criação de uma tabela ou database:
  + SHOW CREATE TABLE xxx;
  + SHOW CREATE DATABASE xxx;
* SELECT:
  + Selecionando por letras no começo: SELECT \* FROM xxx WHERE yyy LIKE ‘ %’
  + Selecionando por letras no começo: SELECT \* FROM xxx WHERE yyy LIKE ‘% ’
  + Selecionando por letras em qualquer lugar: SELECT \* FROM xxx WHERE yyy LIKE ‘% %’
  + Selecionando pelos que não tem tal letra: SELECT \* FROM xxx WHERE yyy NOT LIKE ‘ %’
  + Distinção: SELECT DISTINCT xxx FROM yyy;
  + Agregação:
    - SELECT COUNT(xxx) FROM yyy;
    - SELECT MAX(xxx) FROM yyy;
    - SELECT MIN(xxx) FROM yyy;
    - SELECT SUM(xxx) FROM yyy;
    - SELECT AVG(xxx0) FROM yyy;
  + Agrupamento: SELECT xxx FROM yyy GROUP BY zzz;
    - SELECT xxx FROM yyy GROUP BY zzz HAVING COUNT(xxx) ><<=>= x;
  + Selecionando colunas: SELECT xxx, yyy, zzz FROM xxx;
  + Selecionando linhas: SELECT \* FROM WHERE xxx = ‘ ’ ORDER BY yyy;
  + Para ordenar conforme uma coluna de forma crescente: SELECT \* FROM xxx ORDER BY yyy;
  + Para ordenar conforme uma coluna de forma decrescente: SELECT \* FROM xxx ORDER BY yyy DESC;

OBSERVAÇÕES

TABELAS RELACIONAIS:

* Relação 1 - 1 (um pra um): escolha a chave primária de um lado e jogue como chave estrangeira no lado dominante (com os mesmo tipos primitivos e constraints)
* Relação 1 - n (um pra muitos): pegue a chave primária do lado 1 e jogue no lado muitos como chave estrangeira (com os mesmo tipos primitivos e constraints)
* Relação n - n (muitos pra muitos): o relacionamento vira uma entidade (tabela), criar dois relacionamentos entre a nova entidade e as outras existentes formando uma relação (1 - n | n - 1) e fazer a regra do relacionamento ‘um pra muitos’ jogando a chave estrangeira para a entidade que era relacionamento
* 4 principais regras para uma boa transação: ACID (Atomicidade, Consistência, Isolamento, Durabilidade)