



BANCO DE DADOS RELACIONAIS

PRO. BIANCA

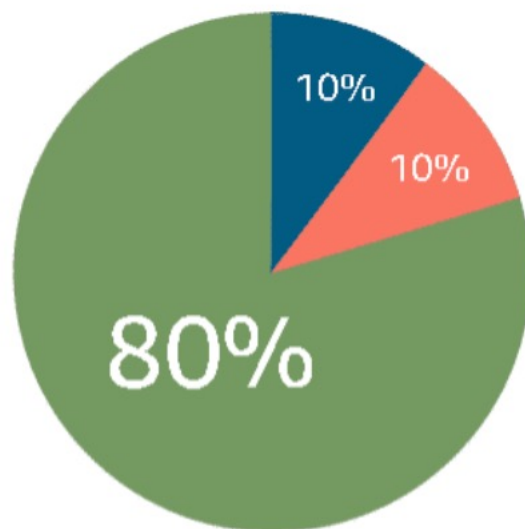
PÓS GRADUAÇÃO EM CIÊNCIA DE DADOS – 2021 @IFSP



CONCEITOS

- NORMALIZAÇÃO
- DIAGRAMA E-R
- ESQUEMAS
- CHAVES PRIMÁRIAS E SECUNDÁRIAS
- ÍNDICES
- SQL

FONTES DE DADOS



Dados estruturados – 10% – Dados tabulares

Dados semiestruturados – 10% – Arquivos CSV, XML, JSON

Dados não estruturados – 80% – Todo o restante

Adaptado do seguinte artigo: <https://www.datamation.com/big-data/structured-vs-unstructured-data.html>

DADOS ESTRUTURADOS



Dados estruturados



Amazon RDS



Amazon Aurora



SQL Server

ORACLE

PRÓS E CONTRAS DOS DADOS ESTRUTURADOS

- Prontos para serem analisados
- Estrutura bem definida (tabela)
- Os dados são facilmente armazenados, editados e recuperados usando uma linguagem SQL comum
- Podem ser volumosos
- Fornece conformidade com ACID/sistema transacional
- Falta de flexibilidade
- Dificuldade no armazenamento de dados não estruturados
- As consultas podem ficar lentas devido às complexas exigências de união
- O esquema pode dificultar a escalabilidade

NORMALIZAÇÃO

- Normalização é um conjunto de regras que funcionam juntas para reduzir a redundância, aumentar a confiabilidade e melhorar a consistência do armazenamento de dados.

DADOS NÃO NORMALIZADOS

idCompra	DataCompra	IdCliente	Nome	Endereco	IdProduto	Descricao	Preco	Qtde	TotalProduto	TotalNF
967	01/08/1998	9670000	Gisele	SP	1	Boné	10	10	100	285
967	02/08/1998	9670000	Gisele	SP	2	Camiseta	25	5	125	285
967	03/08/1998	9670000	Gisele	SP	3	Bermuda	20	3	60	285
971	30/08/1998	9898700	Ivan	Pira	2	Camiseta	25	20	500	1000
971	31/08/1998	9898700	Ivan	Pira	4	Calça	50	10	500	1000
959	25/07/1998	9711111	Bruno	Camp	3	Bermuda	20	6	120	240
959	26/07/1998	9711111	Bruno	Camp	4	Calça	50	2	100	240
959	27/07/1998	9711111	Bruno	Camp	1	Boné	10	2	20	240
951	20/06/1998	9670000	Gisele	SP	4	Calça	50	1	50	50

PRIMEIRA FORMA NORMAL



idCompra	DataCompra	IdCliente	Nome	Endereco
967	01/08/1998	9670000	Gisele	SP
971	30/08/1998	9898700	Ivan	Pira
959	25/07/1998	9711111	Bruno	Camp
951	20/06/1998	9670000	Gisele	SP

PK

PK+FK

idCompra	IdProduto	Descricao	Preco	Qtde	TotalProduto	TotalINF
967	1	Boné	10	10	100	285
967	2	Camiseta	25	5	125	285
967	3	Bermuda	20	3	60	285
971	2	Camiseta	25	20	500	1000
971	4	Calça	50	10	500	1000
959	3	Bermuda	20	6	120	240
959	4	Calça	50	2	100	240
959	1	Boné	10	2	20	240
951	4	Calça	50	1	50	50

SEGUNDA FORMA NORMAL

idCompra	DataCompra	IdCliente	Nome	Endereco
967	01/08/1998	9670000	Gisele	SP
971	30/08/1998	9898700	Ivan	Pira
959	25/07/1998	9711111	Bruno	Camp
951	20/06/1998	9670000	Gisele	SP

IdProduto	Descricao	Preco
1	Boné	10
2	Camiseta	25
3	Bermuda	20
4	Calça	50

PK

PK+FK

idCompra	IdProduto	Qtde	TotalProduto	TotalNF
967	1	10	100	285
967	2	5	125	285
967	3	3	60	285
971	2	20	500	1000
971	4	10	500	1000
959	3	6	120	240
959	4	2	100	240
959	1	2	20	240
951	4	1	50	50

DADOS NORMALIZADOS

idCompra	DataCompra	IdCliente	TotalCompra
967	01/08/1998	9670000	285
971	30/08/1998	9898700	1000
959	25/07/1998	9711111	240
951	20/06/1998	9670000	50

IdCliente	Nome	Endereco
9670000	Gisele	SP
9898700	Ivan	Pira
9711111	Bruno	Camp

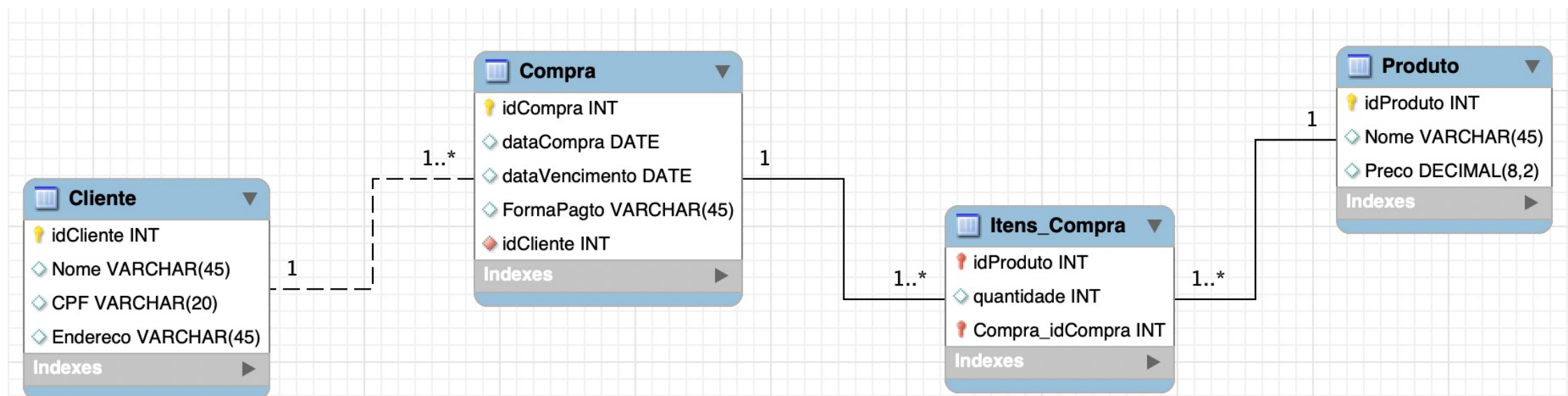
idCompra	IdProduto	Qtde	TotalProduto
967	1	10	100
967	2	5	125
967	3	3	60
971	2	20	500
971	4	10	500
959	3	6	120
959	4	2	100
959	1	2	20
951	4	1	50

IdProduto	Descricao	Preco
1	Boné	10
2	Camiseta	25
3	Bermuda	20
4	Calça	50

PK

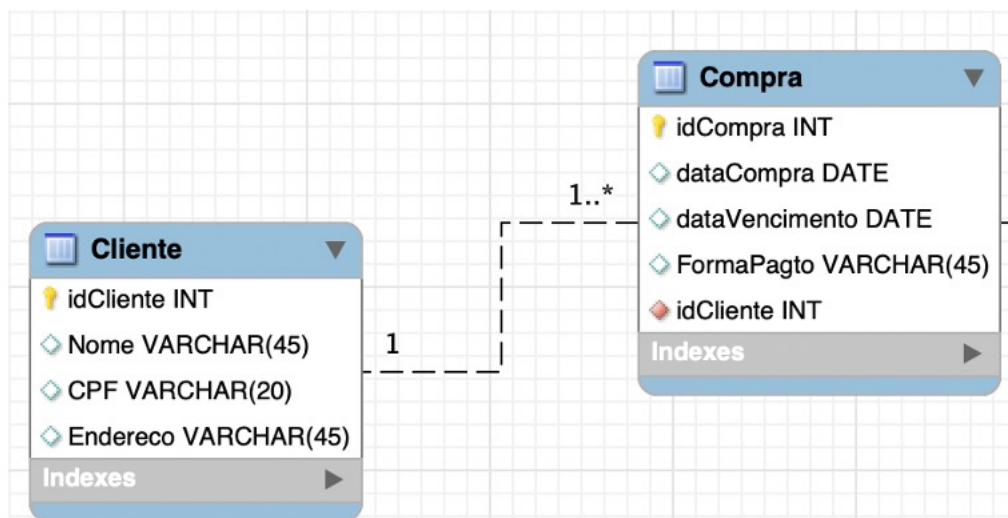
PK+FK

MODELO RELACIONAL



CHAVES

- Chave primária
 - Coluna que contém valores únicos para as linhas de uma tabela
- Chave estrangeira
 - Faz referência a chave primária de uma tabela, garantindo a consistência dos dados

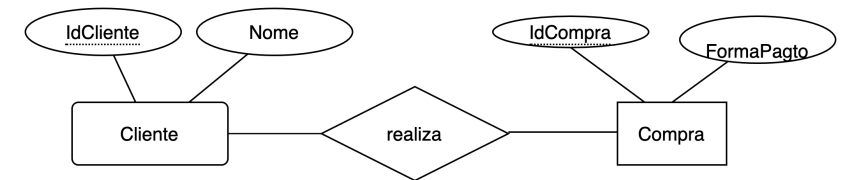


A integridade referencial é o processo que garante que as restrições dos relacionamentos da tabela sejam aplicadas, impedindo registros órfãos, que façam referência a outros registros que não existem mais.

NÍVEIS DE ABSTRAÇÃO

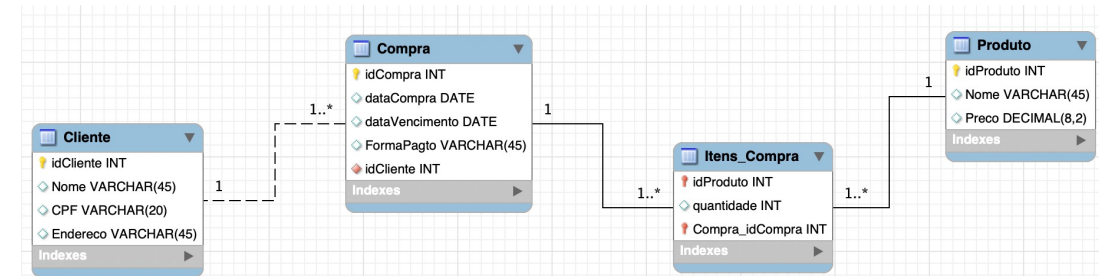
Nível Conceitual

DER



Nível Lógico

Relacional



Nível Físico

SQL

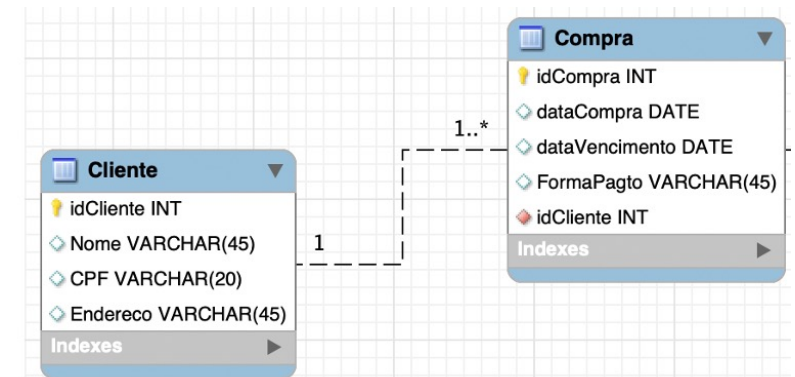
```
CREATE TABLE `Cliente` (  
  idCliente int NOT NULL,  
  Nome varchar(45),  
  CPF varchar(20),  
  Endereco varchar(45),  
  PRIMARY KEY (`idCliente`)  
);
```

SQL

- Linguagem de consulta estruturada
 - Definição de dados (create, alter, drop)
 - Manipulação de dados (insert, update, delete)
 - Linguagem de consulta
 - Junção
 - Visão
 - Agregação

JUNÇÃO

- Recompõem as tabelas quebradas na normalização
- Requer a existência de relacionamentos



```
1 SELECT * FROM
2 cliente inner join compra ON cliente.idcliente=compra.idcliente
```

idCliente	Nome	CPF	Endereco	idCompra	dataCompra	dataVencimento	FormaPagto	idCliente
201	Pedro Oliveira	231213	Rua das Hortências, 100 - Bragança Pta	10	2020-04-01	2020-05-02	CC	201
321	Maria Souza	121234	Rua das Pedras ,20 - Bragança Pta	20	2020-04-01	2020-05-02	CC	321
421	Ana Freire	198716	Rua das Hortências,200 - Campinas	30	2020-04-05	2020-05-02	CC	421
421	Ana Freire	198716	Rua das Hortências,200 - Campinas	100	2020-04-10	2020-05-02	CC	421
201	Pedro Oliveira	231213	Rua das Hortências, 100 - Bragança Pta	112	2020-04-10	2020-05-02	CC	201

AGREGAÇÃO

- Quantas unidades de cada produto foram compradas?

idProduto	Unidades
1	41
2	2
3	1
4	1
5	1

```
1 SELECT idProduto, sum(quantidade) Unidades
2 FROM itens_compra
3 group by idProduto
4 order by UNIDADES DESC;
```


VISÕES

- A visão armazena os dados de uma consulta em uma tabela virtual temporária. Esta tabela virtual com os resultados já compilados permanecem em cache para as consultas sobre aquela visão. Por isto, consultar uma visão torna-se **mais rápido** que executar a consulta em si, realizando todas as suas operações.
- Além disto, você economiza tempo com retrabalho, pois não precisa reescrever aquela instrução enorme toda vez que precisar. Escreva uma vez, armazene e utilize sempre que precisar!
- As views podem mascarar complexidade do banco de dados, pois detalhes da consulta como junções e funções de agregação tornam-se invisíveis.

EXEMPLO DE VISÃO

```
CREATE VIEW TOTAL_COMPRA AS
SELECT ITENS.idCompra, SUM(itens.quantidade * pro.preco) total
FROM itens_compra itens INNER JOIN produto pro ON pro.idProduto=itens.idProduto
GROUP BY ITENS.idcompra;
```

```
SELECT * FROM TOTAL_COMPRA;
```

idCompra	total
112	2.00
10	76.00
100	140.00
30	185.00
20	285.00