Objetivos:

- Apresentar o conceito de normalização
- Apresentar e exemplificar a 1ª forma normal
- Apresentar e exemplificar a 2ª forma normal
- Apresentar e exemplificar a 3ª forma normal

NORMALIZAÇÃO

- O processo de **normalização de dados** representa uma série de passos que se seguem no projeto de um banco de dados que permitem um armazenamento consistente e um eficiente acesso aos dados de um banco de dados relacional.

- Esses passos reduzem a redundância de dados e, consequentemente, as chances de ocorrerem inconsistências.

- A normalização serve para analisar tabelas e organizá-las de forma que sua estrutura seja simples, relacional e estável, para que o gerenciamento possa ser também simples.
- Os objetivos são evitar perda e repetição da informação e atingir uma forma de representação adequada para o que se deseja armazenar.
- Oferecer mecanismos para analisar o projeto de Banco de Dados e a identificação de erros.
- Oferecer métodos para corrigir o problema

Perigos Potenciais nos Projetos de Bancos de Dados Relacionais

- **Repetição da Informação**
  - Informações repetidas consomem espaço de armazenamento
  - Dificuldade de atualização
- **Incapacidade de representar a informação**
  - Incidência de valores nulos
- **Perda da Informação**
  - Projetos mal elaborados sugerem a decomposição de esquemas relacionais com muitos atributos

Através do processo de normalização pode-se, gradativamente, substituir um conjunto de entidades e relacionamentos por um outro, o qual se apresenta "purificado" em relação às anomalias de atualização (inclusão, alteração e exclusão) as quais podem causar certos problemas, tais como:

- grupos repetitivos (atributos multivalorados) de dados;
- variação temporal de certos atributos, dependências funcionais totais ou parciais em relação a uma chave concatenada;
- redundâncias de dados desnecessárias;
- perdas acidentais de informação;
- dificuldade na representação de fatos da realidade observada;
- dependências transitivas entre atributos.

Primeira forma normal (1FN)

Uma tabela está na primeira forma normal quando se todos os seus atributo são monovalorados e atômicos, isto é, não contém tabelas aninhadas

CodProj	Tipo	Descr	Emp					
			CodEmp	Nome	Cat	Sal	DataIni	TempAl
LSC001	Novo Desenv.	Sistema de Estoque	2146	João	A1	4	1/11/91	24
			3145	Silvio	A2	4	2/10/91	24
			6126	José	B1	9	3/10/92	18
			1214	Carlos	A2	4	4/10/92	18
			8191	Mário	A1	4	1/11/92	12
PAG02	Manutenção	Sistema de RH	8191	Mário	A1	4	1/05/93	12
			4112	João	A2	4	4/01/91	24
			6126	José	B1	9	1/11/92	12

Primeira forma normal (1FN)

CodProj	Tipo	Descr	CodEmp	Nome	Cat	Sal	DatIni	TempAl
LSC001	Novo Deseny.	Sistema de Estoque	2146	João	A1	4	1/11/91	24
LSC001	Novo Deseny.	Sistema de Estoque	3145	Silvio	A2	4	2/10/91	24
LSC001	Novo Deseny.	Sistema de Estoque	6126	José	B1	9	3/10/92	18
LSC001	Novo Deseny.	Sistema de Estoque	1214	Carlos	A2	4	4/10/92	18
LSC001	Novo Deseny.	Sistema de Estoque	8191	Mário	A1	4	1/11/92	12
PAG02	Manutenção	Sistema de RH	8191	Mário	A1	4	1/05/93	12
PAG02	Manutenção	Sistema de RH	4112	João	A2	4	4/01/91	24
PAG02	Manutenção	Sistema de RH	6126	José	B1	9	1/11/92	12

Primeira forma normal (1FN)

- Cria-se uma tabela na 1FN referente à tabela Não Normalizada e que contém apenas colunas com valores atômicos, isto é, sem as tabelas aninhadas;
- Para cada tabela aninhada, cria-se uma tabela na 1FN compostas pelas seguintes colunas:
  - A chave primária de uma das tabelas na qual a tabela em questão está aninhada
  - As colunas da própria tabela
- São definidas as chaves primárias das tabelas na 1FN que correspondem a tabelas aninhadas

Primeira forma normal (1FN)

CodProj	Tipo	Descr	CodEmp	Nome	Cat	Sal	DatIni	TempAl
LSC001	Novo Deseny.	Sistema de Estoque	2146	João	A1	4	1/11/91	24
LSC001	Novo Deseny.	Sistema de Estoque	3145	Silvio	A2	4	2/10/91	24
LSC001	Novo Deseny.	Sistema de Estoque	6126	José	B1	9	3/10/92	18
LSC001	Novo Deseny.	Sistema de Estoque	1214	Carlos	A2	4	4/10/92	18
LSC001	Novo Deseny.	Sistema de Estoque	8191	Mário	A1	4	1/11/92	12
PAG02	Manutenção	Sistema de RH	8191	Mário	A1	4	1/05/93	12
PAG02	Manutenção	Sistema de RH	4112	João	A2	4	4/01/91	24
PAG02	Manutenção	Sistema de RH	6126	José	B1	9	1/11/92	12

Segundo a definição da 1FN, para normalizar a tabela acima, é necessário decompô-la em duas:

Proj(CodProj, Tipo, Descr);

Emp(CodProj, CodEmp, Nome, Cat, Sal, DataIni, TempAl);

Primeira forma normal (1FN)

Proj(CodProj, Tipo, Descr);

Emp(CodProj, CodEmp, Nome, Cat, Sal, DataIni, TempAl);

Proj

CodProj	Tipo	Descr
LSC001	Novo Deseny.	Sistema de Estoque
PAG02	Manutenção	Sistema RH

ProjEmp

CodProj	CodEmp	Nome	Cat	Sal	DataIni	TempAl
LSC001	2146	João	A1	4	1/11/91	24
LSC001	3145	Silvio	A2	4	2/10/91	24
LSC001	6126	José	B1	9	3/10/92	18
LSC001	1214	Carlos	A2	4	4/10/92	18
LSC001	8191	Mário	A1	4	1/11/92	12
LSC001	8191	Mário	A1	4	1/05/93	12
PAG02	4112	João	A2	4	4/01/91	24
PAG02	6126	José	B1	9	1/11/92	12

Quando encontrarmos um atributo multivalorado, deve-se criar um novo atributo que individualize a informação que esta multivalorada:

BOLETIM (matricula-aluno, materia, notas)

No caso acima, cada nota seria individualizada identificando a prova a qual aquela nota se refere:

BOLETIM (matricula-aluno, materia, numero-prova, nota)

Quando encontrarmos um atributo não atômico, deve-se dividi-lo em outros atributos que sejam atômicos:

PESSOA (CPF, nome-completo)

Vamos supor que, para a aplicação que utilizará esta relação, o atributo nome-completo não é atômico, a solução então será:

PESSOA (CPF, nome, sobrenome)

Segunda Forma Normal (2FN)

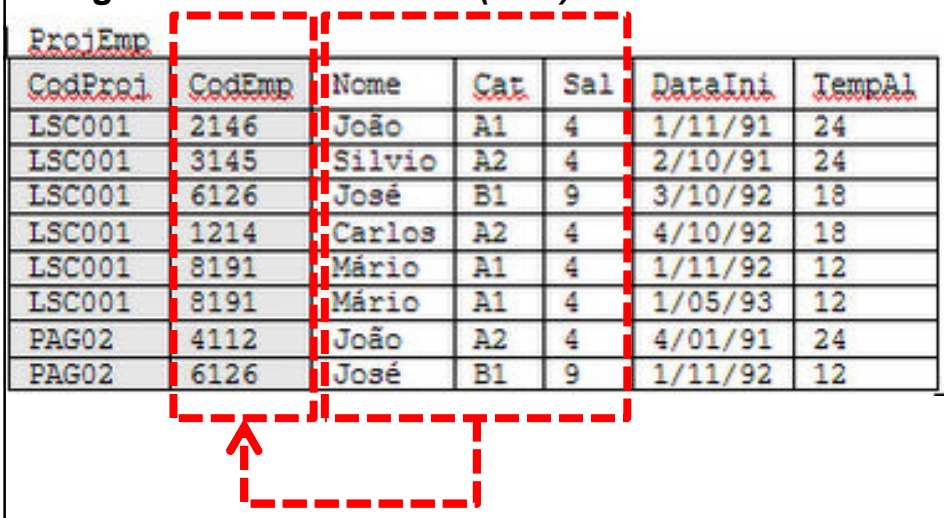
Uma tabela encontra-se na segunda forma normal, quando, além de estar na 1FN, não contem dependências parciais.

Dependência parcial **-> Uma dependência parcial ocorre quando uma coluna depende apenas de parte de uma chave primária composta.**

Segunda Forma Normal (2FN)

ProjEmp

CodProj	CodEmp	Nome	Cat	Sal	DataIni	TempAl
LSC001	2146	João	A1	4	1/11/91	24
LSC001	3145	Silvio	A2	4	2/10/91	24
LSC001	6126	José	B1	9	3/10/92	18
LSC001	1214	Carlos	A2	4	4/10/92	18
LSC001	8191	Mário	A1	4	1/11/92	12
LSC001	8191	Mário	A1	4	1/05/93	12
PAG02	4112	João	A2	4	4/01/91	24
PAG02	6126	José	B1	9	1/11/92	12



Depende apenas de CodEmp e não da Chave Primária (CodProj, CodEmp)

Segunda Forma Normal (2FN)

- Copiar para a 2FN cada tabela que tenha chave primária simples ou que não tenha colunas além da chave.
- Para cada tabela com chave primária composta e com pelo menos uma coluna não chave:
  - Criar na 2FN uma tabela com as chaves primárias da tabela na 1FN
    - Para cada coluna não chave fazer a seguinte pergunta: *"a coluna depende de toda a chave ou de apenas parte dela"*
    - Caso a coluna dependa de toda a chave
      - Criar a coluna correspondente na tabela com a chave completa na 2FN
    - Caso a coluna não dependa apenas de parte da chave
      - Criar, caso ainda não existir, uma tabela na 2FN que tenha como chave primária a parte da chave que é determinante da coluna em questão
      - Criar a coluna dependente dentro da tabela na 2FN

Segunda Forma Normal (2FN)

Proj

CodProj	Tipo	Descr
LSC001	Novo Desenv.	Sistema de Estoque
PAG02	Manutenção	Sistema RH

ProjEmp

CodProj	CodEmp	DataIni	TempAl
LSC001	2146	1/11/91	24
LSC001	3145	2/10/91	24
LSC001	6126	3/10/92	18
LSC001	1214	4/10/92	18
LSC001	8191	1/11/92	12
LSC001	8191	1/05/93	12
PAG02	4112	4/01/91	24
PAG02	6126	1/11/92	12

Emp

CodEmp	Nome	Cat	Sal
2146	João	A1	4
3145	Silvio	A2	4
6126	José	B1	9
1214	Carlos	A2	4
8191	Mário	A1	4

Observe a relação abaixo:

BOLETIM
(matricula-aluno, codigo-materia, numero-prova, nota, data-da-prova, nome-aluno, endereço-aluno, nome-materia)

Fazendo a análise da dependência funcional de cada atributo primo, chegamos às seguintes dependências funcionais:

matricula-aluno, codigo-materia, numero-prova -> nota
codigo-materia, numero-prova -> data-da-prova
matricula-aluno -> nome-aluno, endereço-aluno
codigo-materia -> nome-materia

Concluímos então que apenas o atributo primo nota depende totalmente de toda chave primária. Para que toda a relação seja passada para a segunda forma normal, deve-se criar novas relações, agrupando os atributos de acordo com suas dependências funcionais:

BOLETIM (matricula-aluno, codigo-materia, numero-prova, nota)

PROVA (codigo-materia, numero-prova, data-da-prova)

ALUNO (matricula-aluno, nome-aluno, endereço-aluno)

MATERIA (codigo-materia, nome-materia)

Terceira Forma Normal (3FN)

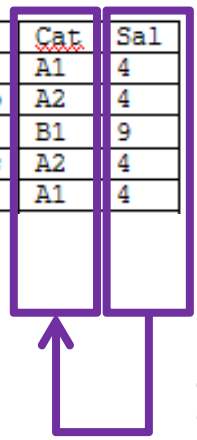
Uma tabela encontra-se na terceira forma normal, quando, além de estar na 2FN, não contém dependências transitivas

Dependência transitiva -> Uma dependência funcional transitiva ocorre quando uma coluna, além de depender da chave primária da tabela, depende de outra coluna ou conjunto de colunas da tabela

Terceira Forma Normal (3FN)

Emp

CodEmp	Nome	Cat	Sal
2146	João	A1	4
3145	Silvio	A2	4
6126	José	B1	9
1214	Carlos	A2	4
8191	Mário	A1	4



Supondo que o salário de um empregado é determinado por sua categoria funcional (Cat)

Terceira Forma Normal (3FN)

- Copiar para o esquema da 3FN cada tabela que tenha menos de duas colunas não chave, pois neste caso não há como haver dependências transitivas
- Para tabelas com duas ou mais colunas não chaves, fazer a seguinte pergunta: ”*a coluna depende de alguma outra coluna não chave?”*
  - Caso dependa apenas da chave
    - Copiar a coluna para a tabela na 3FN
  - Caso a coluna depender de outra coluna
    - Criar, caso ainda não exista, uma tabela no esquema na 3FN que tenha como chave primária a coluna na qual há a dependência indireta
    - Copiar a coluna dependente para a tabela criada
    - A coluna determinante deve permanecer também na tabela original

Terceira Forma Normal (3FN)

A tabela Emp obtida pela aplicação da 2FN, que contém os dados redundantes, pode ser decomposta em outras duas (Emp e Cat)

Emp

CodEmp	Nome	Cat
2146	João	
3145	Silvio	
6126	José	
1214	Carlos	
8191	Mário	

Cat

Cat	Sal
A1	4
A2	4
B1	9

Emp

CodEmp	Nome	Cat	Sal
2146	João	A1	4
3145	Silvio	A2	4
6126	José	B1	9
1214	Carlos	A2	4
8191	Mário	A1	4

CodProj	Tipo	Descr	CodEmp	Nome	Cat	Sal	DatIni	TempAl
LSC001	Novo Deseny.	Sistema de Estoque	2146	João	A1	4	1/11/91	24
LSC001	Novo Deseny.	Sistema de Estoque	3145	Silvio	A2	4	2/10/91	24
LSC001	Novo Deseny.	Sistema de Estoque	6126	José	B1	9	3/10/92	18
LSC001	Novo Deseny.	Sistema de Estoque	1214	Carlos	A2	4	4/10/92	18
LSC001	Novo Deseny.	Sistema de Estoque	8191	Mário	A1	4	1/11/92	12
PAG02	Manutenção	Sistema de RH	8191	Mário	A1	4	1/05/93	12
PAG02	Manutenção	Sistema de RH	4112	João	A2	4	4/01/91	24
PAG02	Manutenção	Sistema de RH	6126	José	B1	9	1/11/92	12

Proj

CodProj	Tipo	Descr
LSC001	Novo Deseny.	Sistema de Estoque
PAG02	Manutenção	Sistema RH

Emp

CodEmp	Nome	Cat
2146	João	
3145	Silvio	
6126	José	
1214	Carlos	
8191	Mário	

ProjEmp

CodProj	CodEmp	DataIni	TempAl
LSC001	2146	1/11/91	24
LSC001	3145	2/10/91	24
LSC001	6126	3/10/92	18
LSC001	1214	4/10/92	18
LSC001	8191	1/11/92	12
LSC001	8191	1/05/93	12
PAG02	4112	4/01/91	24
PAG02	6126	1/11/92	12

Cat

Cat	Sal
A1	4
A2	4
B1	9

Observe a relação abaixo:

PEDIDO(numero-pedido, codigo-cliente, data-pedido, nome-cliente,codigo-cidade-cliente, nome-cidade-cliente)

Fazendo a análise da dependência funcional de cada atributo primo, chegamos às seguintes dependências funcionais:

numero-pedido -> codigo-cliente
numero-pedido -> data-pedido
codigo-cliente -> nome-cliente
codigo-cliente -> codigo-cidade-cliente
codigo-cidade-cliente -> nome-cidade-cliente

Concluímos então que apenas os atributos primos codigo-cliente e data-pedido dependem não transitivamente totalmente de toda chave primária.

Observe que:

numero-pedido -> codigo-cliente -> nome-cliente

numero-pedido -> codigo-cliente -> codigo-cidade-cliente

numero-pedido -> codigo-cliente -> codigo-cidade-cliente -> nome-cidade-cliente

Devemos resolver inicialmente as dependências mais simples, criando uma nova relação onde codigo-cliente é a chave, o codigo-cliente continuará na relação PEDIDO como atributo primo, porém, os atributos que dependem dele devem ser transferidos para a nova relação:

PEDIDO (numero-pedido, codigo-cliente, data-pedido)

CLIENTE (codigo-cliente, nome-cliente, codigo-cidade-cliente, nome-cidade-cliente)

As dependências transitivas da relação PEDIDO foram eliminadas, porém ainda devemos analisar a nova relação CLIENTE:

codigo-cliente -> codigo-cidade-cliente -> nome-cidade-cliente

Observe que o nome-cidade-cliente continua com uma dependência transitiva, vamos resolvê-la da mesma maneira :

PEDIDO (numero-pedido, codigo-cliente, data-pedido)
CLIENTE(codigo-cliente, nome-cliente, codigo-cidade-cliente)
CIDADE (codigo-cidade-cliente, nome-cidade-cliente