Normalização (Técnica para produzir um conjunto de relações com as propriedades desejadas, dados os requisitos de uma empresa)

A normalização é uma técnica de modelação de bases de dados, que começa por examinar os relacionamentos entre atributos (também denominados dependencias funcionais).
Os atributos descrevem propriedades dos dados ou dos relacionamentos entre os dados que são importantes para a empresa. A normalização utiliza uma série de testes (formas normais) para ajudar a identificar o agrupamento ótimo destes atributos de forma a identificar um conjunto de relações que satisfaz os requisitos da empresa.

Objetivos da normalização

- minimizar o número de atributos necessários para suportar os requisitos de dados de uma empresa.

- atributos com um relacionamento lógico próximo (dependencia funcional) são encontrados na mesma relação.

- minimizar a redundância fazendo que cada atributo esta presente apenas uma vez, com excepção de atributos que fazem parte de chaves estrangeiras (as bases de dados relacionais dependem de alguma redundancia de dados, nomeadamente as chaves primárias que fazem de chaves estrangeiras, que estão presentes em várias relaçoes).

Vantagens

- facilitar o uso e manuntenção da base de dados (updates são realizados com um minimo número de operações, reduzindo as oportunidades de inconsistencia de dados).

- ocupar menos espaço de armazenamento de dados (minimizando também os custos).

Como a normalização suporta a modelação de bases de dados

Aproximação 1: Usar normalização como uma técnica "bottom-up" (atributos -> entidades) para criar um conjunto de relações válidas.

Aproximação 2: Usar normalização como uma técnica de validação para confirmar a estrutura dos relacionamentos (usada no trabalho).

Uma apróximação "top-down" é a modelação ER (entidades -> atributos).

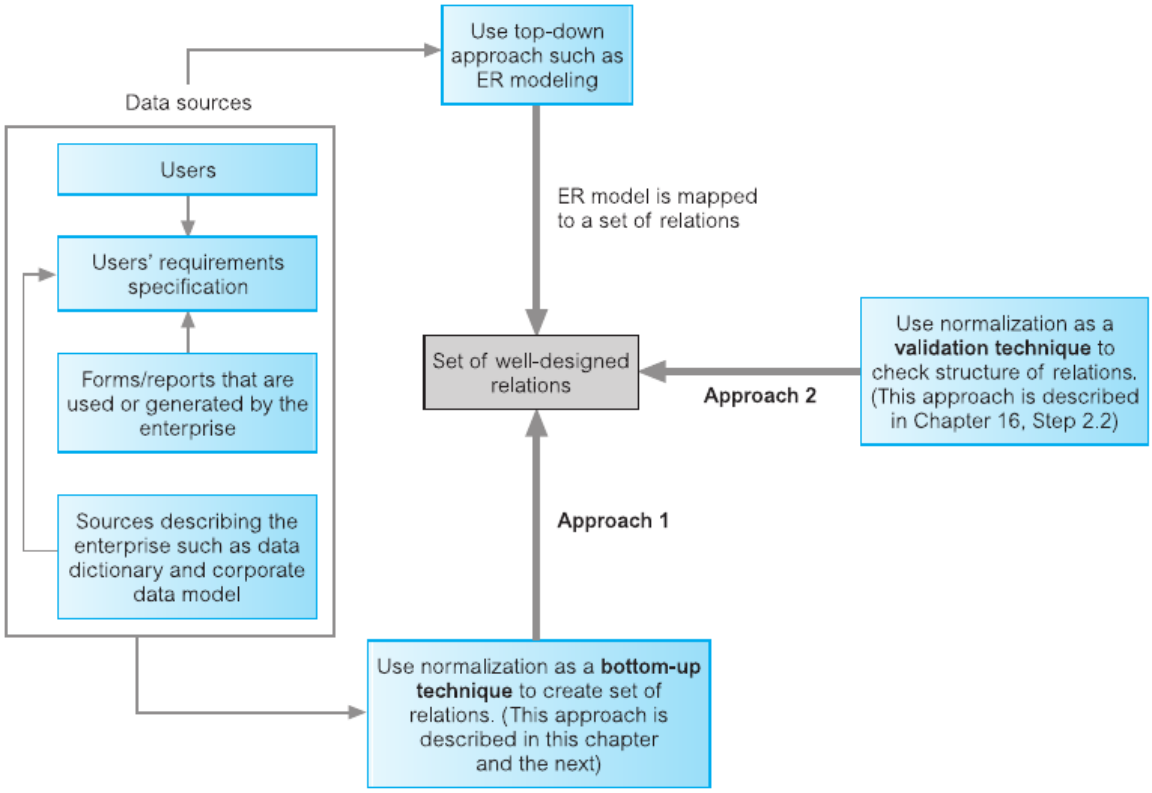


Figure 13.1 How normalization can be used to support database design.

Redundância de dados e anomalias de update

Branch

branchNo	bAddress
B005	22 Deer Rd, London
B007	16 Argyll St, Aberdeen
B003	163 Main St, Glasgow

Staff

staffNo	sName	position	salary	branchNo
SL21	John White	Manager	30000	B005
SG37	Ann Beech	Assistant	12000	B003
SG14	David Ford	Supervisor	18000	B003
SA9	Mary Howe	Assistant	9000	B007
SG5	Susan Brand	Manager	24000	B003
SL41	Julie Lee	Assistant	9000	B005

StaffBranch

staffNo	sName	position	salary	branchNo	bAddress
SL21	John White	Manager	30000	B005	22 Deer Rd, London
SG37	Ann Beech	Assistant	12000	B003	163 Main St, Glasgow
SG14	David Ford	Supervisor	18000	B003	163 Main St, Glasgow
SA9	Mary Howe	Assistant	9000	B007	16 Argyll St, Aberdeen
SG5	Susan Brand	Manager	24000	B003	163 Main St, Glasgow
SL41	Julie Lee	Assistant	9000	B005	22 Deer Rd, London

Anomalias de inserção (solução: dividir em duas tabelas):

- inconsistência de dados: ao adicionar uma nova entrada na tabela, o número do branch deve estar associado a uma descrição. Se houver o mesmo número de branch com diferentes descrições, existe inconsistência.

- não ser possivel inserir apenas um branch: no caso em que apenas se quer inserir um branch, ter-se-ia de meter o resto tudo a nulls. Mas o chave primária da tabela refere-se ao número de staff, pelo que inserir um null na chave primária estaria a violar a integridade da entidade.

Anomalias de remoção:

- Se apagarmos a última entrada de um dado membro que pertença a um dado branch, esse branch deixará de estar presente na base de dados.

Anomalias de modificação:

- Se quisermos mudar o valor de um dos atributos de um dado branch, teremos de atualizar todas entradas que possuam a informação sobre esse branch. Caso isso não seja realizado, haverá inconsistência de dados.

Dependências funcionais

Descrevem o relacionamento entre atributos. Por exemplo, se A e B são atributos de uma relação R, B é funcionalmente dependente de A (A → B), se cada valor de A está associado com exatamente um valor de B (A e B podem consistir cada um de vários atributos). Então, para dois tuplos que possuam o mesmo valor de A, também possuem o mesmo valor de B. No entanto, para cada valor de B podem haver diferentes valores de A (A é chamado o determinante de B).

Dependencia funcional completa:
Indica que se A e B são atributos de uma relação, B é completamente dependente funcionalmente em A se B é funcionalmente dependente em A, mas não em qualquer subconjunto de A.
A dependencia funcional A → B é completa se a remoção de qualquer atributo de A faz com que a dependência deixe de existir.

Dependencia funcional parcial:
A dependencia funcional A → B é parcial se existe um atributo que possa ser removido de A e a dependencia ainda se verificar.

Exemplo:

staffNo, sName → branchNo

É correcto dizer que cada valor de (staffNo, SName) está associado com um único valor de branchNo. No entanto, esta dependência não é completa dado que o branchNo é também funcionalmente dependete de um subconjunto de (staffNo, SName), nomeadamente staffNo. Por outras palavras, a dependencia funcional mostrada acima é um exemplo de dependencia parcial.
Uma dependencia funcional completa seria: staffNo → branchNo

Resumindo, nas dependencias funcionais:

- Existe um relacionamento entre os atributos do lado esquerdo (determinante) para o lado direito de 1-1. (Do lado direito para o lado esquerdo pode ser 1-1 ou 1-*).

- Servem para todo o tempo, e não apenas um dado momento (instancia) da base de dados.

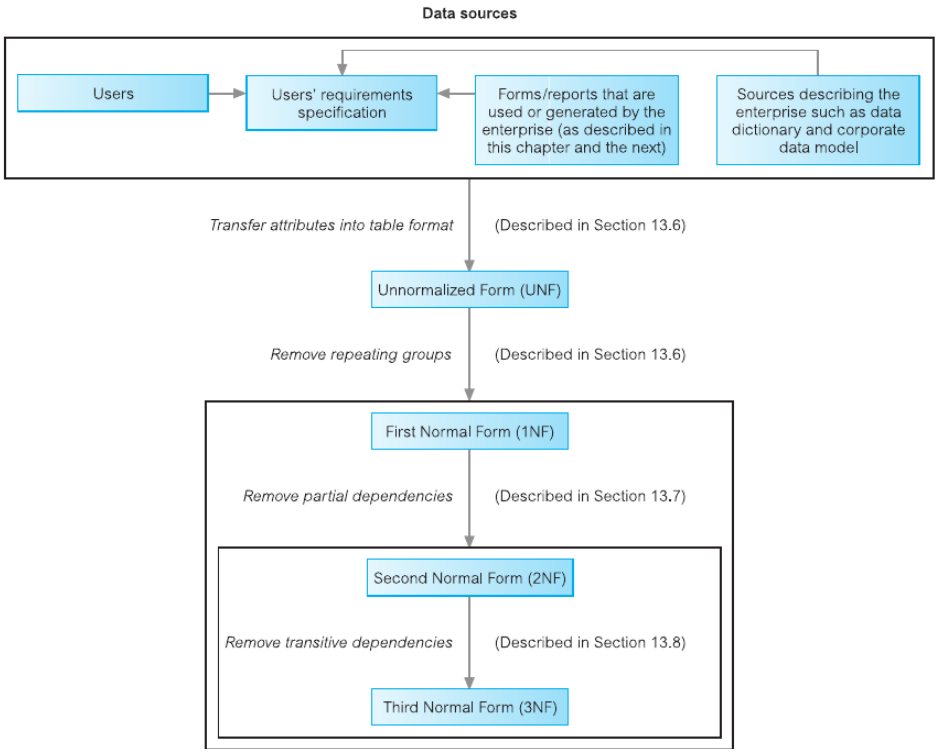
- O determinante tem o número minimo de atributos necessarios para manter a dependencia (só nos interessa dependencia funcional completa).

Regras de dependencias funcionais:

Reflexividade: Se B contido em A, A -> B
Augmentation: Se A -> B entao A, C -> B, C
Transitividade: Se A -> B e B -> C, então A->C (ex: staffNo → sName, position, salary, branchNo, bAddress e branchNo → bAddress)
Auto-determinação: A->A
Decomposição: Se A -> B,C então A->B e A->C
Uniao: Se A->B e A->C, então A -> B,C
Composiçao: Se A->B e C->D, então A,C -> B,D

Normalização

Normalização é uma técnica formal para analizar relações com base na sua chave primária (ou candidata) e dependencias funcionais. A técnica envolve uma série de regras que podem ser usadas para testar relações individuais de forma a que a base de dados possa ser normalizada. Quando um requisito não é cumprido, a relação que viola esse requisito deve ser decomposta em relações que individualmente cumprem os requisitos de normalização.
Para o modelo de dados relacional, é importante dizer que é apenas necessária 1ºFN que é critica ao criar relações. Todas as subsequentes formas normais são opcionais, no entanto, para evitar anomalias de atualização, é geralmente recomendado proceder pelo menos até à 3ºFN.



Primeira Forma Normal

Uma relação na qual a intersecção de cada linha e coluna contem apenas 1 valor (A tabela não pode conter grupos repetidos).
A tabela apenas contém valores atómicos (valor que não pode ser dividido).

Primeiro começamos o processo de normalização por transferir os dados para a tabela. Neste formato, a tabela está desnormalizada. Para transforma a tabela para a primeira forma normal, identificamos e removemos os grupos repetidos da tabela. Um grupo repetido é um atributo ou grupo de atributos dentro da tabela que ocorrem com múltiplos valores para uma dada chave.

ClientRental

clientNo	cName	propertyNo	pAddress	rentStart	rentFinish	rent	ownerNo	oName
CR76	John Kay	PG4	6 Lawrence St, Glasgow	1-Jul-03	31-Aug-04	350	CO40	Tina Murphy
		PG16	5 Novar Dr, Glasgow	1-Sep-04	1-Sep-05	450	CO93	Tony Shaw
CR56	Aline Stewart	PG4	6 Lawrence St, Glasgow	1-Sep-02	10-June-03	350	CO40	Tina Murphy
		PG36	2 Manor Rd, Glasgow	10-Oct-03	1-Dec-04	375	CO93	Tony Shaw
		PG16	5 Novar Dr, Glasgow	1-Nov-05	10-Aug-06	450	CO93	Tony Shaw

Duas abordagens:

- Preencher as colunas vazias das linhas repetindo os dados. Por outras palavras, duplicar os dados onde for necessário.

ClientRental

clientNo	propertyNo	cName	pAddress	rentStart	rentFinish	rent	ownerNo	oName
CR76	PG4	John Kay	6 Lawrence St, Glasgow	1-Jul-03	31-Aug-04	350	CO40	Tina Murphy
CR76	PG16	John Kay	5 Novar Dr, Glasgow	1-Sep-04	1-Sep-05	450	CO93	Tony Shaw
CR56	PG4	Aline Stewart	6 Lawrence St, Glasgow	1-Sep-02	10-Jun-03	350	CO40	Tina Murphy
CR56	PG36	Aline Stewart	2 Manor Rd, Glasgow	10-Oct-03	1-Dec-04	375	CO93	Tony Shaw
CR56	PG16	Aline Stewart	5 Novar Dr, Glasgow	1-Nov-05	10-Aug-06	450	CO93	Tony Shaw

- Substituir os dados repetidos, junto com uma cópia da chave, numa relação em separado. Por vezes podem ocorrer vários grupos repetidos, pelo que o processo é repetido até não haver mais grupos.

Client

clientNo	cName
CR76	John Kay
CR56	Aline Stewart

PropertyRentalOwner

clientNo	propertyNo	pAddress	rentStart	rentFinish	rent	ownerNo	oName
CR76	PG4	6 Lawrence St, Glasgow	1-Jul-03	31-Aug-04	350	CO40	Tina Murphy
CR76	PG16	5 Novar Dr, Glasgow	1-Sep-04	1-Sep-05	450	CO93	Tony Shaw
CR56	PG4	6 Lawrence St, Glasgow	1-Sep-02	10-Jun-03	350	CO40	Tina Murphy
CR56	PG36	2 Manor Rd, Glasgow	10-Oct-03	1-Dec-04	375	CO93	Tony Shaw
CR56	PG16	5 Novar Dr, Glasgow	1-Nov-05	10-Aug-06	450	CO93	Tony Shaw

Apesar de ambas as abordagens estarem corretas, a primeira introduz mais redundancia, enquanto que a segunda cria mais relações com menos redundancia (2ºabordagem é melhor).

Segunda Forma Normal

A segunda forma normal é baseada no conceito de dependencia funcional completa. A 2FN é aplicada a relações com atributos chave compostos. Uma relação com um atributo chave único (apenas um atributo) está automaticamente em 2NF.
Assim, uma relação que esteja na segunda forma normal, tem de estar na primeira forma normal e cada atributo não chave deve estar funcionalmente completamente dependente da chave primária.

Abordagem:
- Remover dependências parciais, criando novas relações em que todos os atributos sejam completamente dependentes da chave primária.

ClientRental

clientNo	propertyNo	cName	pAddress	rentStart	rentFinish	rent	ownerNo	oName
CR76	PG4	John Kay	6 Lawrence St, Glasgow	1-Jul-03	31-Aug-04	350	CO40	Tina Murphy
CR76	PG16	John Kay	5 Novar Dr, Glasgow	1-Sep-04	1-Sep-05	450	CO93	Tony Shaw
CR56	PG4	Aline Stewart	6 Lawrence St, Glasgow	1-Sep-02	10-Jun-03	350	CO40	Tina Murphy
CR56	PG36	Aline Stewart	2 Manor Rd, Glasgow	10-Oct-03	1-Dec-04	375	CO93	Tony Shaw
CR56	PG16	Aline Stewart	5 Novar Dr, Glasgow	1-Nov-05	10-Aug-06	450	CO93	Tony Shaw

Client

clientNo	cName
CR76	John Kay
CR56	Aline Stewart

Rental

clientNo	propertyNo	rentStart	rentFinish
CR76	PG4	1-Jul-03	31-Aug-04
CR76	PG16	1-Sep-04	1-Sep-05
CR56	PG4	1-Sep-02	10-Jun-03
CR56	PG36	10-Oct-03	1-Dec-04
CR56	PG16	1-Nov-05	10-Aug-06

PropertyOwner

propertyNo	pAddress	rent	ownerNo	oName
PG4	6 Lawrence St, Glasgow	350	CO40	Tina Murphy
PG16	5 Novar Dr, Glasgow	450	CO93	Tony Shaw
PG36	2 Manor Rd, Glasgow	375	CO93	Tony Shaw

Terceira Forma Normal

Se existe uma dependência transitiva dentro da mesma tabela, tem de ser eliminada.
Uma relação está na 3FN se está na 1FN e 2FN e na qual nenhum atributo não chave se encontra transitivamente dependente da chave primária.

Se existe uma dependencia transitiva, removemos o atributo transitivamente dependente da relação e colocamo-lo numa nova relação juntamente com uma cópia do atributo determinante.

PropertyOwner

propertyNo	pAddress	rent	ownerNo	oName
PG4	6 Lawrence St, Glasgow	350	CO40	Tina Murphy
PG16	5 Novar Dr, Glasgow	450	CO93	Tony Shaw
PG36	2 Manor Rd, Glasgow	375	CO93	Tony Shaw

PropertyForRent

propertyNo	pAddress	rent	ownerNo
PG4	6 Lawrence St, Glasgow	350	CO40
PG16	5 Novar Dr, Glasgow	450	CO93
PG36	2 Manor Rd, Glasgow	375	CO93

Owner

ownerNo	oName
CO40	Tina Murphy
CO93	Tony Shaw