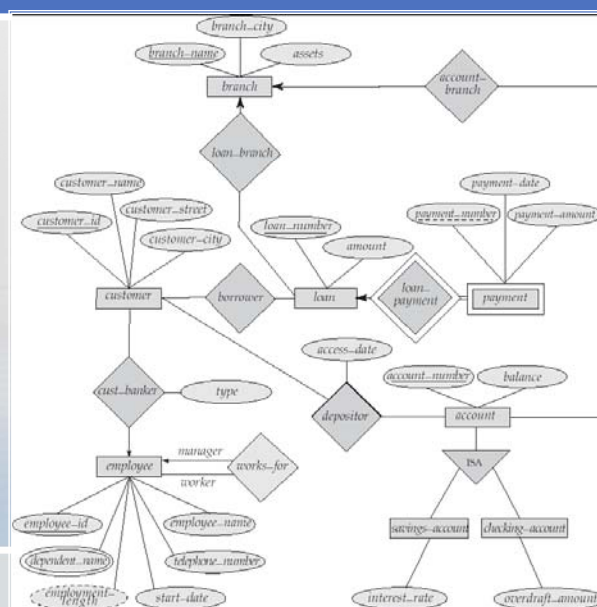


Bases de Dados

Conversão para Modelo Relacional

Modelo Entidade-Associação



Conversão de entidades

- Entidade forte converte-se numa tabela

loan = (loan_number, amount)

- atributos simples mantêm-se
- chave é a mesma da entidade

Conversão de entidades

- Entidade fraca inclui uma coluna extra com a chave primária da entidade identificadora

*payment = (loan_number, payment_number,
payment_date, payment_amount)*

- dispensa conversão da associação identificadora

Conversão de associações

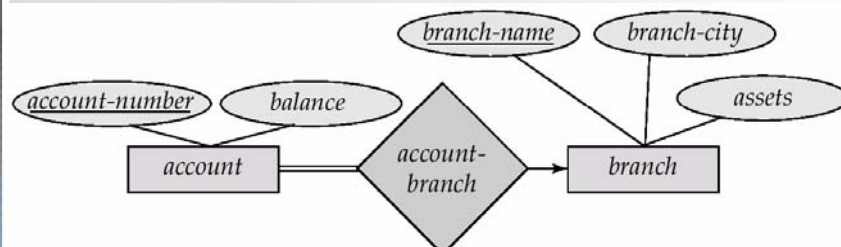
- Associação “muitos-para-muitos” converte-se numa tabela com chaves primárias e atributos descritivos

borrower = (*customer_id*, *loan_number*)

- chave primária da tabela é conjunto das chaves
- Associações *n*-árias são tratadas de modo semelhante
 - se houver seta essa entidade não faz parte da chave

Conversão de associações

- Associações “um-para-muitos” ou “muitos-para-um”
 - pode não ser necessário criar uma tabela
 - exemplo
 - existe participação total do lado “muitos”
 - adicionar o atributo *branch_name* à tabela *account*



Conversão de associações

- Associação “um-para-um”
 - pode não ser necessário criar uma tabela
 - qualquer um dos lados pode ter uma chave estrangeira
 - se a participação for parcial, aparecem *nulls*

Conversão de atributos

- Atributos compostos
 - representados como atributos individuais
 - exemplo
 - *customer* com *name* composto (*first_name* e *last_name*)
 - resulta em *name.first_name* e *name.last_name*

Conversão de atributos

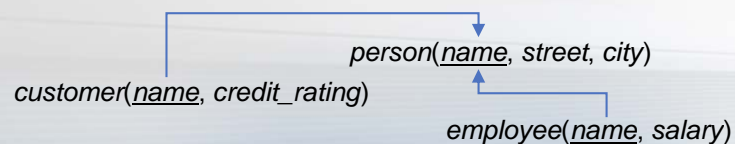
- Atributos multi-valor
 - necessária tabela adicional \approx “um-para-muitos”
 - chave primária da entidade E e valor do atributo M
 - exemplo: dependentes

employee_dependent_names = (employee_id, dname)

- para um funcionário com chave primária 123-45-6789 e dependentes *João* e *Teresa*
 - ('123-45-6789', 'João') e ('123-45-6789', 'Teresa')

Conversão de especialização/generalização

- Método 1
 - tabela para a entidade genérica
 - tabela para cada entidade específica, com chave para a entidade genérica



- desvantagem: informação dispersa

Conversão de especialização/generalização

▪ Método 2

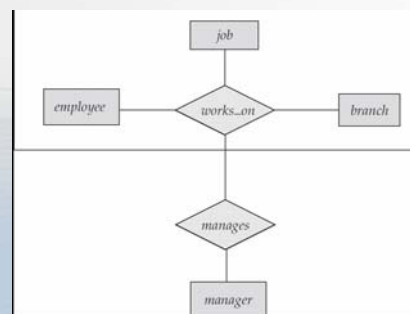
- repetir os atributos em cada tabela

person(name, street, city)
customer(name, street, city, credit_rating)
employee(name, street, city, salary)

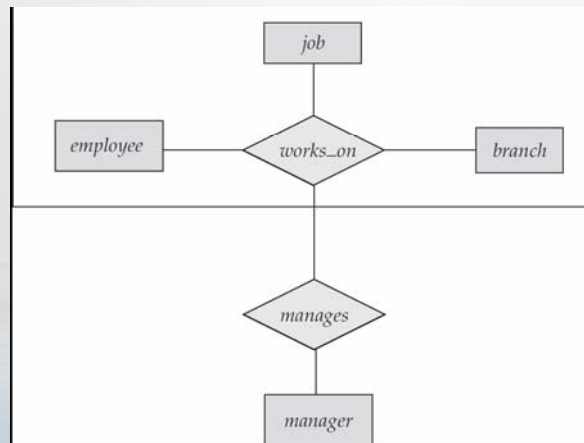
- se a especialização for total, super-classe parece dispensável
 - mas continua a ser necessária para chaves estrangeiras!
- desvantagem: redundância

Conversão de agregação

- Criar uma nova tabela, contendo:
 - chave primária da associação agregada
 - chave primária da entidade associada
 - atributos descritivos

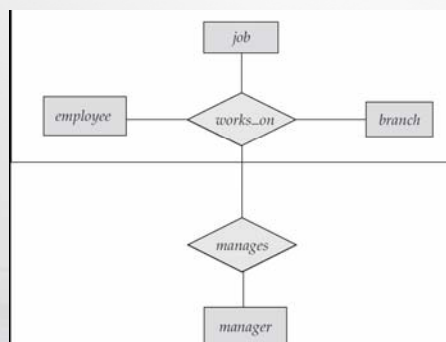


Conversão de agregação



manages (employee_id, branch_name, title, manager_name)

Conversão de agregação



- Podia evitar-se uma nova tabela guardando *manager_name* em *works_on*
 - admitindo que *manager_name* pode ser *null*

Bases de Dados

Modelo Relacional

Bases de dados relacionais

- Estrutura de uma base de dados relacional
 - conjunto de tabelas com nome único
 - cada tabela guarda entidades desse tipo
- Dentro das tabelas
 - cada coluna é um atributo da entidade
 - cada linha contém os valores desses atributos
- Cada tabela estabelece uma relação entre um conjunto de valores

IST • DEI • Bases de Dados 16

Exemplo

- A relação “conta bancária” (*account*)
 - exemplo de atributos: número, agência, saldo

<i>account_number</i>	<i>branch_name</i>	<i>balance</i>
A-101	Downtown	500
A-102	Perryridge	400
A-201	Brighton	900
A-215	Mianus	700
A-217	Brighton	750
A-222	Redwood	700
A-305	Round Hill	350

Atributos

- Cada atributo tem um domínio
 - o conjunto de valores possíveis
- Para o exemplo anterior
 - agência – conjunto de todas as localidades
 - cada linha é um tuplo (v_1, v_2, v_3)
 - cada v_i tem domínio D_i
- Uma relação contém apenas um sub-conjunto de
 $D_1 \times D_2 \times \dots \times D_n$

Tuplos

<i>account_number</i>	<i>branch_name</i>	<i>balance</i>
A-101	Downtown	500
A-102	Perryridge	400
A-201	Brighton	900
A-215	Mianus	700
A-217	Brighton	750
A-222	Redwood	700
A-305	Round Hill	350

- A relação contém 7 tuplos
- Para o primeiro tuplo t
 - $t[\text{account_number}] = \text{"A-101"}$
 - $t[\text{branch_name}] = \text{"Downtown"}$

Esquema

- Esquema vs. instância
 - conteúdo da relação num determinado instante
 - analogia com linguagens de programação
 - relação \approx variável
 - esquema da relação \approx tipo

- Esquema da relação

<i>account_number</i>	<i>branch_name</i>	<i>balance</i>
A-101	Downtown	500
A-102	Perryridge	400
A-201	Brighton	900
A-215	Mianus	700
A-217	Brighton	750
A-222	Redwood	700
A-305	Round Hill	350

$\text{account_schema} = (\text{account_number}, \text{branch_name}, \text{balance})$

A relação “agência” (*branch*)

branch_schema = (*branch_name*, *branch_city*, *assets*)

<i>branch_name</i>	<i>branch_city</i>	<i>assets</i>
Brighton	Brooklyn	7100000
Downtown	Brooklyn	9000000
Mianus	Horseneck	400000
North Town	Rye	3700000
Perryridge	Horseneck	1700000
Pownal	Bennington	300000
Redwood	Palo Alto	2100000
Round Hill	Horseneck	8000000

Atributos comuns

- *branch_name* é um atributo comum
 - forma de relacionar tuplos de relações distintas

<i>account_number</i>	<i>branch_name</i>	<i>balance</i>
A-101	Downtown	500
A-102	Perryridge	400
A-201	Brighton	900
A-215	Mianus	700
A-217	Brighton	750
A-222	Redwood	700
A-305	Round Hill	350

<i>branch_name</i>	<i>branch_city</i>	<i>assets</i>
Brighton	Brooklyn	7100000
Downtown	Brooklyn	9000000
Mianus	Horseneck	400000
North Town	Rye	3700000
Perryridge	Horseneck	1700000
Pownal	Bennington	300000
Redwood	Palo Alto	2100000
Round Hill	Horseneck	8000000

A relação “cliente” (*customer*)

customer_schema = (*customer_name*, *customer_street*, *customer_city*)

<i>customer_name</i>	<i>customer_street</i>	<i>customer_city</i>
Adams	Spring	Pittsfield
Brooks	Senator	Brooklyn
Curry	North	Rye
Glenn	Sand Hill	Woodside
Green	Walnut	Stamford
Hayes	Main	Harrison
Johnson	Alma	Palo Alto
Jones	Main	Harrison
Lindsay	Park	Pittsfield
Smith	North	Rye
Turner	Putnam	Stamford
Williams	Nassau	Princeton

A relação “titular” (*depositor*)

depositor_schema = (*customer_name*, *account_number*)

<i>customer_name</i>	<i>account_number</i>
Hayes	A-102
Johnson	A-101
Johnson	A-201
Jones	A-217
Lindsay	A-222
Smith	A-215
Turner	A-305

A relação “empréstimo” (*loan*)

loan_schema = (*loan_number*, *branch_name*, *amount*)

<i>loan_number</i>	<i>branch_name</i>	<i>amount</i>
L-11	Round Hill	900
L-14	Downtown	1500
L-15	Perryridge	1500
L-16	Perryridge	1300
L-17	Downtown	1000
L-23	Redwood	2000
L-93	Mianus	500

A relação “devedor” (*borrower*)

borrower_schema = (*customer_name*, *loan_number*)

<i>customer_name</i>	<i>loan_number</i>
Adams	L-16
Curry	L-93
Hayes	L-15
Jackson	L-14
Jones	L-17
Smith	L-11
Smith	L-23
Williams	L-17

Notas

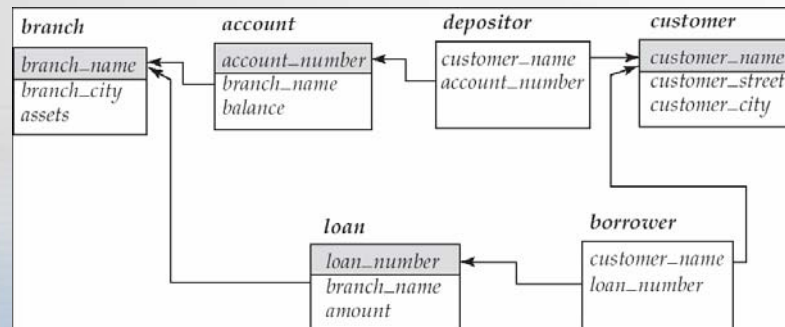
- A ordem dos tuplos é irrelevante
- Podiam existir valores *null*
- Assume-se que os clientes não têm nomes iguais
- Assume-se que pode haver clientes com conta mas sem empréstimo, ou vice-versa, ou nenhuma das coisas

Chaves

- Como distinguir tuplos?
 - não pode haver 2 tuplos exactamente iguais
 - super-chave
 - o conjunto de atributos que identificam univocamente
 - chave candidata
 - conjunto mínimo de atributos
 - chave primária
 - a chave candidata escolhida
 - normalmente prefere-se um atributo separado por si só
 - atenção aos atributos (supostamente) únicos

Chaves estrangeiras

- Chave estrangeira
 - referência entre relações



Restrições de integridade

- As restrições de integridade garantem a consistência da BD
 - exemplos
 - RI 1: o saldo de uma conta não pode ser *null*
 - RI 2: nenhuma conta pode ter o mesmo número de outra
 - RI 3: todos os números de conta em *depositor* tem que existir em *account*
 - RI 4: o salário de um funcionário tem que ser pelo menos 5€/hora
 - ...