

ORACLE

Academy

Database Design

9-3

Mapeamento de Relacionamentos

ORACLE
Academy



Objetivos

- Esta lição abrange os seguintes objetivos:
 - Aplicar a regra de mapeamento de relacionamentos para transformar corretamente relacionamentos 1:M e barrados
 - Aplicar a regra de mapeamento de relacionamentos para transformar corretamente relacionamentos M:M
 - Transformar relacionamentos 1:1
 - Aplicar a regra de mapeamento de relacionamentos para transformar corretamente relacionamentos em um arco

Finalidade

- Suponha que você esteja construindo uma casa para alguém
- Você tem todos os materiais – madeira, tinta, portas, janelas, pregos, parafusos, etc. – e as habilidades, mas não tem um projeto
- Ao começar, você não sabe quantos quartos deve incluir, onde as janelas devem ficar, o sentido das portas nem a cor que cada quarto deve ser pintado

Finalidade

- Você pode construir uma casa dessa forma e tomar essas decisões durante o processo, mas se você não começar com um plano do projeto estrutural, o produto final poderá não ser a casa que o cliente tem em mente



Finalidade

- Os relacionamentos são mapeados entre chaves primárias e estrangeiras para permitir que uma tabela faça referência a outra
- Se não mapearmos os relacionamentos, só teremos várias tabelas independentes com informações que não se conectam a nada no banco de dados
- Mapear relacionamentos entre entidades serve como um "primeiro passo" essencial para facilitar a discussão entre o cliente, o designer, o desenvolvedor e o administrador do produto de banco de dados

Regras para Relacionamentos

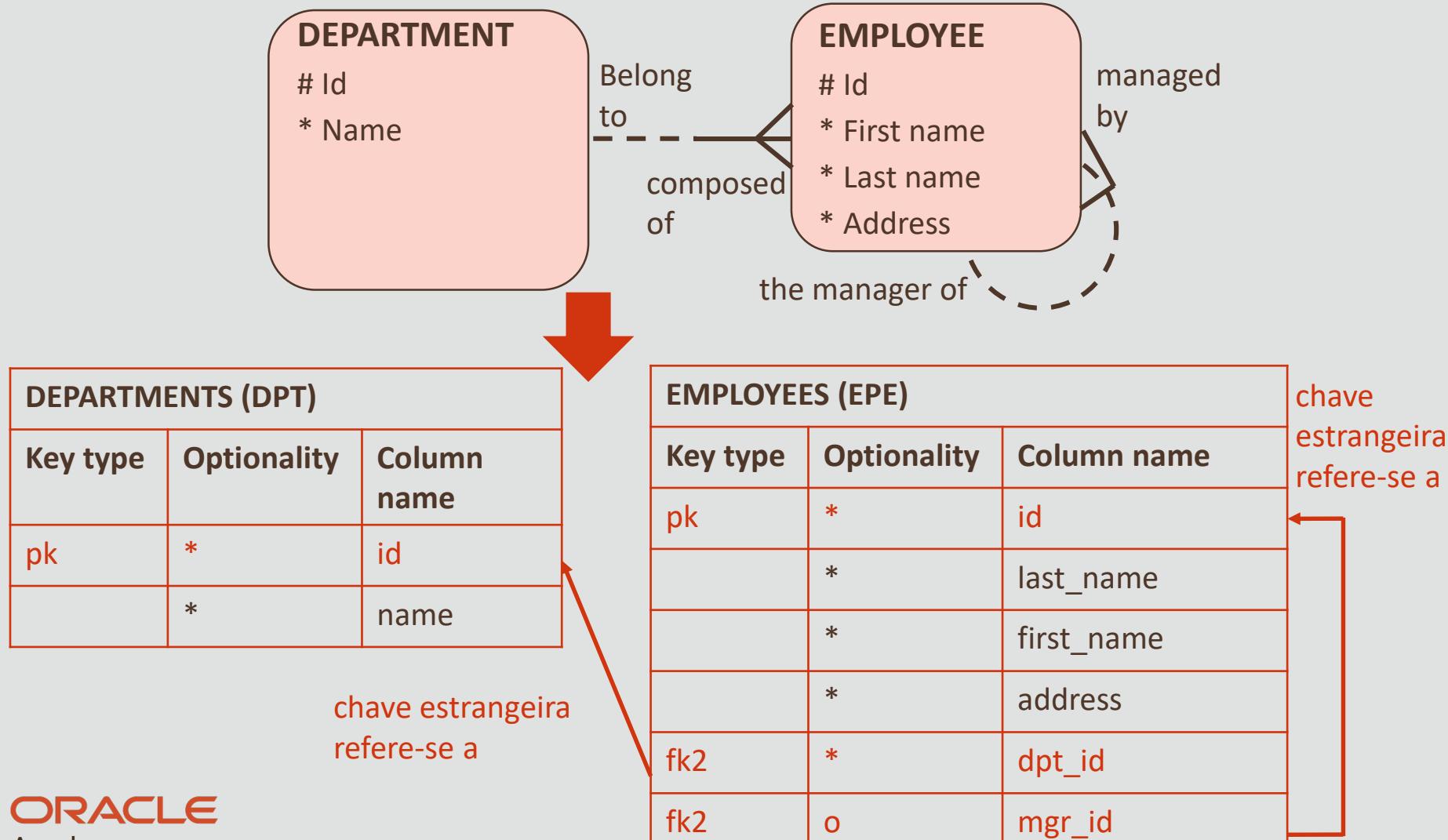
- Um relacionamento cria uma ou mais colunas de chave estrangeira na tabela no lado muitos do relacionamento
- Usamos o nome curto da tabela para nomear a coluna de chave estrangeira
- No exemplo na próxima página, a coluna de chave estrangeira na tabela EMPLOYEES é dpt_id para o relacionamento com DEPARTMENT, e mgr_id para o relacionamento recursivo com ele mesmo

Regras para Relacionamentos

- A coluna de chave estrangeira pode ser obrigatória ou opcional dependendo das necessidades da empresa
- No exemplo, dpt_id é obrigatório e mgr_id é opcional



Ilustração das Regras para Relacionamentos



Mapeamento do Relacionamento Obrigatório no Lado Um

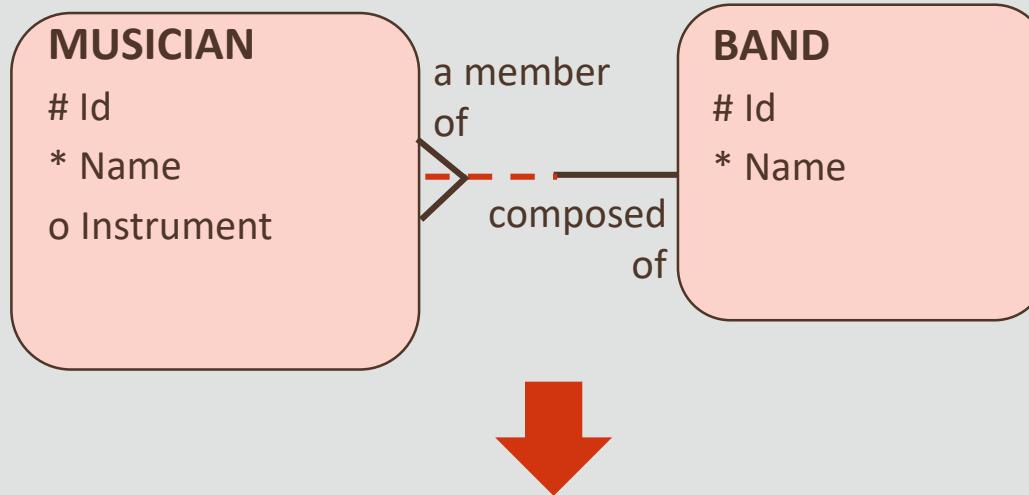
- Os relacionamentos que são obrigatórios no lado um ou obrigatórios nos dois lados são mapeados exatamente da mesma forma que um relacionamento que seja opcional no lado um
- O modelo conceitual é completo o suficiente para capturar a optionalidade nos dois lados do relacionamento
- No entanto, o modelo físico é limitado no sentido de que a restrição de chave estrangeira pode reforçar um relacionamento obrigatório somente no lado muitos

Mapeamento do Relacionamento Obrigatório no Lado Um

- No exemplo a seguir, o modelo físico não pode reforçar que uma BANDA deva ser composta por pelo menos um MÚSICO
- A opcionalidade na ponta um precisará ser implementada por programação adicional



Aplicando a Opcionalidade



MUSICIANS (MSN)		
Key type	Optionality	Column name
pk	*	id
	*	name
	o	instrument
fk	o	bad_id

BANDS (BAD)		
Key type	Optionality	Column name
pk	*	id
	*	name

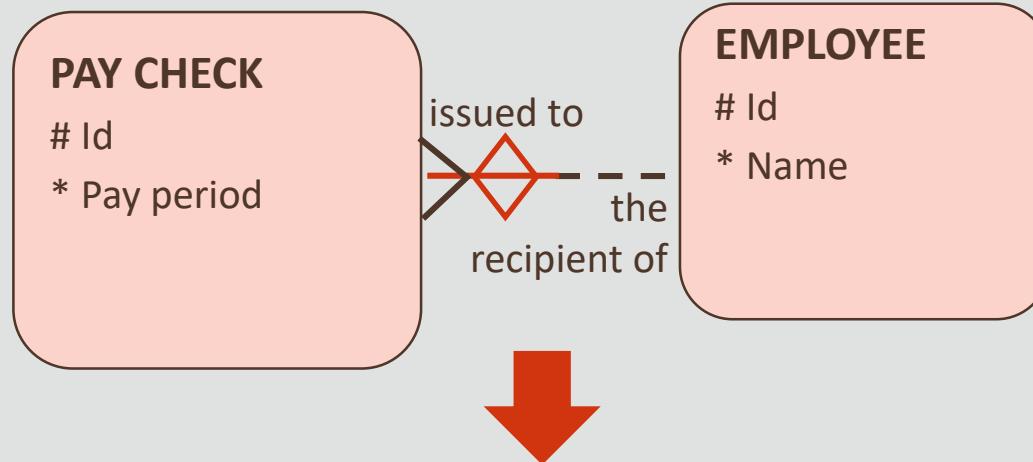
chave
estrangeira
refere-se a



Mapeamento de Relacionamentos Intransferíveis

- Um relacionamento intransferível no modelo conceitual significa que a coluna de chave estrangeira na tabela de banco de dados não pode ser atualizada
- A restrição de chave estrangeira isolada não pode reforçar isso no banco de dados
- Será necessária programação adicional para garantir que o banco de dados siga essa regra de negócios
- É importante documentar regras como essa para que a equipe se lembre de escrever o código adequado e aplicar essa regra de negócios

Reforçando Relacionamentos Intransferíveis



PAYCHECKS (PCK)		
Key type	Optionality	Column name
pk	*	id
	*	name
fk	*	epe_id

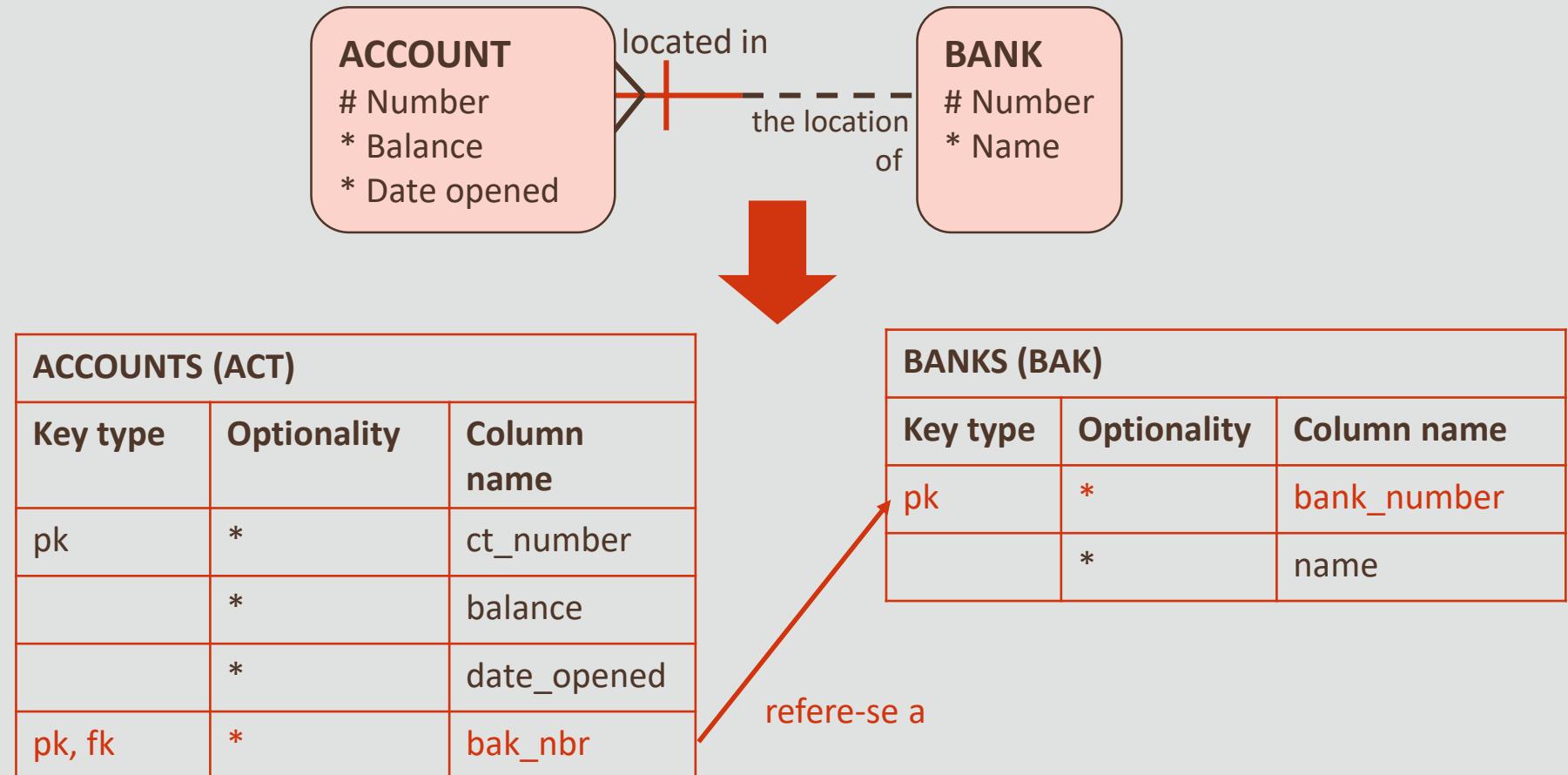
EMPLOYEES (EPE)		
Key type	Optionality	Column name
pk	*	id
	*	name

o valor nessa coluna de chave estrangeira não pode ser alterado

Mapeamento de Relacionamentos Barrados

- Um relacionamento barrado é mapeado para uma coluna de chave estrangeira no lado muitos, como qualquer outro relacionamento 1:M
- Nesse caso, a coluna de chave estrangeira exerce dois papéis porque também faz parte da chave primária
- No exemplo, `bak_number` é uma coluna de chave estrangeira em `ACCOUNTS` que se refere à chave primária de `BANKS`
- Também faz parte da chave primária de `ACCOUNTS`

Mapeamento de Relacionamentos Barrados





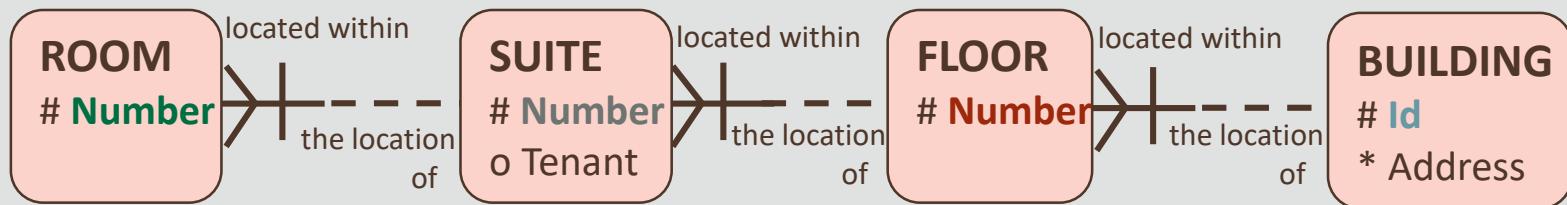
Relacionamentos Barrados em Cascata

- Hierarquias podem levar a relacionamentos barrados em cascata, em que o UID da entidade na parte superior da hierarquia é transferida para o UID da entidade na parte inferior da hierarquia
- No exemplo, o UID de ROOM é composto pelo número de ROOM, número de SUITE, número de FLOOR e id de BUILDING
- Isso é representado pelos relacionamentos barrados

Relacionamentos Barrados em Cascata

- Quando isso é mapeado para um modelo físico, o resultado pode ser um nome muito longo de coluna de chave estrangeira porque ela usa os nomes curtos das tabelas originais como prefixo
- A convenção sugerida é nunca usar mais que dois prefixos de tabela
- No exemplo a seguir, a coluna de chave estrangeira em ROOMS que aparece a todo mundo desde BUILDINGS é denominada sue_bdg_id, em vez de sue_flr_bdg_id

Relacionamentos Barrados em Cascata



ROOMS (ROM)		
pk	*	rom_nbr
pk, fk	*	sue_nbr
pk, fk	*	flr_nbr
pk, fk	*	flr_bdg_id

SUITES (SUE)		
pk	*	sue_nbr
pk, fk	*	flr_nbr
pk, fk	*	flr_bdg_id
	o	tenant

FLOORS (FLR)		
pk	*	flr_nbr
pk, fk	*	bdg_id

BUILDINGS (BDG)		
pk	*	id
	*	address

Ilustração de Relacionamentos Barrados em Cascata

- Os exemplos de dados para cada tabela ilustram os relacionamentos barrados em cascata

BUILDINGS

id	address
100	40 Potters Lane
201	57G Maricopa Way

FLOORS

flr_nbr	bdg_id
1	100
2	100
1	201
2	201

SUITES

sue_nbr	flr_nbr	flr_bdg
15	2	100
25	2	100
5E	1	201
7B	2	201

ROOMS

rom_nbr	sue_nbr	sue_flr_nbr	sue_bdg_id
1	15	2	100
2	15	2	100
1	7B	2	201

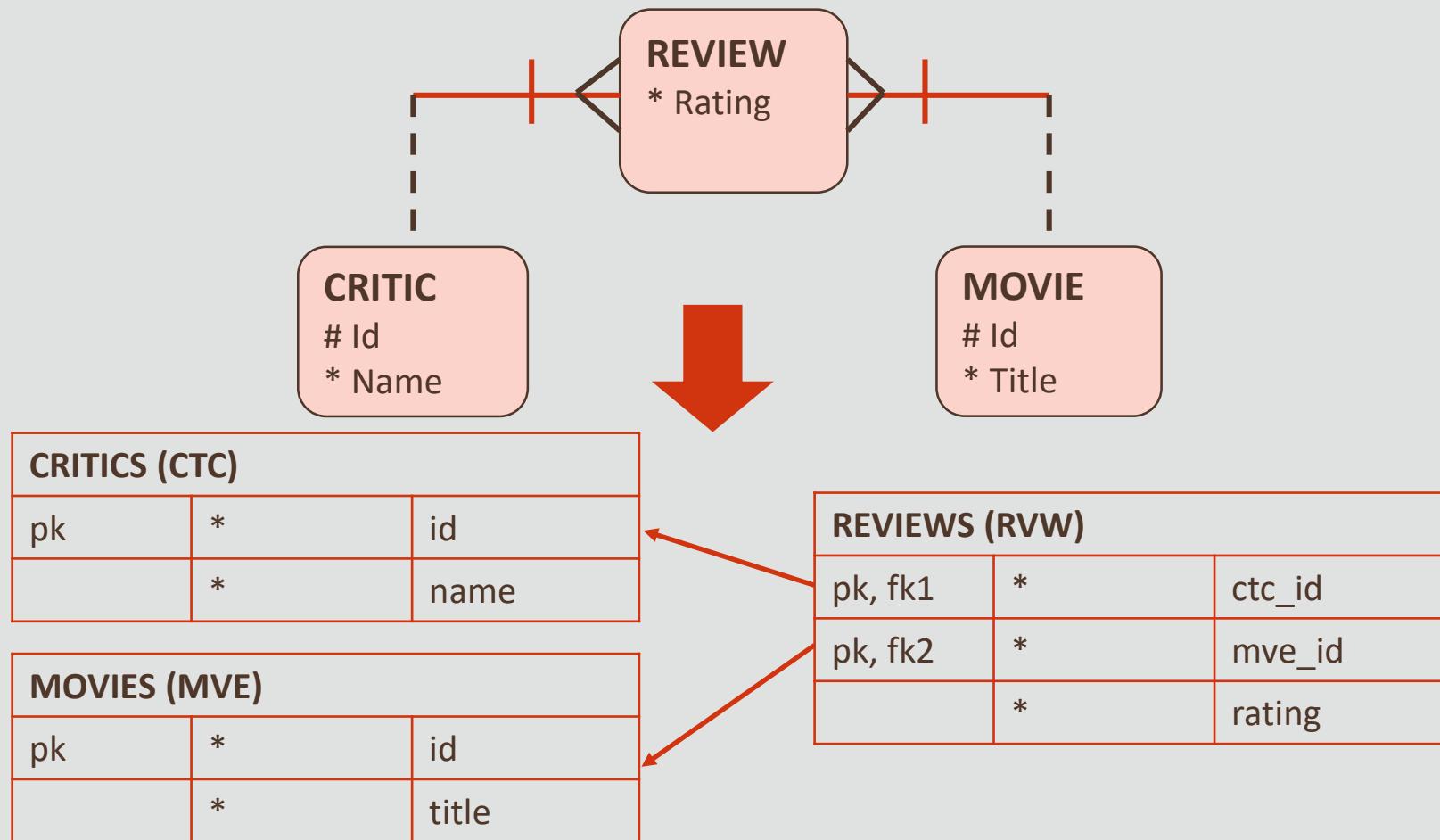


Mapeando Relacionamentos Muitos para Muitos

- Um relacionamento M:M é resolvido com uma entidade de intersecção, que é mapeada para uma tabela de intersecção
- Essa tabela de intersecção conterá colunas de chave estrangeira que se referem às tabelas originais
- No exemplo, REVIEWS contém todas as combinações que existem entre um CRITIC e um MOVIE



Mapeando Relacionamentos Muitos para Muitos

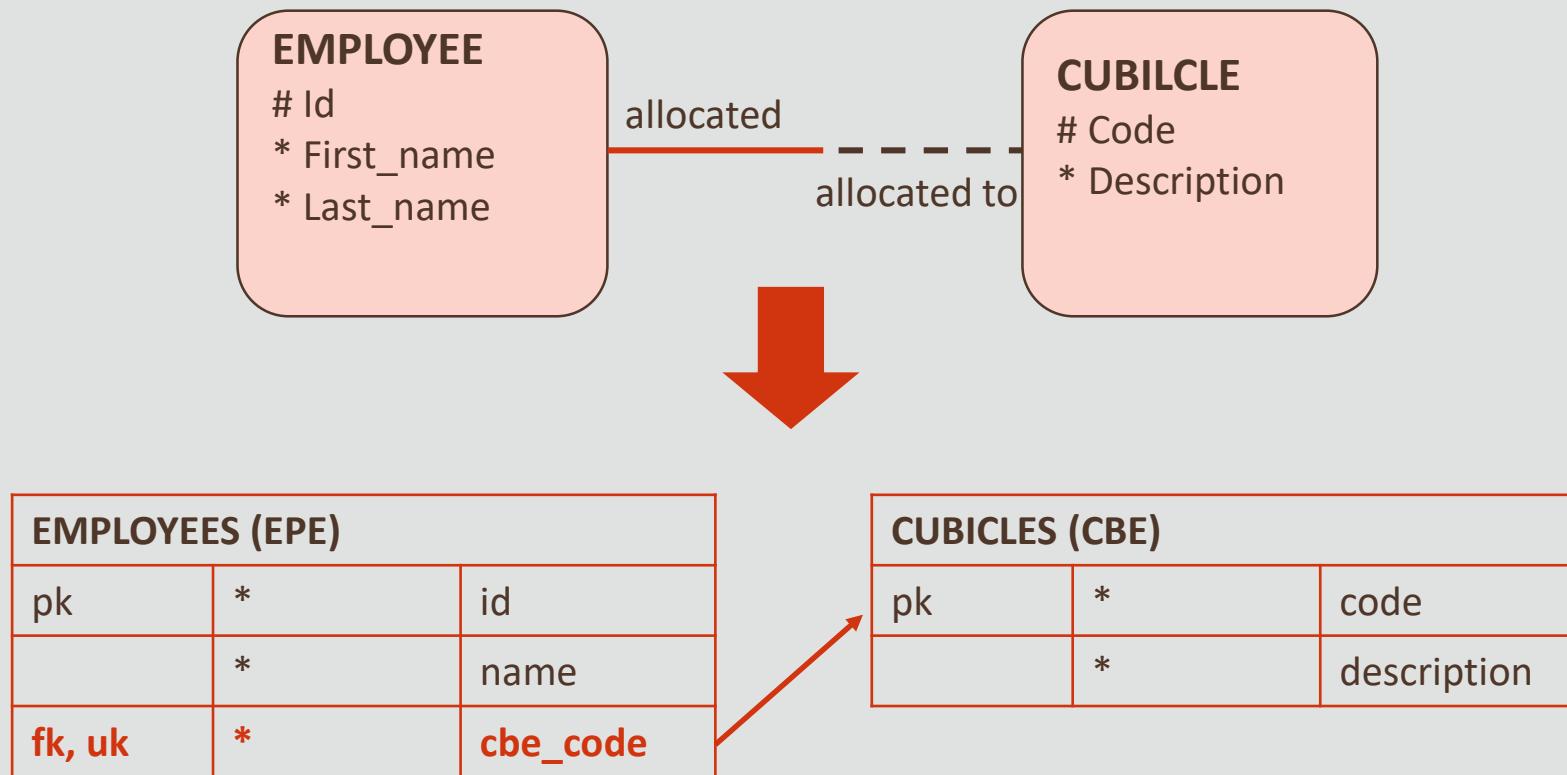




Mapeando Relacionamentos Um para Um

- Ao transformar um relacionamento 1:1, crie uma chave estrangeira e uma chave exclusiva
- Todas as colunas dessa chave estrangeira também fazem parte da chave exclusiva
- Se o relacionamento for obrigatório no lado um, a chave estrangeira será criada na tabela correspondente
- No exemplo, `cbe_code` é a coluna de chave estrangeira em `EMPLOYEES` que se refere à chave primária de `CUBICLES`
- `Cbe_code` também seria exclusivo na tabela `EMPLOYEES`

Mapeando Relacionamentos Um para Um



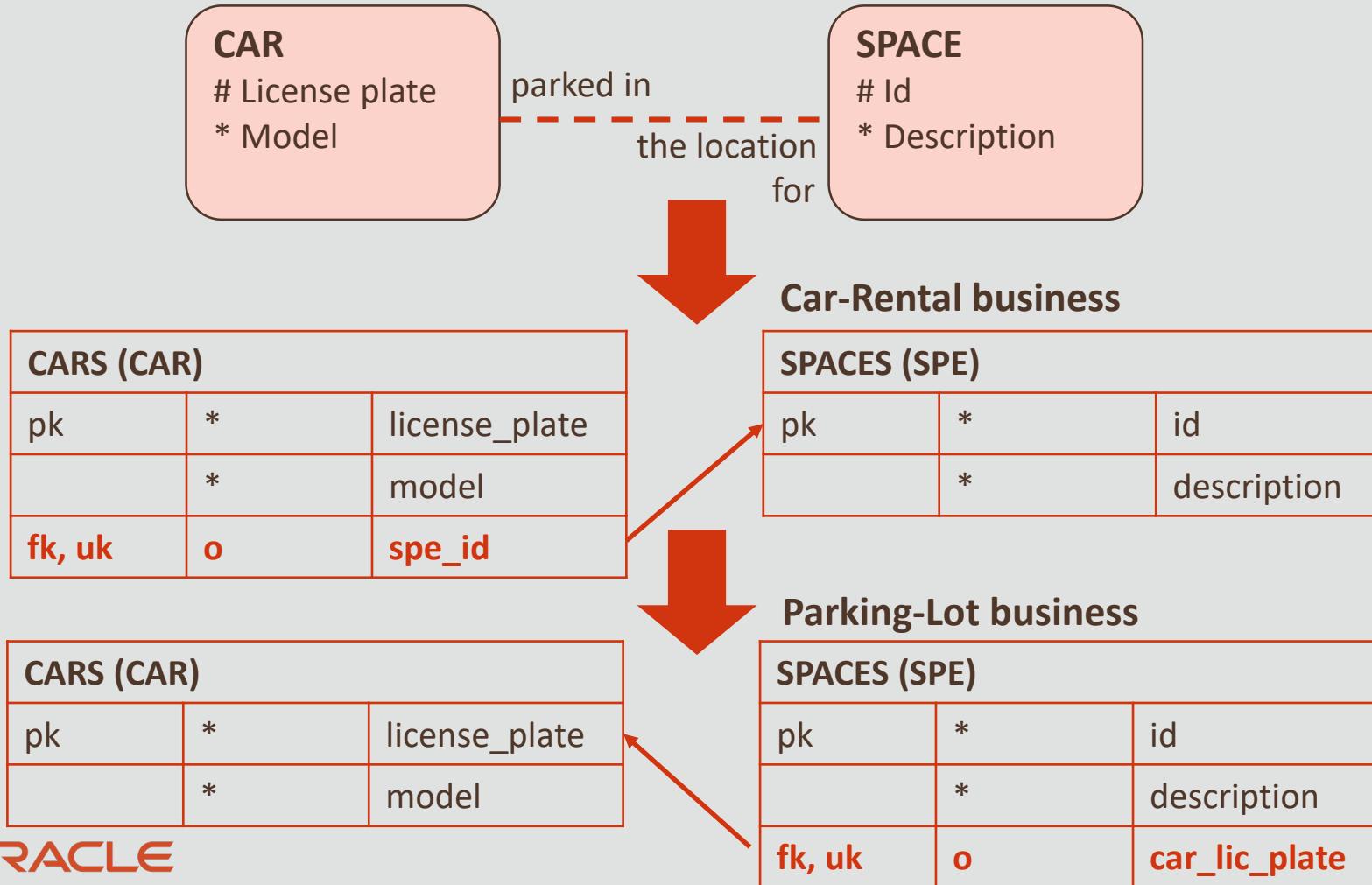
Um para Um Opcional

- Se o relacionamento for opcional nos dois lados, você poderá escolher quais tabelas obtêm a chave estrangeira
- Não há regras absolutas, mas há algumas diretrizes:
 - Implemente a chave estrangeira na tabela que tenha menos linhas para economizar espaço
 - Implemente a chave estrangeira onde ela faça mais sentido para a empresa

Um para Um Opcional

- No exemplo, uma agência de locação de automóveis estaria mais preocupada com carros do que com espaço, então faz sentido colocar a chave estrangeira em CARS
- No entanto, em uma empresa de estacionamento, o principal objeto é o espaço de estacionamento
- Dessa forma, faria sentido colocar a chave estrangeira em SPACES

Regras de Negócios para Um para Um Opcional



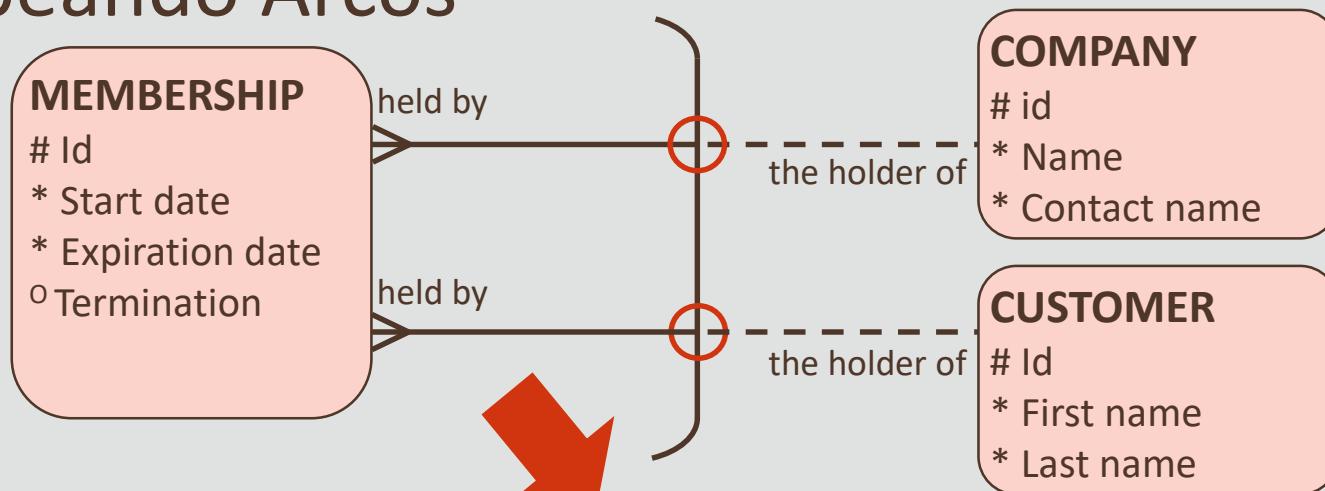
Aplicando Um para Muitos

- Se o relacionamento for obrigatório nas duas pontas, você terá a mesma limitação no banco de dados que um relacionamento 1:M que é obrigatório em uma ponta
- Dessa forma, você precisaria gravar código adicional para aplicá-lo

Mapeando Arcos

- A entidade que tiver o arco será mapeada para uma tabela que contenha chaves estrangeiras das tabelas no lado "um" dos relacionamentos
- Observe que, mesmo se os relacionamentos no arco forem obrigatórios no lado muitos, as chaves estrangeiras resultantes precisarão ser opcionais (porque uma delas sempre ficará em branco)

Mapeando Arcos



MEMBERSHIPS (MBP)		
pk	*	id
	*	start_date
	*	expiration_date
	o	termination
fk1	o	cpe_id
fk2	o	cms_id

COMPANIES (CPE)		
pk	*	id
	*	name
	*	contact_name
CUSTOMERS (CMS)		
pk	*	id
	*	first_name
	*	last_name



Mapeando Arcos

- Como o arco representa relacionamentos exclusivos, é necessário código adicional para reforçar que somente uma das chaves estrangeiras tenha um valor para cada linha da tabela
- Uma restrição de verificação armazenada no banco de dados pode fazer isso com facilidade
- No exemplo, o código para a restrição de verificação teria a seguinte aparência:
 - CHECK (pse_id não é nulo E phe_id é nulo)
 - OR (pse_id é nulo E phe_id não é nulo)
- Se os relacionamentos fosse totalmente opcionais, você adicionaria:
 - - OR (pse_id é nulo E phe_id é nulo)

Terminologia

- Estes são os principais termos usados nesta lição:
 - Relacionamentos barrados em cascata
 - Entidade de intersecção
 - Relacionamento intransferível

Resumo

- Nesta lição, você deverá ter aprendido a:
 - Aplicar a regra de mapeamento de relacionamentos para transformar corretamente relacionamentos 1:M e barrados
 - Aplicar a regra de mapeamento de relacionamentos para transformar corretamente relacionamentos M:M
 - Transformar relacionamentos 1:1
 - Aplicar a regra de mapeamento de relacionamentos para transformar corretamente relacionamentos em um arco

ORACLE

Academy