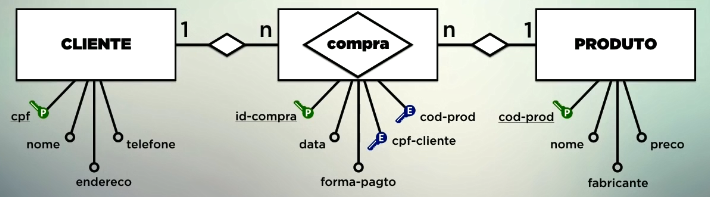
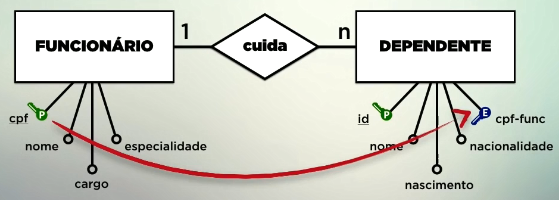
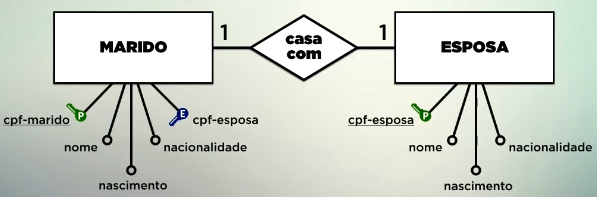
Tabela = Entidade | Colunas = Atributos OR Campos



**D**iagrama **E**ntidade **R**elacionamento “**DER**” É Representado por um **Losango**

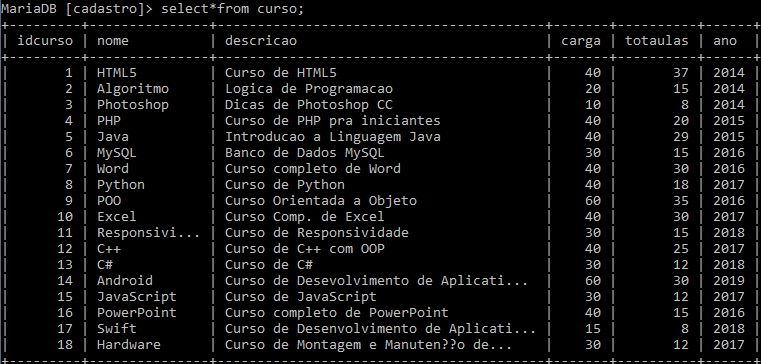
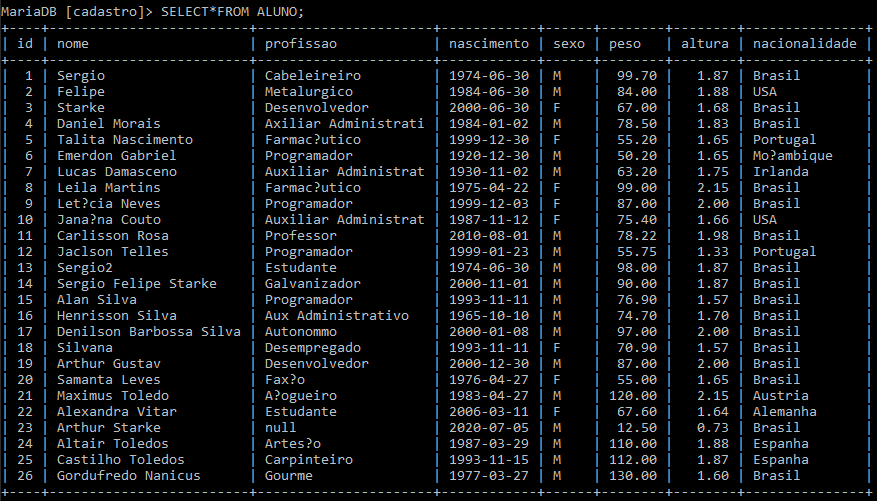
**Cardinalidade:** Nela, é definida o grau de relação entre duas entidades. Ex: **1:N, N:N, 1:1. Um para muito, muito para muito e um para um.**

Regra para Adição da Chave Estrangeira na Entidade Dominante

**** Há bancos de dados que são de Relacionamento; Duplos, Ternário, Quartenário, Multiplos...

IMPORTANTE: A implementação da chave estrangeira quando **1:1** é opcional. **1:N** vai para a entidade muito. **N:N** é “criado nova entidade” o qual recebera as chaves estrangeiras.

A chave ESTRANGEIRA, não necessariamente precisa ter o mesmo nome. Contudo, **DEVERÁ** ser do mesmo, Tipo e Quantidade, da Chave Primária da outra Entidade.

Princípio Fundamental de Banco de Dados, Regras Transacionais do **ACID;**

**A**TOMICIDADE:A Execução da Tarefa é Completa “Do Início ao Fim”.

**C**ONSISTÊNCIA:Os Dados Permanecem Consistentes após a Transação;

**I**SOLAMENTO: Permite o Acesso ao Mesmo Dado, Por Vários Usuários sem Interferência.

**D**URABILIDADE: Que Garanta a Persistência dos Dados Alocados.

Implementar Chave Estrangeira

IMPORTANTE: A chave ESTRANGEIRA, não necessariamente precisa ter o mesmo nome. Contudo, **DEVERÁ** ser do mesmo, Tipo e Quantidade, da Chave Primária da outra Entidade. \*RK: RecebeKey. \*Est: Estrangeira. \*Prim: Primára.

1º Adicionar Atributo”Coluna”:

ALTER TABLE n\_tabelaRK ADD COLUMN n\*coluna INT AFTER n\_coluna”Desejado”;



2º “Implementar” como Chave Estrangeira:

ALTER TABLE n\_tabelaRK **ADD FOREIGN KEY** (n\*coluna) **REFERENCES** **n\_tebelaEst**(**n\_colunaPrim**);



Implementar Conteúdo Esterno

UPDATE n\_tabelaRK SET n\*colunaEst = ‘?’ WHERE n\_colunaPri = ‘?’;



**INNER** JOIN ON

SELECT FROM JOIN ON

Apresenta apenas as registros “linhas”, que estiverem compartilhando informações entre tabelas.

O termo INNER não se faz necessário, já será considerado um “**INNER**” JOIN.

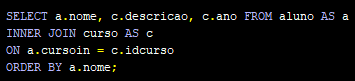
SELECT n\_tabelaRK.n\_campo, n\_tebelaEst.n\_campo FROM n\_tabelaRK **INNER** JOIN n\_tebelaEst ON n\_tebelaRK.n\_campoEst = n\_tabelaEst.n\_campoPrim;



AS

SELECT FROM JOIN ON

Abreviação das ENTIDADES. A expressão AS, não é obrigatório na sintaxe.





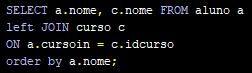
“LEFT or RIGHT” **OUTER** JOIN ON

SELECT FROM JOIN ON

“OUTER” É apresentado todo conteúdo da ENTIDADE desejada.

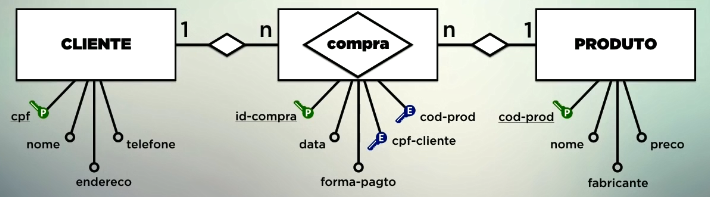
Nos exemplos abaixo, o 1º dará preferência a todos os nomes da entidade aluno. O 2º dará preferência aos nomes dos cursos da entidade curso.

Note que no 2º ex. NÃO fez uso do termo **OUTER**. O uso preferencial ”RIGHT or LEFT”, já subtende ser um “**OUTER**” JOIN.

Sintaxe para seleção **específica** da O.S. Onde ao invés do **nº** do Cliente, é o **Nome**



Criar Entidade **N:N** 

IMPORTANTE: A chave ESTRANGEIRA, não necessariamente precisa ter o mesmo nome. Contudo, **DEVERÁ** ser do mesmo, Tipo e Quantidade, da Chave Primária da outra Entidade. \*RK: RecebeKey. \*Est: Estrangeira. \*Prim: Primára.

1º Adicionar Entidade:

CREATE TABLE n\_tabelaRK (

n\*\_atributo INT NOT NULL AUTO\_INCREMENT,

n\_atributo DATE, “timestamp default current\_timestamp”

n\*atributoEst INT,

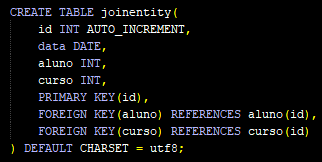
n\*atributoEst INT,

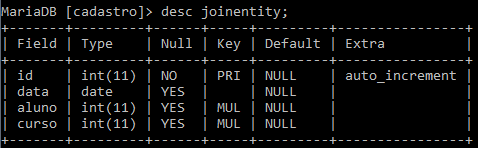
PRIMARY KEY (n\*\_atributo) ,

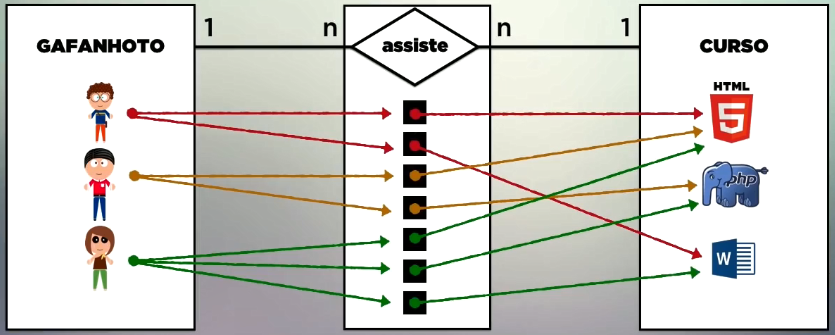
FOREIGN KEY (n\*atributoEst) REFERENCES n\_tebelaEst1 (n\_atributoPri) ,

FOREIGN KEY (n\*atributoEst) REFERENCES n\_tebelaEst2 (n\_atributoPri)

) DEFAULT CHARSET = utf8;





2º Implementar Relação

INSERT INTO n\_tabelaRK VALUES (DEFAULT,’?’,’?’,’?’);



3º Selec Relação

SELECT n\_tabelaRK.n\_campo, n\_tebelaEst.n\_campo FROM n\_tabelaRK

JOIN n\_tebelaEst1 ON n\_tebelaRK.n\_campoEst = n\_tabelaEst1.n\_campoPrim

JOIN n\_tebelaEst2 ON n\_tebelaRK.n\_campoEst = n\_tabelaEst2.n\_campoPrim;



