

# Bases de Dados e SQL

---

- DBMS - Database management system
  - SQL - Structured Query Language
  - Bases de dados relacionais
  - Modelo Entidade-Associação
  - Normalização
- 
- Recapitular a disciplina de bases de dados:
    - <http://fernandolobo.info/bd1617/>
    - <http://www.tomjewett.com/dbdesign/>
    - <http://www.tutorialspoint.com/dbms/>
    - <https://www.tutorialspoint.com/mysql/>

# Bases de dados relacionais

---

- Exemplo de uma base de dados relacional: notar que as tabelas estão relacionadas

Tabela de Produtores de vinho

Produtor ID	Nome	Endereço	Região ID
1	Terras del Rei	Estrada da Boa Pinga	3
2	Hardy Brothers	Jones St.	1
3	Penfolds	Arthurton Rd	1
4	Lindemans	Smith Av.	2
5	Orlando	Jones St.	1

PK → (pointing to Produtor ID)

FK (pointing to Região ID)

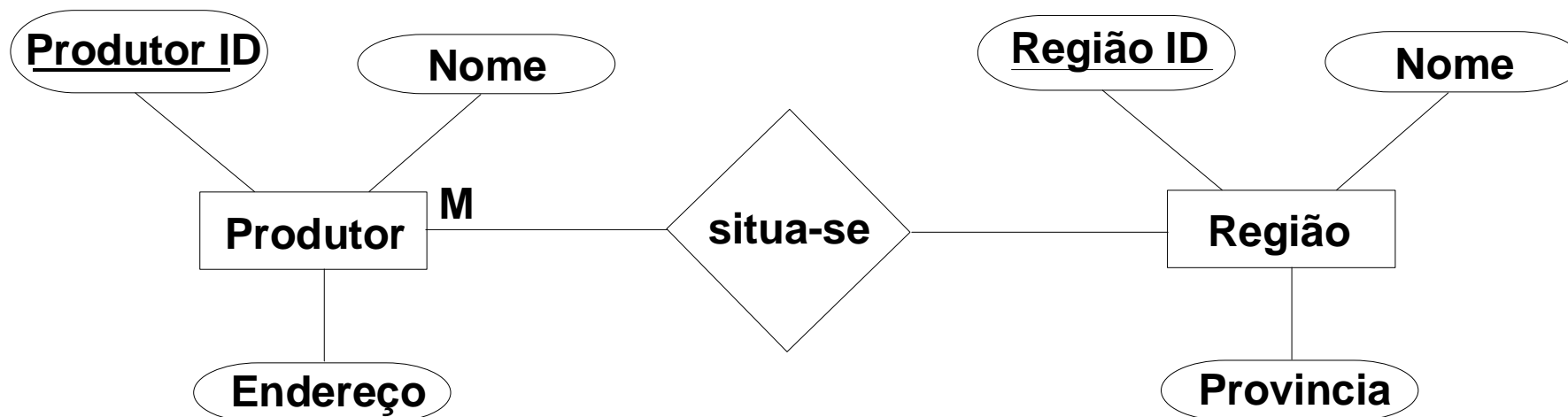
Tabela de Regiões

Região ID	Nome	Provincia
1	Barossa Valley	South Australia
2	Margaret River	Western Australia
3	Monsaraz	Alentejo

PK → (pointing to Região ID)

# Modelo Entidade-Associação

---



- Entidades (Produtor, Região)
- Atributos (Nome, Endereço, ...)
- Associação (situa-se)
- Multiplicidade da Associação
  - Um - um
  - Um - muitos
  - Muitos - muitos
- Chaves primárias (Produtor ID, Região ID)

# Como converter o modelo Entidade-Associação para tabelas em 5 passos

---

## **STEP 1 entidades fortes**

- Criar uma tabela com um nome de cada entidade
- Criar uma coluna com o nome de cada atributo e com o tipo de dados apropriado
- Definir uma chave primária

## **STEP 2 entidades fracas**

- Igual ao STEP 1, mais:
  - Criar uma coluna com a chave primária da entidade mãe (chave estrangeira)
  - Definir uma chave primária que inclua a chave primária da entidade mãe

# EXEMPLO

## Customers

PK	custid	cfirstname	clastname	cphone	cstreet	zipcode
	1234	Tom	Jewett	714-555-1212	10200 Slater	92708
	5678	Alvaro	Monge	562-333-4141	2145 Main	90840
	9012	Wayne	Dick	562-777-3030	1250 Bellflower	90840

## Orders

FK	custid	orderdate	soldby
	5678	2003-07-14	Patrick
	9012	2003-07-14	Patrick
	5678	2003-07-18	Kathleen
	5678	2003-07-20	Kathleen

### Customers

custID	cFirstName	cLastName	cPhone	cStreet	cZipCode
PK	Candidate Key (1 of 2)				

1..1 (parent)

Orders 0..\* (child)

FK	custID	orderDate	soldBy
	Primary Key		

# Como converter o modelo Entidade-Associação para tabelas em 5 passos (2)

---

## **STEP 3 associações 1 - 1**

- **Considerar fazer a fusão das duas identidades**
- **Se as entidades NÃO são redundantes:**
  - **Criar uma tabela para cada identidade**

**Em uma das tabelas criar uma coluna com a chave primária da outra tabela**

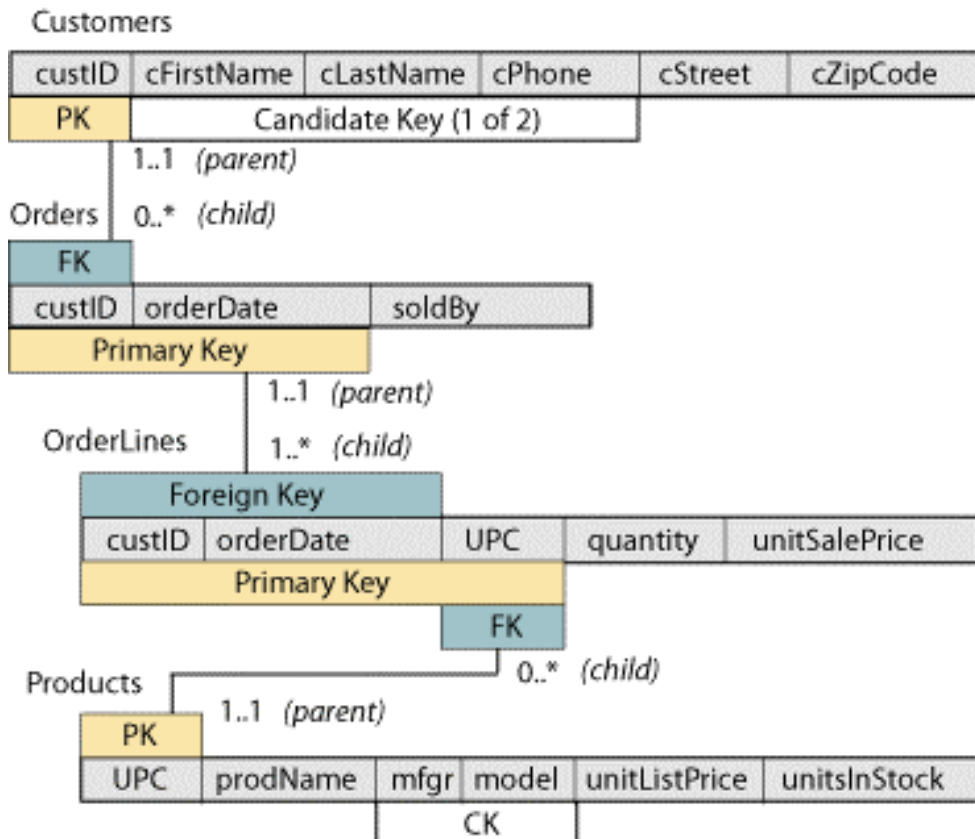
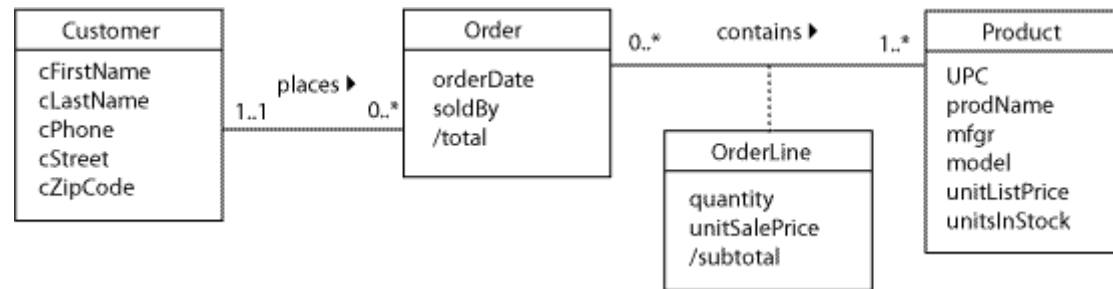
## **STEP 4 associações 1 - muitos**

- **Criar tabelas para cada identidade**
- **Na tabela do lado "muitos" criar uma coluna com a chave primária da tabela do lado "um" (chave estrangeira)**

## **STEP 5 associações muitos - muitos**

- **Criar uma tabela com o nome composto pelas duas entidades associadas**
- **Criar duas colunas com os nomes das chaves primárias das duas entidades (2 chaves estrangeiras)**

# EXEMPLO associação muitos-muitos



# Normalização

---

- uma tabela de CDs ...

Banda	Titulo CD	Editora	Musicas
Stevie Wonder	Talking Book	Motown	You are the sunshine of my life Maybe your baby Superstition ...
Miles Davis	Miles Smiles	Columbia	Orbits Circle ...
Wayne Shorter	Speak no evil	Bluenote	Witch hunt Fee-Fi-fo-fum ...
Herbie hancock	headhunters	Columbia	Chameleon Watermelon Maiden voyage...
herbie Hancock	Maiden voyage	Blue note	Maiden voyage



# Primeira forma normal 1NF

---

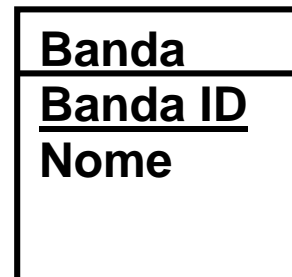
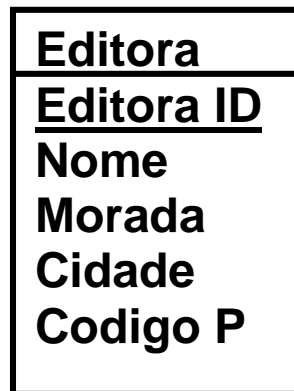
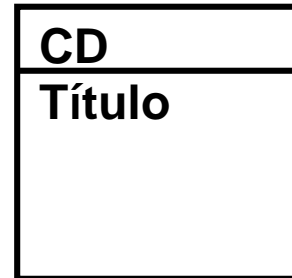
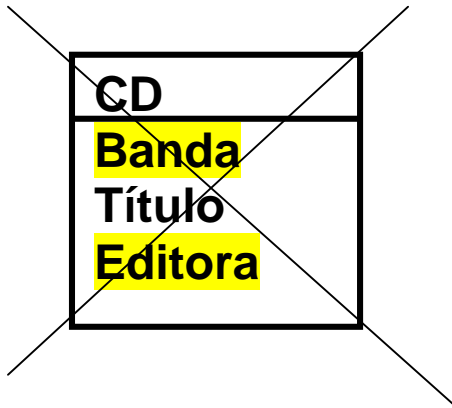
- Todos os atributos tem que ter apenas um valor (single-valued)



# Segunda forma normal 2NF

---

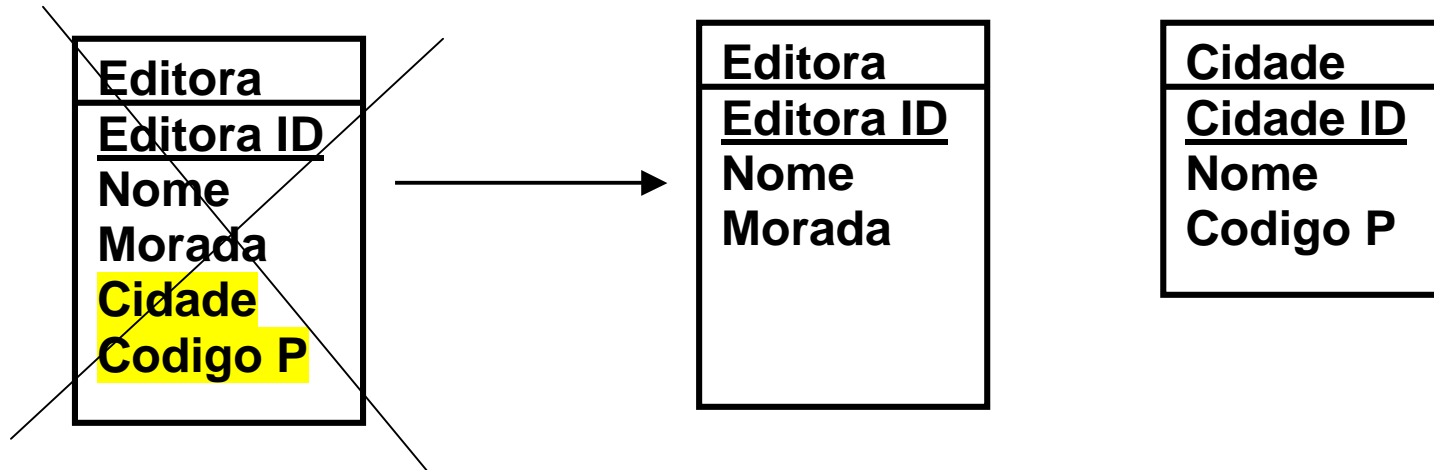
- 1NF mais:
- todos os atributos dependem apenas da chave primária



# Terceira forma normal 3NF

---

- 2NF mais:
- todos os atributos são independentes entre si (não há sub-chaves na tabela)



# Atributos repetidos

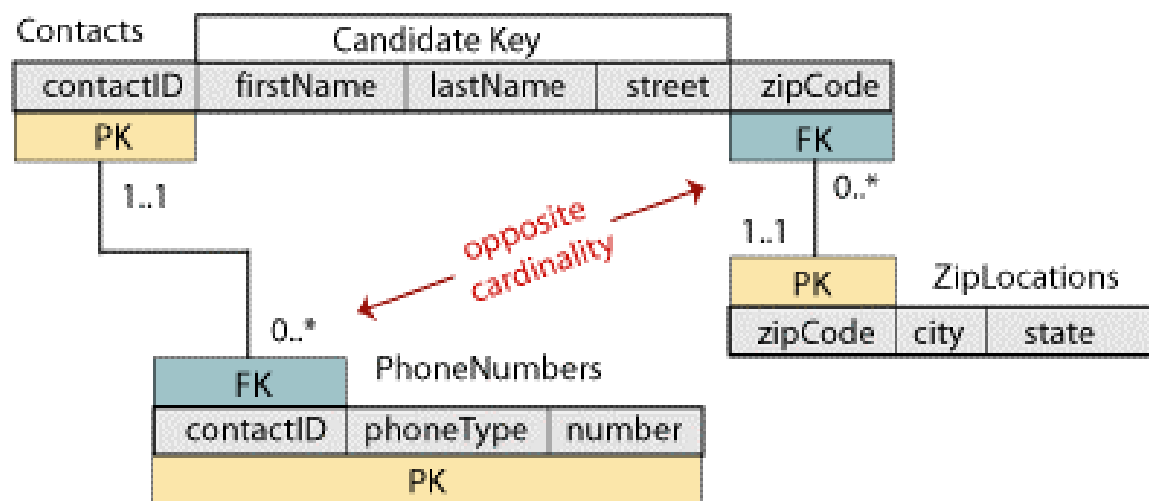
---

Contact
firstName
lastName
street
homePhone
workPhone
cellPhone
fax
pager

## Users

firstName	lastName	homePhone	workPhone	cellPhone	fax	pager
George	Barnes	562-874-1234		310-999-3628		
Susan	Noble	562-975-3388	714-847-3366			
Erwin	Star				714-997-5885	714-997-2428
Alice	Buck		562-577-1200	562-561-1921		
Frank	Borders	714-968-8201				
Hanna	Diedrich			562-786-7727		

# Atributos repetidos - solução



**Contacts**

contactid	firstname	lastname	street	zipcode
1639	George	Barnes	1254 Bellflower	90840
5629	Susan	Noble	1515 Palo Verde	90840
3388	Erwin	Star	17022 Brookhurst	92708
5772	Alice	Buck	3884 Atherton	90836
1911	Frank	Borders	10200 Slater	92708
4848	Hanna	Diedrich	1699 Studebaker	90840

**Phone numbers**

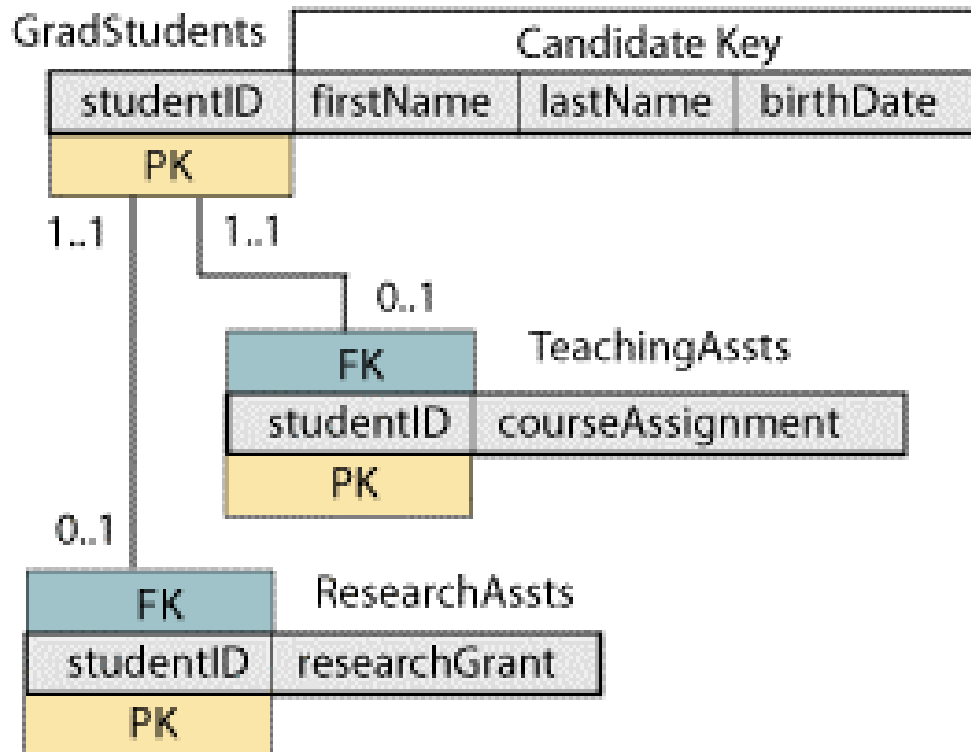
contactid	phonetype	number
1639	Home	562-874-1234
1639	Cell	310-999-3628
5629	Home	562-975-3388
5629	Work	714-847-3366
3388	Fax	714-997-5885
3388	Pager	714-997-2428
5772	Work	562-577-1200
5772	Cell	562-561-1921
1911	Home	714-968-8201
4848	Cell	562-786-7727

# Subclasses

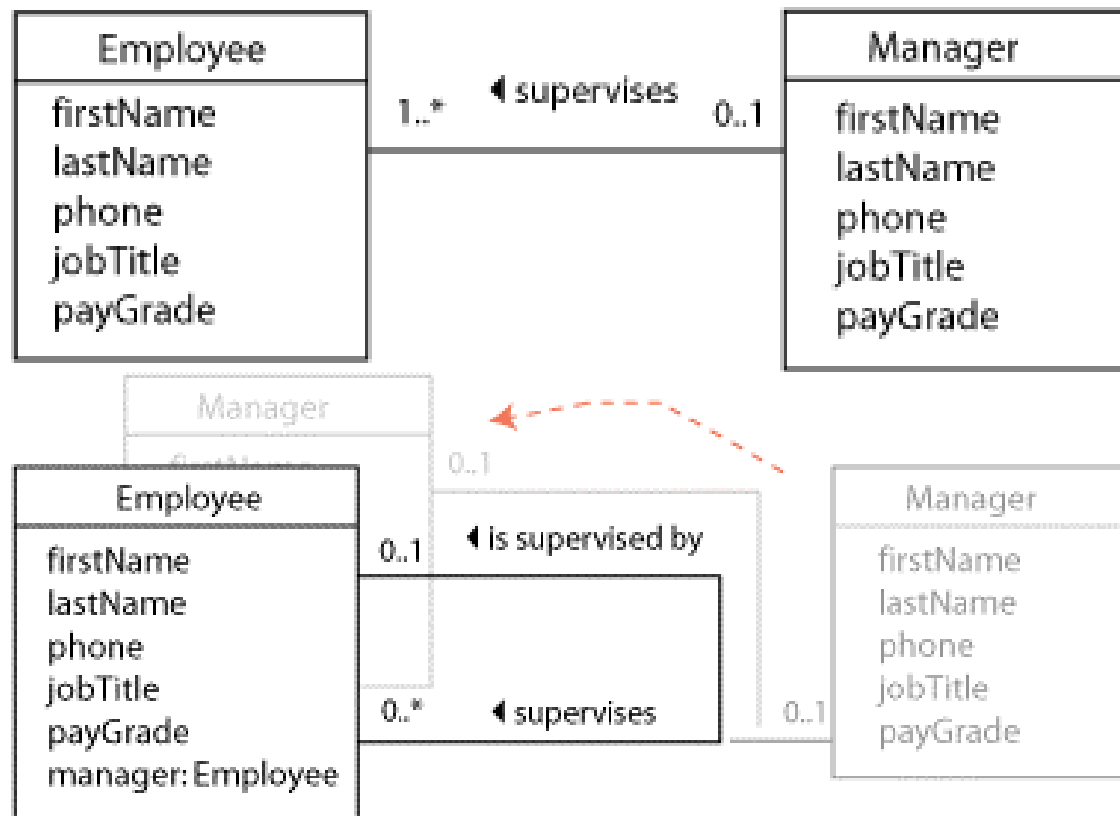
GradStudent
firstName
lastName
birthDate
courseAssignment
researchGrant

←----- *only applies to TAs*

←----- *only applies to RAs*



# Associação recursiva



Employees		Candidate Key					
employeeID	firstName	lastName	phone	jobTitle	payGrade	managerID	
PK						FK	
0..1						0..*	

# SQL Basics

---

- **CREATE TABLE**

```
CREATE TABLE table_name (column_name_1 type modifiers,  
                           column_name_2 type modifiers,...  
) type=table_type ;
```

- **Exemplo**

```
CREATE TABLE customer (  
    cust_pk int(5) NOT NULL PRIMARY KEY AUTO_INCREMENT,  
    lastname varchar(50),  
    firstname varchar(50),  
    phone char(15) DEFAULT '999999999',  
    birthday date,  
    status_fk int(2),  
    FOREIGN KEY (status_fk) REFERENCES status(status_pk)  
) type= InnoDB [MyISAM] ;
```

- **Outros comandos**

```
SHOW CREATE TABLE customer;  
ALTER TABLE customer MODIFY firstname TEXT(15);  
DROP TABLE customer;
```



# SQL Basics

---

- TIPOS DE DADOS (DATA TYPES)

	Espaço ocupado por uma string de 30 caracteres	Maximum size
INT		±2147483647
TINYINT		±127
REAL		
CHAR(length)	length	255
VARCHAR(length)	31	255
TINYTEXT(length)	31	255
TEXT(length)	32	65535
MEDIUMTEXT(length)	33	16777215
LONGTEXT(length)	34	4294967295
DATE		YYYY-MM-DD
TIME		HH:MM:SS
DATETIME		YYYY-MM-DD HH:MM:SS
TIMESTAMP		YYYYMMDDHHMMSS

# SQL Basics

---

- **INSERT DADOS**

```
INSERT INTO table_name (coluna_1, coluna_2, ..., coluna_N)
    VALUES (valor_1, valor_2, ..., valor_N);
```

```
INSERT INTO table_name SET coluna_1=valor_1, coluna_2= valor_2, ...,
coluna_N= valor_N ;
```

- **Exemplos**

```
INSERT INTO customer(cust_id, lastname, city)
    VALUES (1, 'Williams', 'New York');
```

```
INSERT INTO customer SET cust_id=1, lastname='Williams', city='New
York';
```

- **Outros comandos**

```
UPDATE FROM customer SET lastname='Johnson' WHERE cust_id=1;
DELETE FROM customer WHERE lastname='Williams';
```

# SQL Queries

---

```
SELECT coluna_1, coluna_2, ..., coluna_N  
FROM tabela_1, tabela_2, ..., tabela_M  
[WHERE clausula]  
[ORDER BY coluna];
```

- **Exemplos**

```
SELECT * FROM region;  
SELECT * FROM region WHERE region_id <=3;  
SELECT CURTIME();  
SELECT * FROM customer WHERE lname='Marzall' AND fname='Dimitri';  
SELECT * FROM customer WHERE (lname='Marzall' AND fname LIKE 'M%') OR  
birthday='1980-07-14';  
SELECT city, COUNT(*) FROM customer GROUP BY city;
```

# SQL Queries (cont)

---

```
SELECT *  
FROM customers;
```

cfirstname	clastname	cphone	cstreet	czipcode
Tom	Jewett	714-555-1212	10200 Slater	92708
Alvaro	Monge	562-333-4141	2145 Main	90840
Wayne	Dick	562-777-3030	1250 Bellflower	90840

```
SELECT *  
FROM customers  
WHERE cZipCode = '90840';
```

cfirstname	clastname	cphone	cstreet	czipcode
Alvaro	Monge	562-333-4141	2145 Main	90840
Wayne	Dick	562-777-3030	1250 Bellflower	90840

# SQL Queries (cont)

---

```
SELECT cLastName, cFirstName, cPhone
FROM customers
WHERE cZipCode = '90840';
```

Columns from SELECT

cLastName	cFirstName	cPhone
Monge	Alvaro	562-333-4141
Dick	Wayne	562-777-3030

```
SELECT cLastName, cFirstName, cPhone
FROM customers
WHERE cZipCode = '90840'
ORDER BY cLastName, cFirstName;
```

Rows in order

cLastName	cFirstName	cPhone
Dick	Wayne	562-777-3030
Monge	Alvaro	562-333-4141

# SQL Join

---

**Customers**

cfirstname	clastname	cphone	cstreet	czipcode
Tom	Jewett	714-555-1212	10200 Slater	92708
Alvaro	Monge	562-333-4141	2145 Main	90840
Wayne	Dick	562-777-3030	1250 Bellflower	90840

**Orders**

cfirstname	clastname	cphone	orderdate	soldby
Alvaro	Monge	562-333-4141	2003-07-14	Patrick
Wayne	Dick	562-777-3030	2003-07-14	Patrick
Alvaro	Monge	562-333-4141	2003-07-18	Kathleen
Alvaro	Monge	562-333-4141	2003-07-20	Kathleen

# SQL INNER JOIN (cont)

---

```
SELECT * FROM customers, orders
WHERE customers.cphone=orders.cphone
```

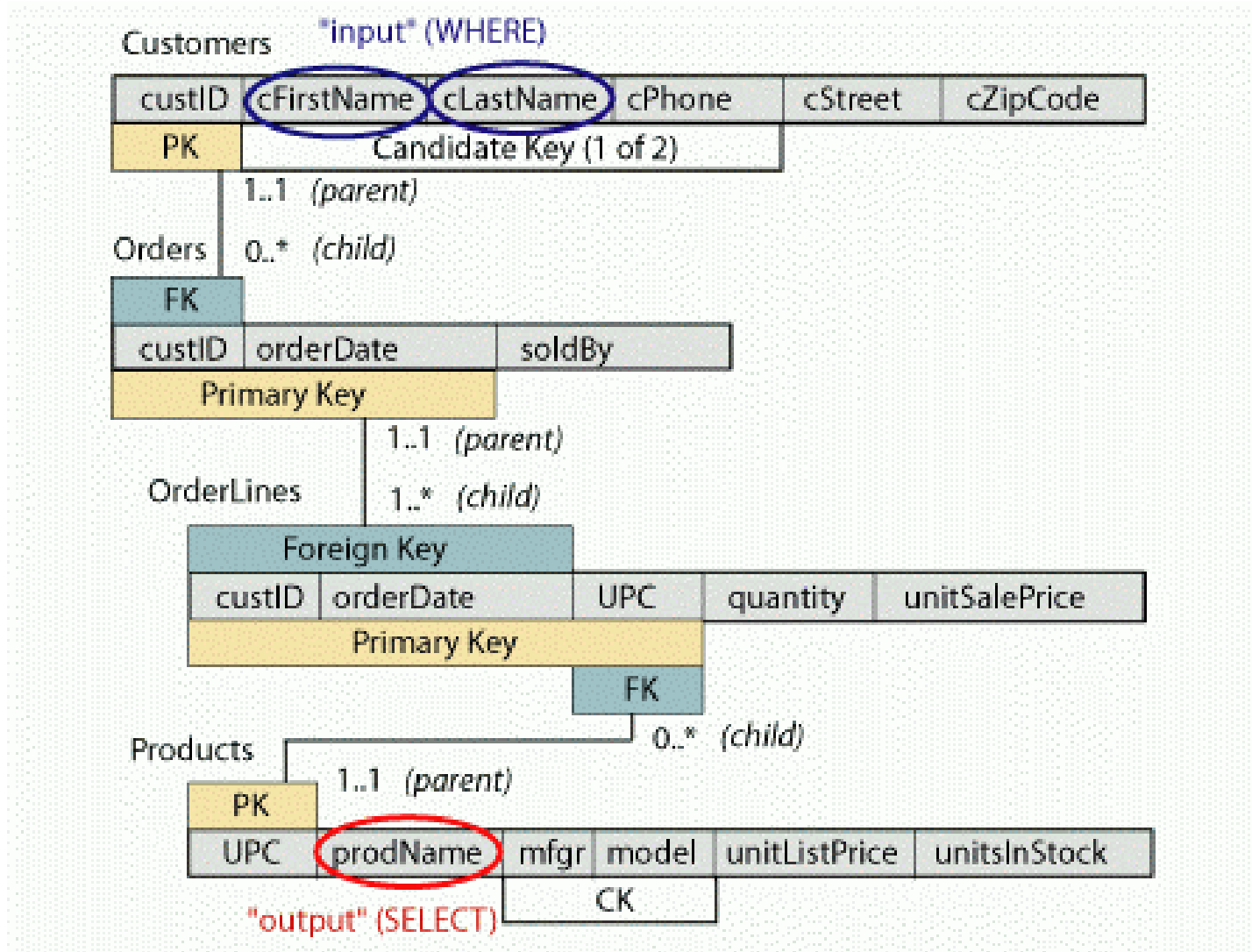
```
SELECT * FROM customers INNER JOIN orders
ON customers.cphone=orders.cphone
```

**Customers joined to Orders**

cfirstname	clastname	cphone	cstreet	czipcode	orderdate	soldby
Alvaro	Monge	562-333-4141	2145 Main	90840	2003-07-14	Patrick
Wayne	Dick	562-777-3030	1250 Bellflower	90840	2003-07-14	Patrick
Alvaro	Monge	562-333-4141	2145 Main	90840	2003-07-18	Kathleen
Alvaro	Monge	562-333-4141	2145 Main	90840	2003-07-20	Kathleen

# SQL INNER JOIN (cont)

Task: encontrar todos os produtos que foram comprados pelo cliente Alvaro Monge





# SQL INNER JOIN (cont)

---

**Solução: ligar todas as tabelas através das chaves primárias e estrangeiras**

```
SELECT c.cFirstName, c.cLastName, p.prodName
FROM customers AS c INNER JOIN (orders AS o, orderlines AS ol,
products AS p)
ON (c.custID=o.custID AND o.custID=ol.custID AND
o.orderDate=ol.orderDate AND ol.UPC=p.UPC)
WHERE c.cFirstName = 'Alvaro' AND c.cLastName = 'Monge';
```

cFirstName	cLastName	prodName
Alvaro	Monge	Hammer, framing, 20 oz.
Alvaro	Monge	Hammer, framing, 20 oz.
Alvaro	Monge	Screwdriver, Phillips #2, 6 inch
Alvaro	Monge	Screwdriver, Phillips #2, 6 inch
Alvaro	Monge	Pliers, needle-nose, 4 inch

# SQL OUTER JOIN

```
SELECT cfirstname, clastname, orderdate
FROM customers AS c LEFT OUTER JOIN orders AS o
ON (c.firstname = o.firstname AND c.lastname = o.lastname) ;
```

**Customers**

cfirstname	clastname	cphone	cstreet	czipcode
Tom	Jewett	714-555-1212	10200 Slater	92708
Alvaro	Monge	562-333-4141	2145 Main	90840
Wayne	Dick	562-777-3030	1250 Bellflower	90840

**Orders**

cfirstname	clastname	cphone	orderdate	soldby
Alvaro	Monge	562-333-4141	2003-07-14	Patrick
Wayne	Dick	562-777-3030	2003-07-14	Patrick
Alvaro	Monge	562-333-4141	2003-07-18	Kathleen
Alvaro	Monge	562-333-4141	2003-07-20	Kathleen

**All customers and order dates**

cfirstname	clastname	orderdate
Tom	Jewett	
Alvaro	Monge	2003-07-14
Alvaro	Monge	2003-07-18
Alvaro	Monge	2003-07-20
Wayne	Dick	2003-07-14

# SQL FUNÇÕES

---

```
SELECT p.prodname AS "product name",  
       COUNT(p.prodname) AS "times ordered"  
FROM products AS p INNER JOIN orderlines AS ol  
ON p.UPC=ol.UPC  
GROUP BY p.prodname  
HAVING COUNT(p.prodname) > 0;
```

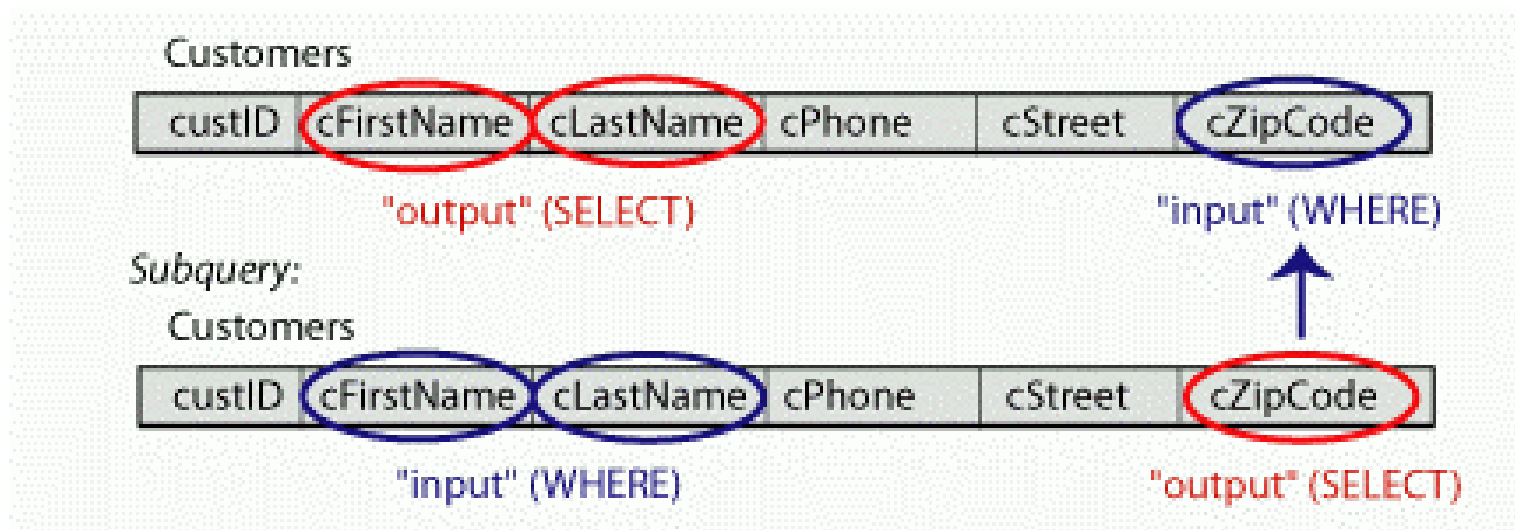
**Product orders**

product name	times ordered
Hammer, framing, 20 oz.	3
Pliers, needle-nose, 4 inch	1
Saw, crosscut, 10 tpi	1
Screwdriver, Phillips #2, 6 inch	2

# Nested queries (subqueries)

- Task: encontrar todos os clientes que moram na mesma cidade que o Wayne Dick
- Problema: não sabemos o código zip do Wayne...

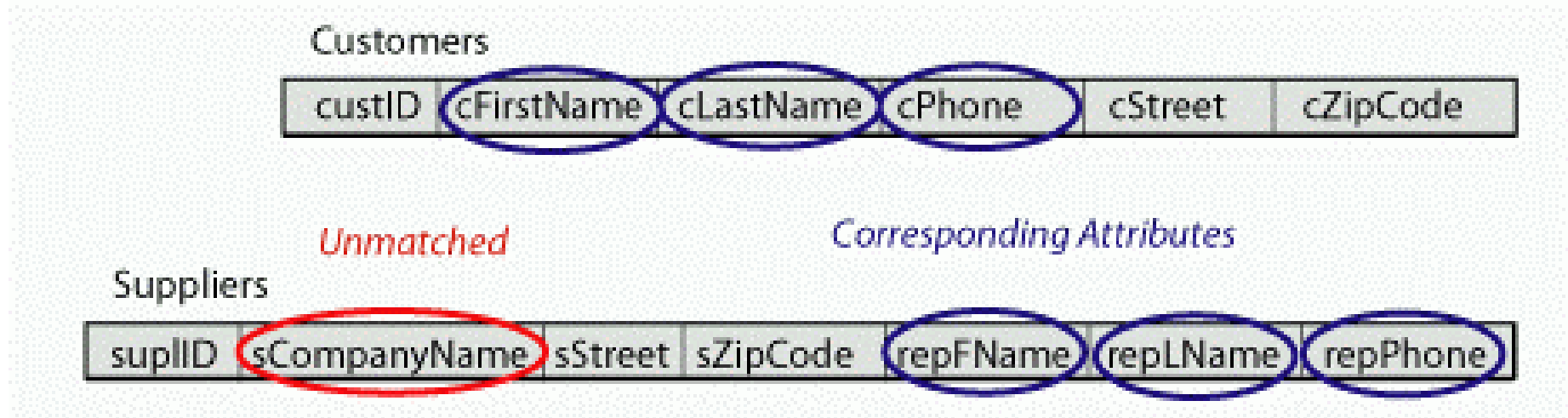
```
SELECT cFirstName, cLastName, cZipCode
FROM customers
WHERE cZipCode =
    (SELECT cZipCode
     FROM customers
     WHERE cFirstName = 'Wayne' AND cLastName = 'Dick');
```



# UNIONS

- JOIN junta duas entidades (tabelas) com atributos (colunas) distintos
- UNION apenas trabalha com tabelas com colunas (atributos) idênticas (do mesmo tipo)

**EXEMPLO:** produzir uma lista de todas as pessoas (clientes e fornecedores) com que a empresa trabalha



# UNIONS (cont)

---

```
SELECT cLastName AS "Last Name", cFirstName AS "First Name",
       cPhone as "Phone", 'Customer' AS "Company"
FROM customers
UNION
SELECT repLName, repFName, repPhone, sCompanyName
FROM suppliers
ORDER BY "Last Name";
```

**Phone list**

Last Name	First Name	Phone	Company
Bradley	Jerry	888-736-8000	Industrial Tool Supply
Dick	Wayne	562-777-3030	Customer
Jewett	Tom	714-555-1212	Customer
Monge	Alvaro	562-333-4141	Customer
O'Brien	Tom	949-567-2312	Bosch Machine Tools

# MINUS

---

**EXEMPLO: encontrar clientes que não fizeram compras em 2002**

```
SELECT cLastName, cFirstName, cStreet, cZipCode
FROM Customers
WHERE custID NOT IN
    (SELECT custID
     FROM Orders
     WHERE TO_CHAR(orderDate, 'YYYY') = '2002');
```