

# **Data Base Relazionali**

**Ing. Maria Grazia Celentano**  
**[www.mariagraziacelentano.it](http://www.mariagraziacelentano.it)**

# Introduzione

- La raccolta, l'organizzazione e la conservazione dei dati sono sempre stati i principali compiti dei **sistemi informatici**
  - Dati degli utenti di una banca
  - Prenotazione dei voli aerei di una compagnia
  - Prenotazione di un albergo
- **I sistemi informatici devono garantire:**
  - Memorizzazione (di grandi quantità di dati)
  - Aggiornamento
  - Accesso (a molteplici utenti contemporaneamente)

# Sistemi informativi e informatici

- Nello svolgimento di un'attività è essenziale la disponibilità di informazioni e la loro gestione efficace
- Un sistema informativo organizza e gestisce le informazioni necessarie per perseguire gli scopi dell'informazione stessa
  - NOTA: non è necessariamente automatizzato.

Es. Le banche hanno archivi da molto più tempo dell'esistenza dei computer!

# Sistemi informativi e informatici

- Un sistema informatico è la porzione automatizzata di un sistema informativo
  - La diffusione dell'informatica fa sì che spesso i **sistemi informativi** siano anche **sistemi informatici**
  - E' necessario strutturare e organizzare la conoscenza per poi poterla rappresentare

# Dati e informazioni

- Le **informazioni** sono elementi che consentono di arricchire la nostra conoscenza del mondo e spesso devono essere organizzate e rappresentate
  - Es. lingua scritta, numeri, disegni ...
- Nei sistemi informatici le informazioni vengono rappresentate per mezzo di **dati**
  - “... i dati da soli non hanno alcun significato, ma una volta interpretati e correlati opportunamente essi forniscono informazioni ...”

# Le Basi di Dati

- **BASE DI DATI** (*database*): collezione di **dati correlati** utilizzati per rappresentare le informazioni di interesse di un sistema informativo.

Ogni giorno le nostre azioni implicano l'accesso a una base di dati da parte di qualcuno

*(prelievo/versamento bancario, acquisto biglietto aereo, prenotazione alberghiera, acquisto nel supermercato ...)*

- Una **base di dati** può essere di qualsiasi dimensione e complessità; può essere manuale (es. lo schedario di una biblioteca) o mantenuta e gestita in maniera computerizzata utilizzando appositi applicativi

# Proprietà delle Basi di Dati

- Rappresentare un aspetto del mondo reale (Mini-mondo o Universo del Discorso), ogni cambiamento nel mini-mondo determina un cambiamento nella base dati
- Essere una collezione di dati logicamente coerenti e con un certo significato intrinseco
- Essere sempre progettate, costruite o popolate per uno scopo specifico, quindi per particolari tipi di utenti

# Sistema di Gestione di Basi di Dati

Un **DBMS** (*DataBase Management System*) è un insieme di programmi che permettono agli utenti di creare e mantenere una base di dati.



- in grado di gestire collezioni di dati che siano **Grandi, Condivise, Persistenti**
- assicurando **Affidabilità, Privatezza**
- in modo **Efficace** ed **Efficiente**

# Caratteristiche dei DBMS

- **Grandi**

- I **DBMS** devono essere in grado di gestire ingenti quantità di dati memorizzati anche in memoria secondaria

- **Condivise**

- I **dati** devono poter essere usati da applicazioni e utenti diversi secondo le proprie modalità

- **Persistenti**

- I **dati** durano nel tempo, oltre le singole applicazioni

# Caratteristiche dei DBMS

- **Affidabilità**
  - **DBMS** devono conservare i dati anche in caso di malfunzionamento HW e SW (backup e recovery)
- **Privatezza**
  - I **DBMS** devono consentire ad ogni utente solo le azioni di sua competenza (meccanismi di autorizzazione)
- **Efficienza**
  - I **DBMS** devono operare in modo da richiedere risorse (tempo e spazio) accettabili per gli utenti
- **Efficacia**
  - I **DBMS** devono rendere produttive le attività degli utenti (fornendo i servizi di cui necessitano)

# Caratteristiche dei DBMS

Il DBMS permette di gestire i processi di:

- **DEFINIZIONE**

- definire la struttura delle DB (specificare i tipi di dati, le loro strutture e i vincoli per i dati che devono essere memorizzati)

- **COSTRUZIONE**

- immagazzinare i dati entro un certo mezzo di memorizzazione che è controllato dal DBMS

- **MANIPOLAZIONE**

- interrogare il DB, analizzare i dati, visualizzare e stampare i dati specifici, aggiornare gli stessi per rispecchiare i cambiamenti del mini-mondo

# Gestione dei Dati

- **SISTEMA TRADIZIONALE**

- L'approccio convenzionale sfrutta i **file** (archivi) per memorizzare i dati su memorie di massa.
- I file consentono di memorizzare in modo semplice, ma non hanno meccanismi adeguati per l'accesso e la condivisione dei dati.
  - Archivio anagrafico in un file di testo.
  - Problemi: modifiche, ricerche,.....
- **Ciascun utente definisce ed implementa i file necessari per una specifica applicazione**
  - Ufficio contabilità e l'ufficio segreteria di una Università

# Gestione dei Dati

- **SISTEMA CON BASI DI DATI**

- **Natura autodescrittiva**

- Ciascun sistema di basi di dati contiene al suo interno una descrizione completa della sua struttura e dei suoi vincoli (Metadati). Tale catalogo fornisce informazioni sulla struttura della base dati.

- **Indipendenza tra dati e programmi**

- La BD ha vita indipendente dal programma applicativo. Essendo la struttura della Base Dati descritta in un catalogo, l'inserimento di un nuovo campo non richiede la modifica del programma che lo sta utilizzando.

- **Indipendenza tra programmi e applicazioni**

- I programmi applicativi dell'utente operano sui dati invocando delle funzioni (attraverso il loro nome e argomenti). E' l'implementazione di tali funzioni che può cambiare.

Astrazione dei dati

# Gestione dei Dati

- **SISTEMA CON BASI DI DATI**

- **Supporto alle viste multiple**

- Essendo la base dati accessibile a più utenti, è possibile creare diverse applicazioni che forniscono una vista parziale dello stesso BD

- **Condivisione dei dati**

- Un DBMS multi-utente deve consentire a più utenti di accedere contemporaneamente alla base di dati. Il DBMS deve garantire la coerenza di tali dati mediante:
  - **Gestione della concorrenza**: garantisce che gli aggiornamenti effettuati da più utenti avvengano in modo controllato
  - **Elaborazione delle transazioni**: assicurare che le transazioni concorrenti operino correttamente. (Es. prenotazione di un volo aereo)

# Modello dei dati

- Per gestire i dati tramite un sistema informatico è necessario organizzarli e descrivere la struttura

**Il MODELLO DEI DATI** è un *insieme di concetti utilizzati per organizzare i dati di interesse e descriverne la struttura in modo che essa sia comprensibile ad un elaboratore*

# Modello Relazionale

- Il **Modello Relazionale** dei dati si basa sul costrutto di **relazione** e consente di organizzare i dati per mezzo di record a struttura fissa

I dati sono organizzati in relazioni (tabelle)

- Righe → Record
- Colonne → Campi

**LIBRI**

<b>Titolo</b>	<b>Autore</b>
<b>I Promessi Sposi</b>	<b>A. Manzoni</b>
<b>La Divina Commedia</b>	<b>D. Alighieri</b>

# Modelli dei dati

- i modelli dei dati sono detti **Modelli Logici** per indicare che le strutture da loro usate riflettono una particolare organizzazione dei dati
  - I DBMS commerciali usano i meccanismi forniti dal modello per organizzare i dati
  - Es. DBMS relazionale rappresenta i dati usando le tabelle
- è fondamentale che il modello logico sia **Potente** (espressivo) e **Semplice** (intuitivo)
- esistono anche **Modelli Concettuali**
  - Usati per descrivere i concetti del modello reale, non i dati utili a rappresentarli
  - Sono usati in fase di analisi preliminare e poi “mappati” su un modello logico

# Modelli di Dati Concettuali

- **Modelli di dati di alto livello o *CONCETTUALI***
  - Fornisce una descrizione astratta del Minimondo
  - La tecnologia usata prevede schemi semiformali molto vicini al linguaggio naturale.
  - Si usano concetti come:

*Entità*

*Attributi*

*Associazioni*

# Modelli di Dati Concettuali

- **ENTITA'**

rappresenta un oggetto o concetto del mondo reale (un impiegato, un progetto,...) descritto nella BD

- **ATTRIBUTO**

rappresenta una proprietà di interesse che descrive più a fondo un'entità (nome, salario dell'impiegato,...)

- **ASSOCIAZIONE tra 2 o più entità**

rappresenta un'interazione tra le entità

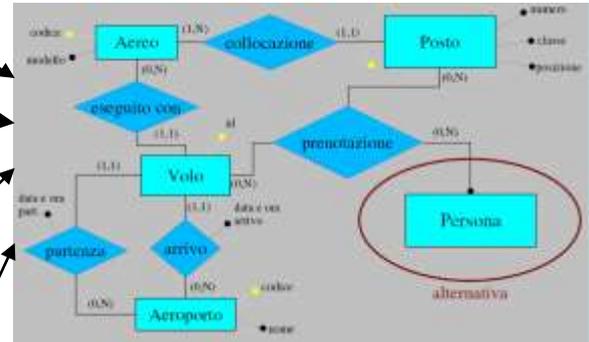
# Progettazione di una base di dati

- **Progettazione Concettuale**
  - Modello Entità-Relazione
- **Progettazione Logica**
  - Schema Relazionare
- **Progettazione Fisica**
  - Implementazione mediante un DBMS

# Modelli dei dati

## 1. Proget. Concettuale

- Fatture
- Ordini dei clienti
- Indirizzi dei clienti
- Prodotti in magazzino



Requisiti

Modello concettuale (COSA)

## 2. Proget. Logica

Entità	Descrizione	Attributi	Identificatori
Volo	Un singolo volo compiuto in una data e ora ben precisa.	Codice	Codice
Aeroporto	Un aeroporto	Codice, Nome	Codice
Aereo	Un singolo aeroplano	Codice, Nome, Data primo volo	Codice

Modello Logico (COME)

# Schemi, Istanze e Stato di un DB

- Le basi di dati sono costituite da:

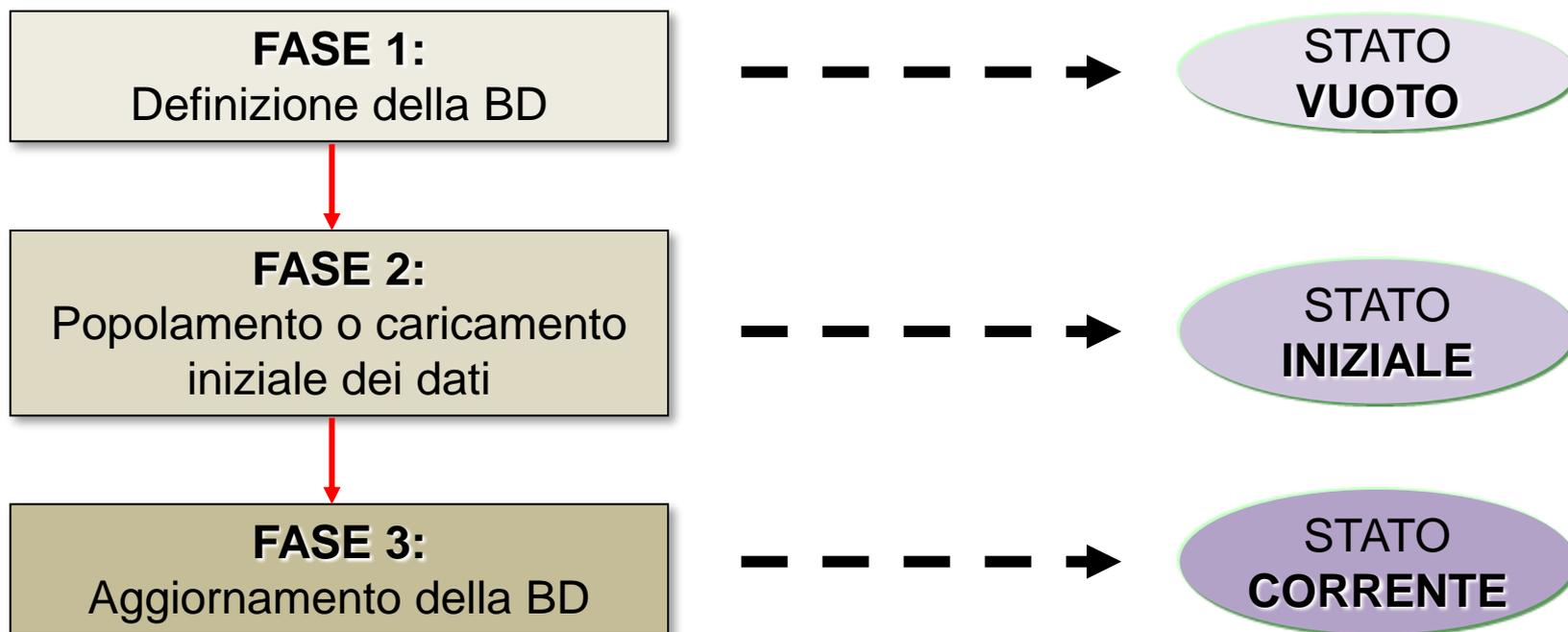
- **Schema** (Es. modello relazionale):
  - Parte invariante nel tempo
  - Caratteristiche dei dati
- **Istanza o stato**
  - Parte variabile nel tempo
  - Valore effettivi dei dati

- Es. DBMS relazionale
  - **Schema**: struttura delle tabelle
  - **Istanza o stato**: righe delle tabelle

## LIBRI

<b>Titolo</b>	<b>Autore</b>
<b>I Promessi Sposi</b>	<b>A. Manzoni</b>
<b>La Divina Commedia</b>	<b>D. Alighieri</b>

# Schemi, Istanze e Stato di un DB



# Esempi di DBMS

- **ACCESS**
  - DBMS relazionale semplice da usare
  - Si basa sul modello logico delle tabelle
  - Gestisce migliaia di record di dati organizzati in tabelle
- **Microsoft SQL server**
- **MySQL**
- **Oracle**

# Progettazione di una base di dati

- **Progettazione Concettuale**
  - Modello Entità-Relazione
- **Progettazione Logica**
  - Schema Relazionare
- **Progettazione Fisica**
  - Implementazione in Access

# Il Modello Entità-Relazione

## **II MODELLO ENTITA'-RELAZIONE**

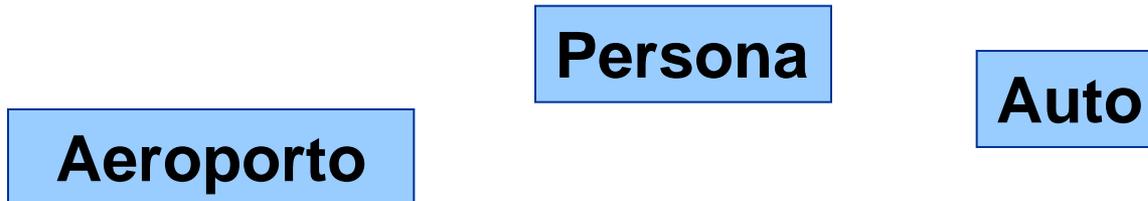
**è un modello concettuale utilizzato per descrivere la realtà di interesse.**

E' composto da costrutti che si combinano tra loro per formare degli schemi concettuali, i quali descrivono la struttura della realtà di interesse.

# Le ENTITA'

Ogni entità rappresenta una **classe di oggetti** (fatti, cose, persone, ecc.) che hanno delle proprietà comuni ed una esistenza “autonoma”.

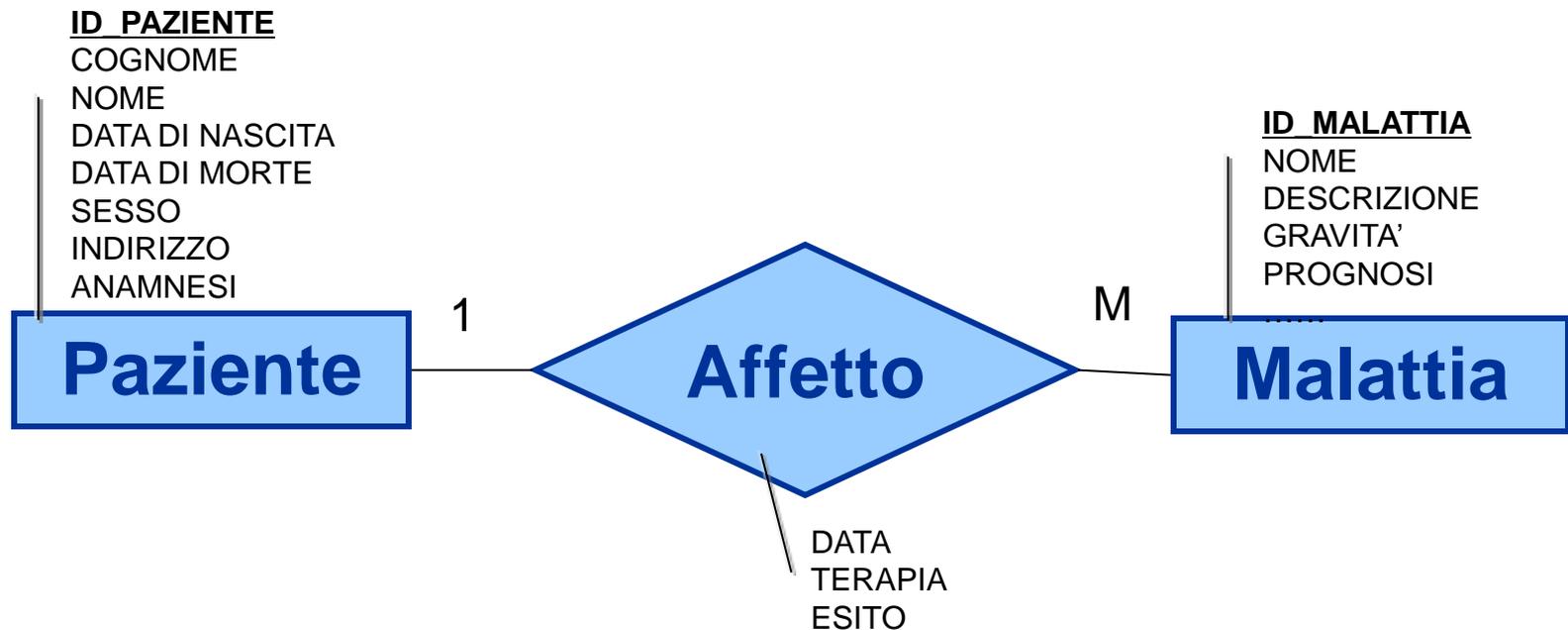
Le entità sono rappresentate da rettangoli che racchiudono il nome (al singolare) delle entità



- Una **ISTANZA** di una entità è un oggetto della classe che l'entità rappresenta.
- L'istanza non è un insieme di valori che identificano un oggetto, ma è proprio l'oggetto.

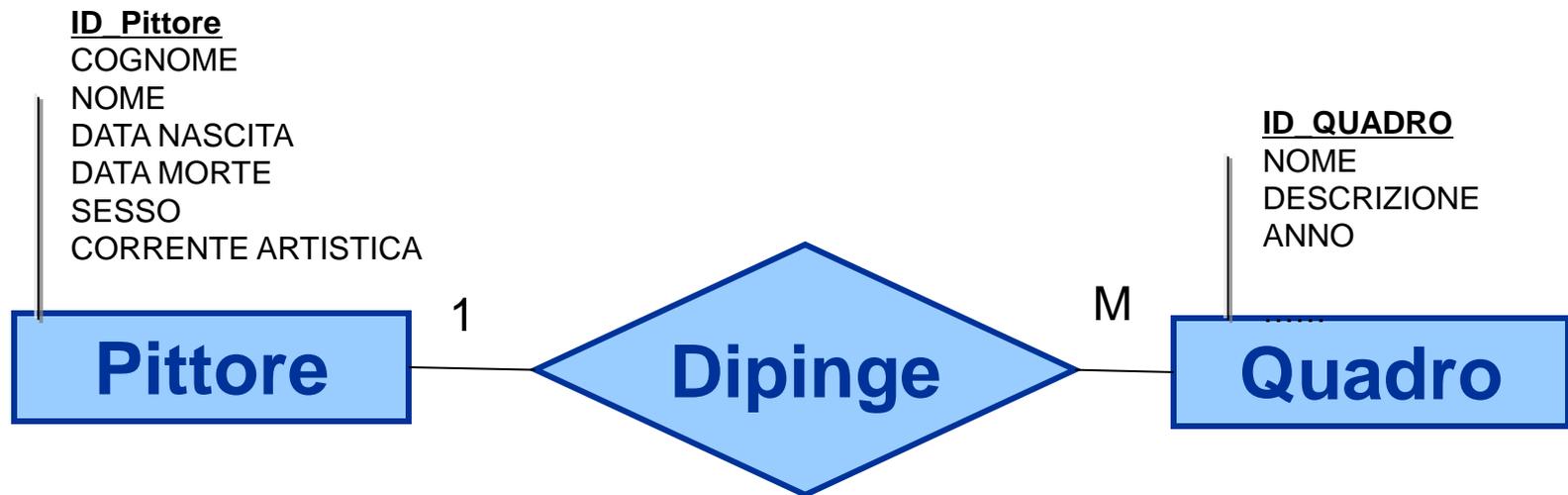
# Le RELAZIONI

Una **RELAZIONE** rappresenta un legame logico, significativo per l'applicazione, tra 2 o più entità.



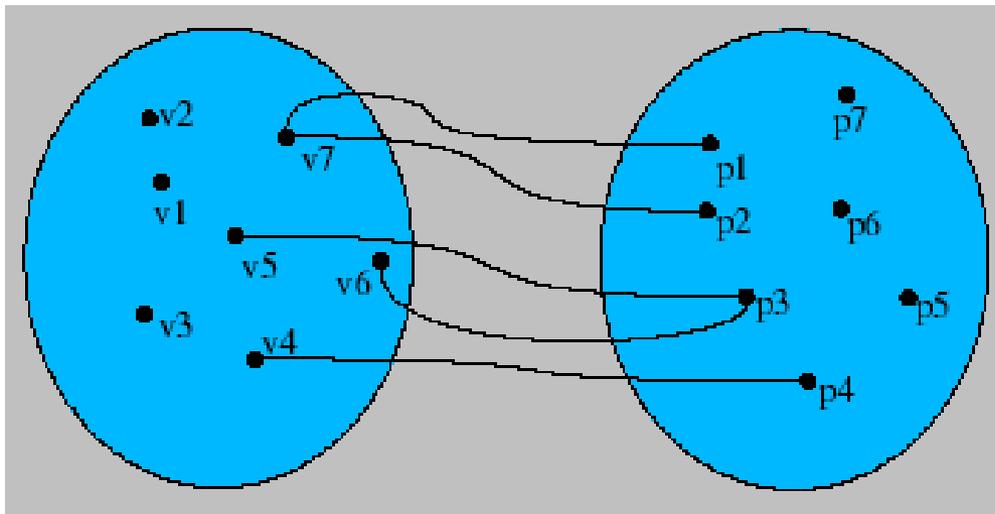
# Le RELAZIONI

Una **RELAZIONE** rappresenta un legame logico, significativo per l'applicazione, tra 2 o più entità.



# Le ISTANZE

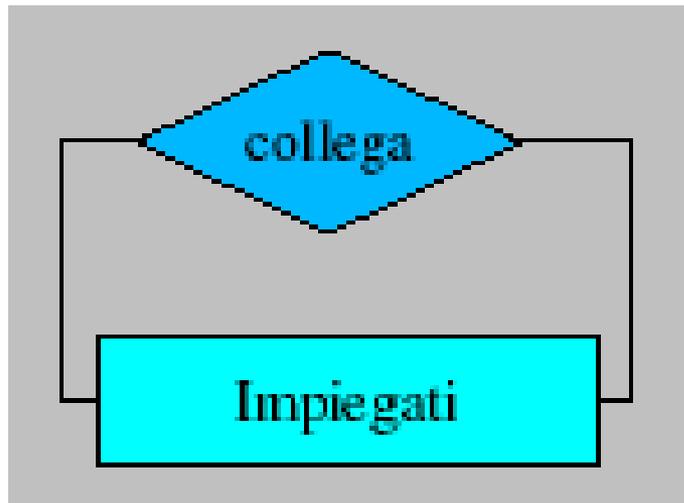
Una **ISTANZA** di una relazione è una  
Ennupla (coppia di relazioni binarie)  
costituita da occorrenze di entità, una per ogni entità coinvolta.



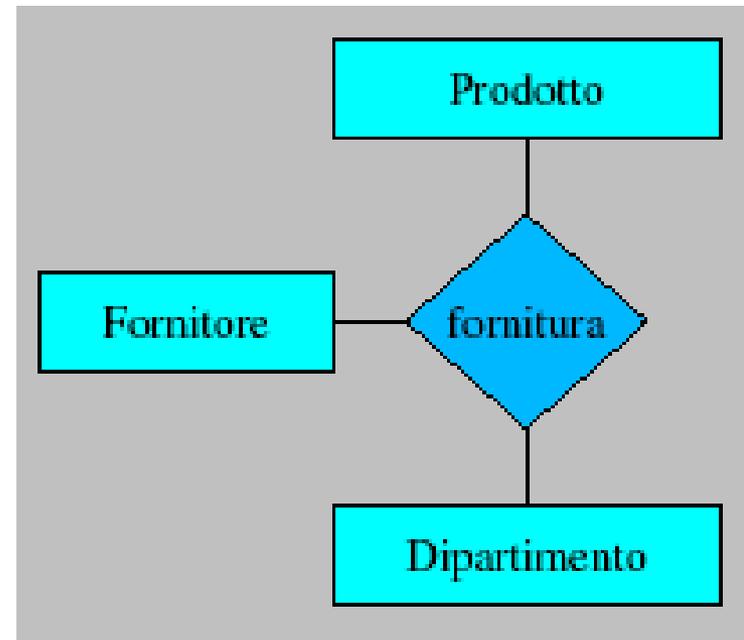
Sono istanze le coppie:  
(v7,p1)  
(v7,p2)  
(v4,p4)  
.....

# Le RELAZIONI

## Relazioni ricorsive

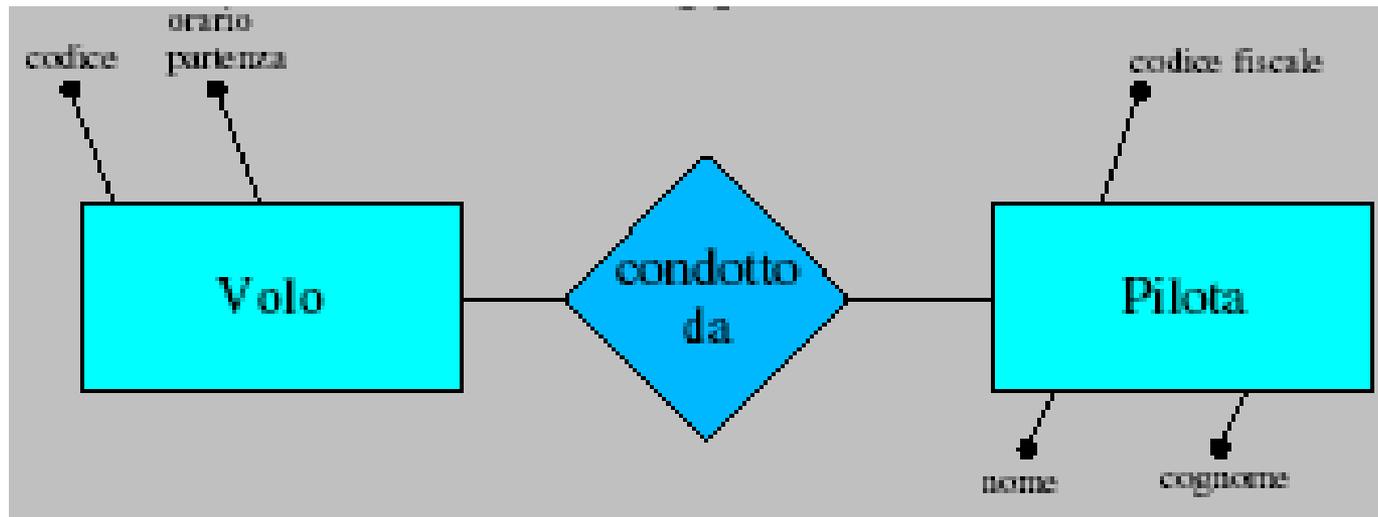


## Relazioni ternarie

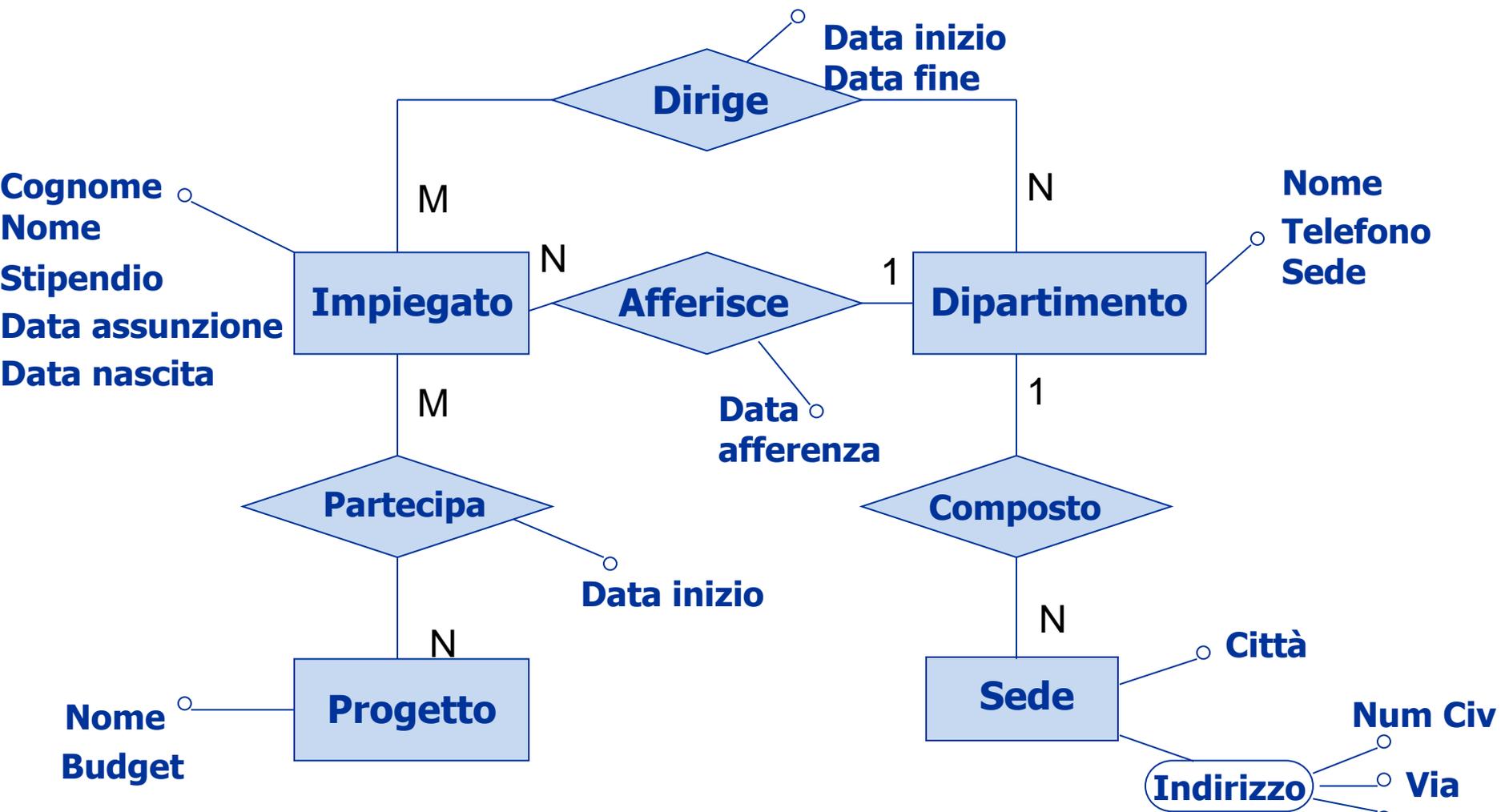


# Gli ATTRIBUTI

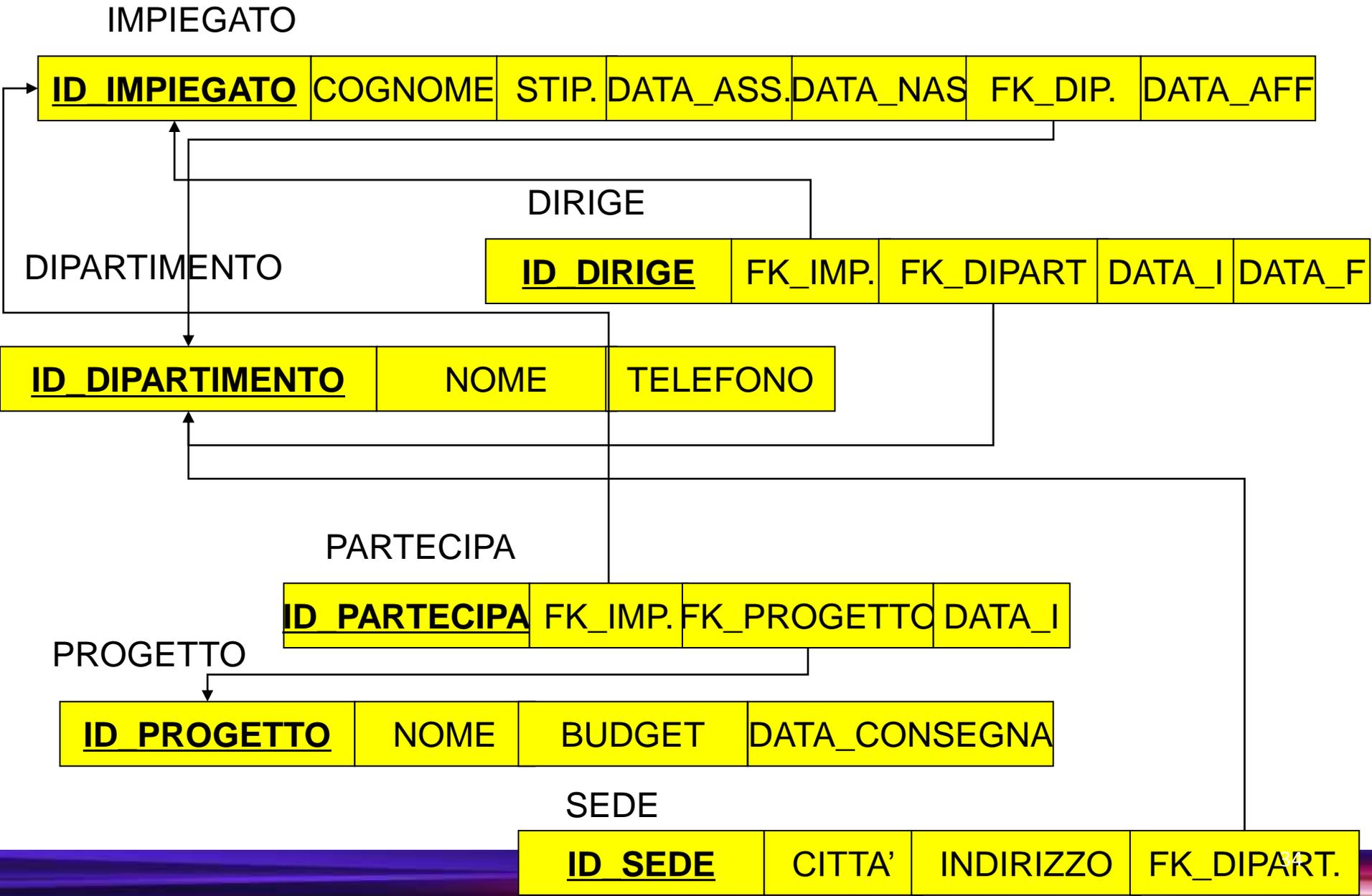
Descrivono le proprietà elementari di entità o relazioni che sono di interesse ai fini dell'applicazione.



# Modello Entità-Relazione



# Modello Relazionale



# Cardinalità delle relazioni

quante volte, in una relazione tra entità, una occorrenza di una di queste entità può essere legata a occorrenze delle altre entità coinvolte



Ad ogni impiegato possono essere assegnati da un minimo di 1 fino a un massimo di 5 incarichi.

Un incarico può essere assegnato fino a 50 impiegati

# Cardinalità delle relazioni

- Nella maggiore parte dei casi, è sufficiente utilizzare solo tre valori:
  - **Zero**
  - **Uno**
  - Il simbolo **M**: indica genericamente un intero maggiore di uno

# Cardinalità delle relazioni

- Esempio 1:



- Ogni persona può essere residente in una e una sola città
- Ogni città può non avere residenti oppure avere molti residenti
- Relazione **UNO A MOLTI**

# Cardinalità delle relazioni

- Esempio 2:



- Cardinalità massima pari a uno per entrambe le entità coinvolte: definisce una corrispondenza uno a uno tra le occorrenze di tali entità
- Relazione UNO A UNO

# Cardinalità delle relazioni

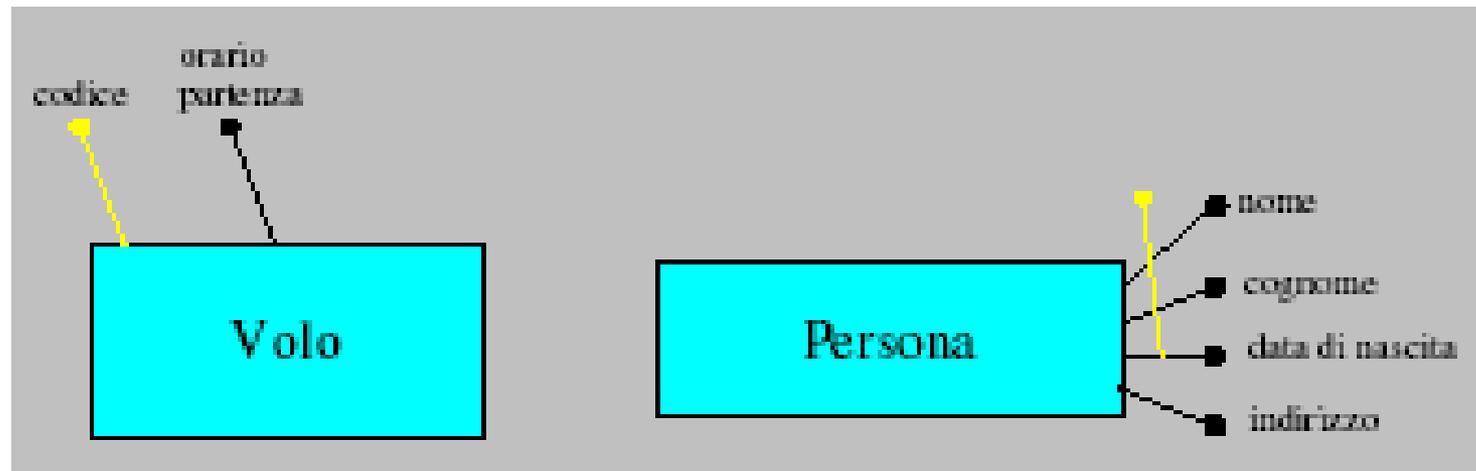
- Esempio 3:



- Cardinalità massima pari a N per entrambe le entità coinvolte
- Relazione **MOLTI A MOLTI**

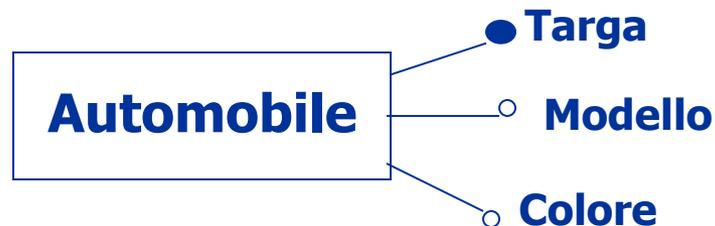
# Identificatori delle Entità: chiave primaria

- Descrivono i concetti (attributi e/o entità) che permettono di identificare univocamente le occorrenza delle entità
- In molti casi, uno o più attributi di una entità sono sufficienti a individuare un identificatore
  - Un identificatore *interno* (o *chiave*)



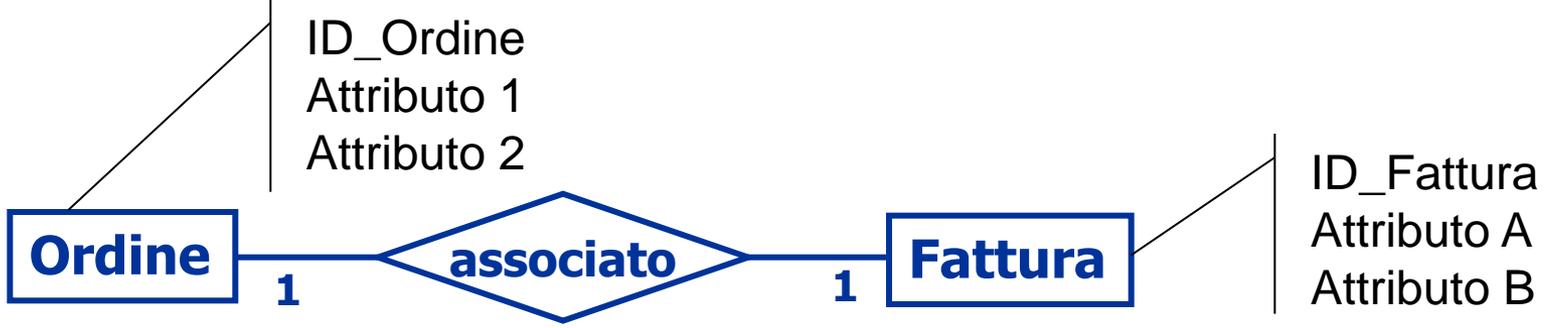
# Identificatori delle Entità

- Per esempio: non possono esistere due automobili con la stessa targa
- Targa può essere un identificatore interno per l'entità Automobile



# RELAZIONE 1:1

## MODELLO ENTITÀ-RELAZIONE



## MODELLO RELAZIONALE

Tab ORDINE

<b>ID_Ordine</b>	<b>Attributo 1</b>	<b>Attributo 2</b>	<b>FK_Fattura</b>
------------------	--------------------	--------------------	-------------------

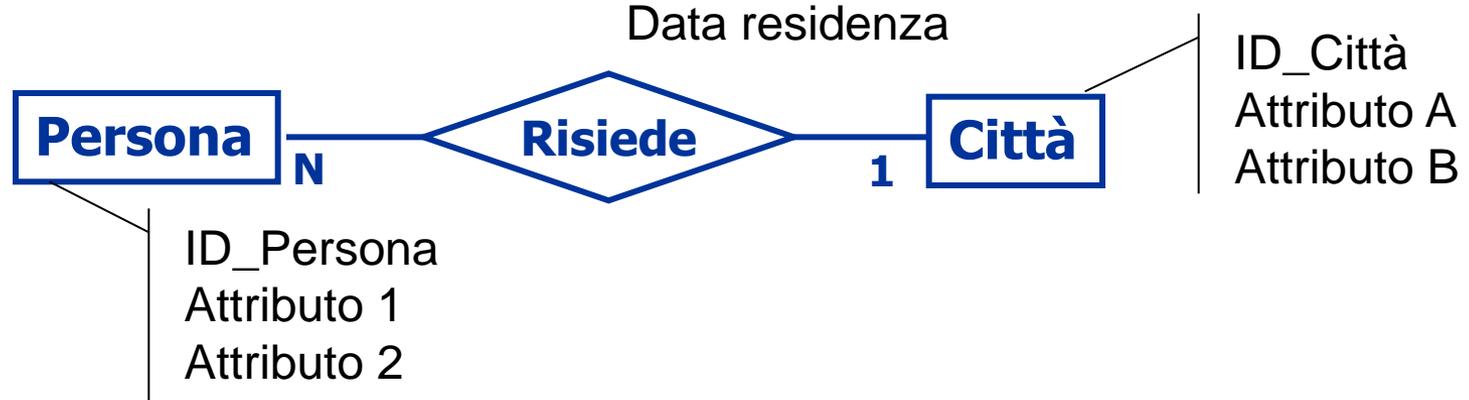
Tab FATTURA

<b>ID_Fattura</b>	<b>Attributo A</b>	<b>Attributo B</b>
-------------------	--------------------	--------------------

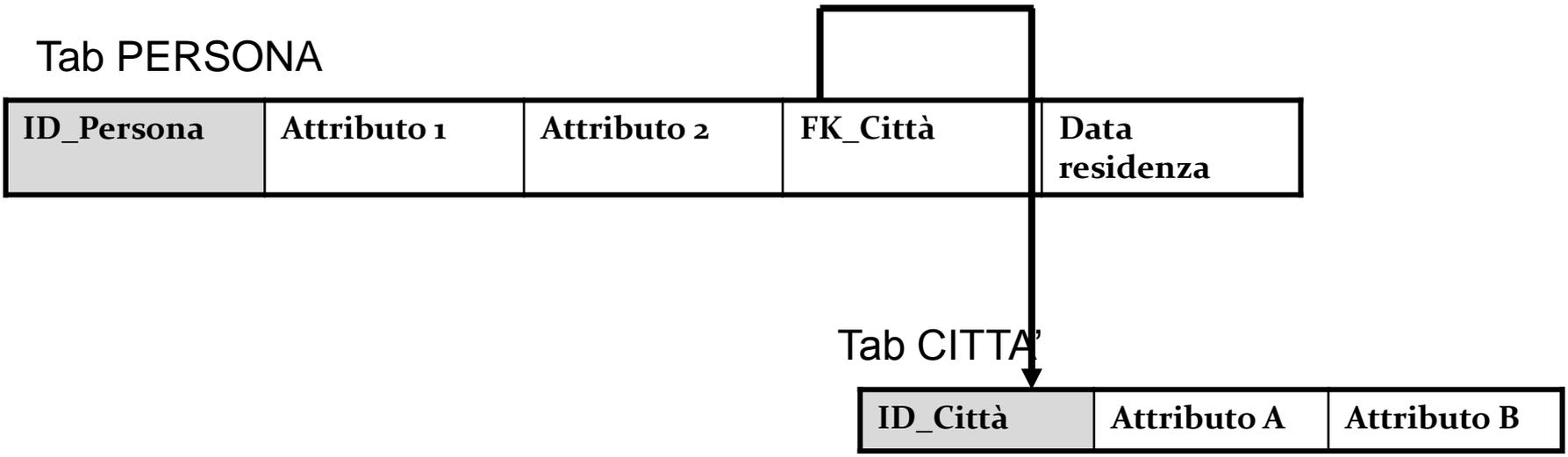


# RELAZIONE 1:N

## MODELLO ENTITÀ-RELAZIONE

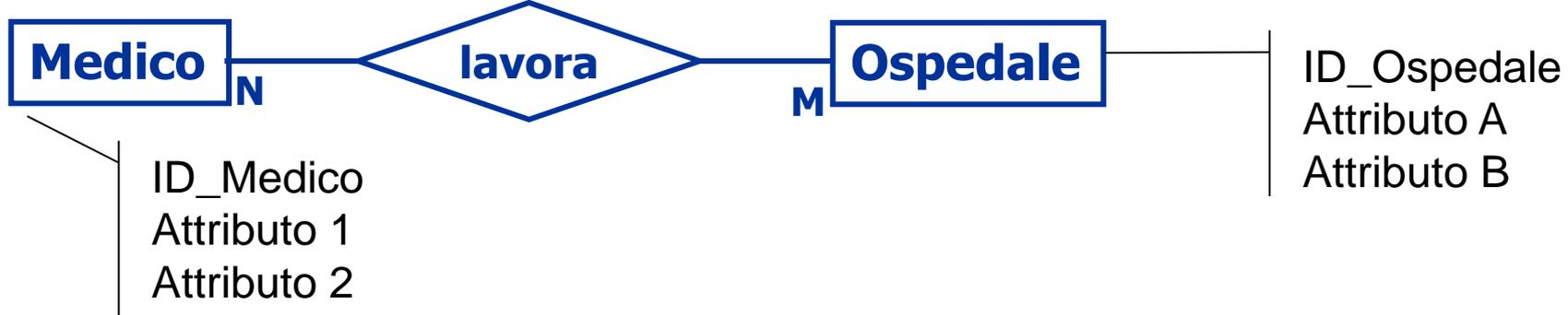


## MODELLO RELAZIONALE

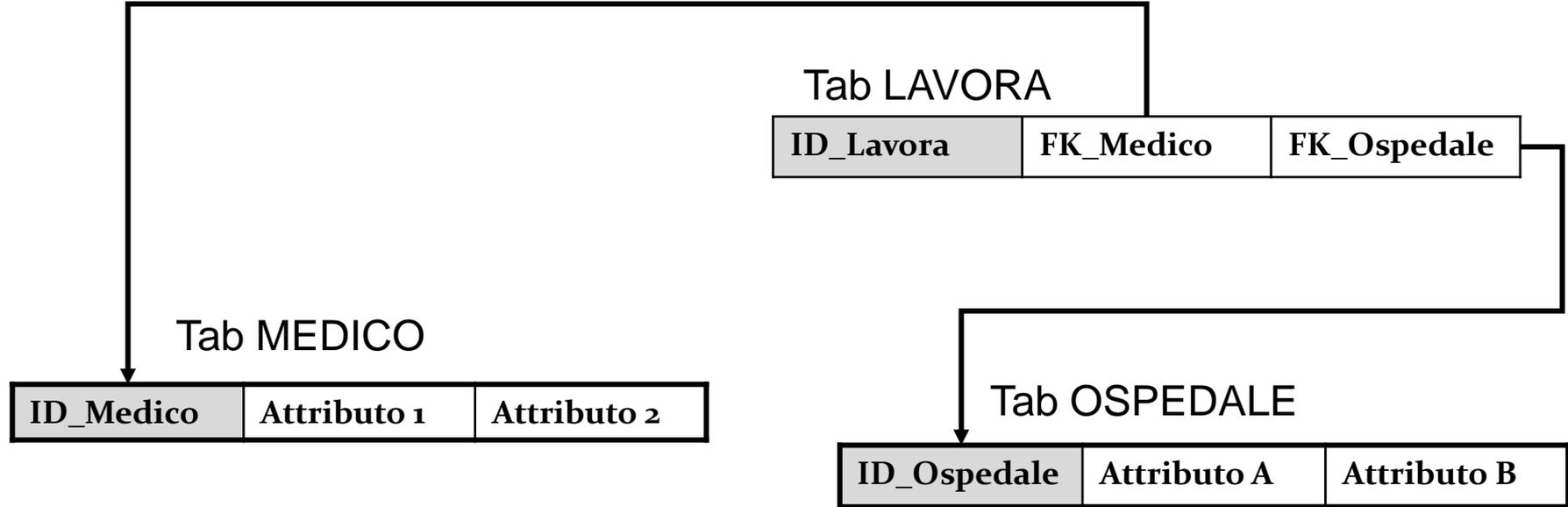


# RELAZIONE N:M

## MODELLO ENTITÀ-RELAZIONE



## MODELLO RELAZIONALE



# Linguaggi per le basi di dati

- I linguaggi per le basi di dati servono ad effettuare operazioni su schemi e istanze
  - **Linguaggi di Definizione dei Dati** (DDL: Data Definition Language)
    - Definiscono gli schemi logici, esterni e fisici e le autorizzazioni per l'accesso
  - **Linguaggi per la Manipolazione dei Dati** (DML: Data Manipulation Language)
    - Utilizzati per l'interrogazione e l'aggiornamento delle istanze di basi di dati

# Linguaggi per le basi di dati

- **Linguaggi di Interrogazione (DML)**
  - **Permettono di trovare un dato basandosi su delle proprietà.**
    - » **Es. tabella STUDENTI → trovare tutti gli adolescenti affetti da bulimia nella provincia di Lecce**
  - **Permettono di trovare dati basandosi su confronti tra i contenuti di più tabelle.**
    - » **Es. tabella ELEZIONI, tabella PRESIDENTI → trovare gli anni in cui è stato eletto un presidente proveniente dalla Puglia**

# QUERY

- Dopo aver inserito i dati nel DB, dovremo aver modo di recuperarli in modo agevole ed ottimizzato
- Una **query** è un'interrogazione sul DB
  - fornisce come risultato un insieme di dati che soddisfano le condizioni imposte nella query
  - normalmente negli strumenti per la gestione dei DB si ha la possibilità di creare la query sia visualmente, sia scrivendola in un linguaggio apposito

# SQL: definizioni

- SQL (**Structured Query Language**) è un linguaggio per la formulazione di query
- Una query scritta in SQL ha la forma **Select-From-Where**
  - **SELECT**: per indicare i campi richiesti (\* per tutti)
  - **FROM**: per indicare su quali tabelle si deve effettuare la query
  - **WHERE**: per indicare i vincoli

# SQL esempio

Ad esempio se volessimo estrarre le informazioni dello studente la cui matricola è 69002, potremmo scrivere la query in SQL:

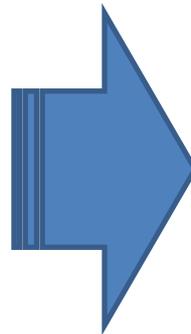
```
SELECT Nome, Cognome  
FROM Studente  
WHERE matricola = 69002
```

- Il risultato sarà: Carlo Verdi

# Linguaggi per le basi di dati

- **SQL**

```
SELECT Cognome, Nome  
FROM Medico, Ospedale  
WHERE Nome_Ospedale = "Fito Fazzi"
```



Cognome	Nome
Rossi	Mario
Verdi	Carlo

# Le interrogazioni

- Per visualizzare o estrarre i dati da una tabella occorre utilizzare le query di selezione, basate sull'uso del comando **SELECT**
- Come dice il nome, una tale interrogazione permette di 'selezionare' dei particolari dati di una tabella e visualizzarli
- La selezione viene fatta sulla base dei nomi delle colonne della tabella
- La parola chiave **FROM** permette di scegliere la tabella del database da cui si vogliono estrarre i dati

```
select ListaCampi  
from Tabella
```

# Esempio di interrogazione

Nome	Età	Reddito
Angelo	27	48
Marco	26	45
Piero	19	50

persone

Individuare tutti i nomi e i rispettivi redditi

```
select Nome, Reddito  
from Persone;
```



Nome	Reddito
Angelo	48
Marco	45
Piero	50

# Criteri di ordinamento dei dati

- Per avere in output un insieme di righe ordinate secondo un certo campo, si può utilizzare il comando **Order by**

```
select ListaCampi  
from Tabella  
order by campox
```

- **Esempio**

persone

Nome	Età	Reddito
------	-----	---------

Individuare nomi e redditi ordinati per Reddito

```
select Nome, Reddito  
from persone  
order by Reddito;
```

# Come filtrare i dati

- La clausola **WHERE** permette di specificare una condizione che deve essere soddisfatta dai dati estratti
- Su ciascuna riga vengono valutati i valori dei campi coinvolti nella condizione: solo se essi verificano la condizione i campi di tale riga specificati dalla SELECT vengono selezionati

```
select ListaCampi  
from Tabella  
where Condizione
```

La condizione è un predicato (o più) che usa gli operatori =, <>, <, >, <=, >=

# Esempio di filtro

Nome	Età	Reddito
Angelo	27	48
Marco	26	45
Piero	19	50

persone

Individuare nomi e redditi delle persone che guadagnano più di 45

```
select Nome, Reddito  
from Persone  
where Reddito > 45;
```



Nome	Reddito
Angelo	48
Piero	50

# Filtraggio avanzato

- Attraverso gli operatori **AND** ed **OR** è possibile combinare più criteri di selezione
- L'operatore **IN** permette di specificare la condizione di appartenenza ad un certo insieme di valori (eventualmente risultato dell'esecuzione di un'altra query)
- L'operatore **NOT** è utilizzabile per creare delle condizioni che sono il negato di altre
- Per verificare se un attributo è **null** si usa l'operatore **IS**
  - ▶ Es. *Attributo is null* oppure *Attributo is not null*

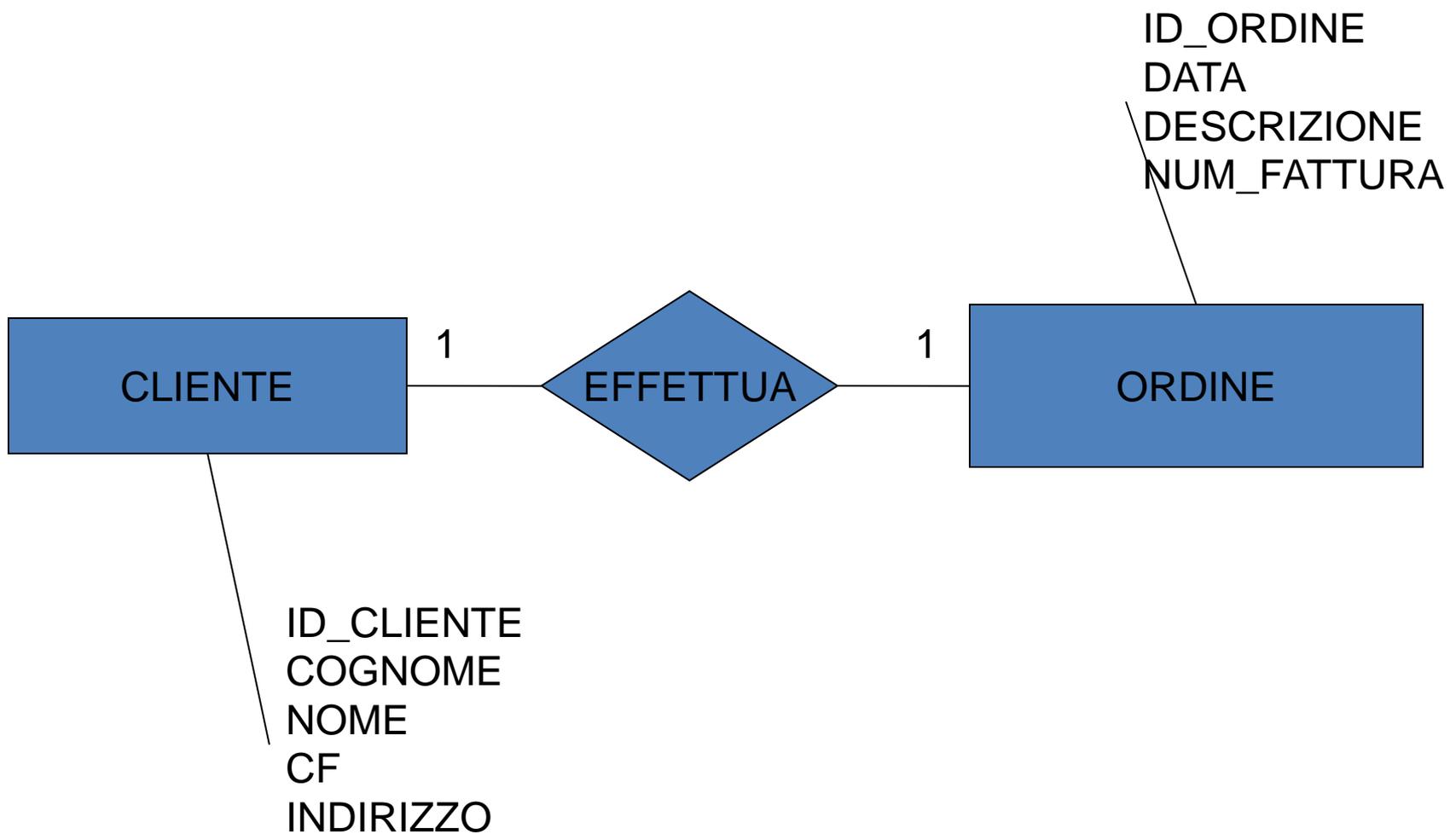
# TABELLE DI VERITA': OR

<b>OR</b>	F	V
F	F	V
V	V	V

# TABELLE DI VERITA': AND

<b>AND</b>	F	V
F	F	F
V	F	V

# MODELLO E-R



# MODELLO RELAZIONALE: OPZ. 1

TAB: CLIENTE

<u>ID_CLIENTE</u>	COGNOME	NOME	INDIRIZZO	CF	FK_ORDINE
-------------------	---------	------	-----------	----	-----------

TAB: ORDINE

<u>ID_ORDINE</u>	DATA	DESCRIZIONE	NUM_FATTURA
------------------	------	-------------	-------------



# MODELLO RELAZIONALE: OPZ. 2

TAB: CLIENTE

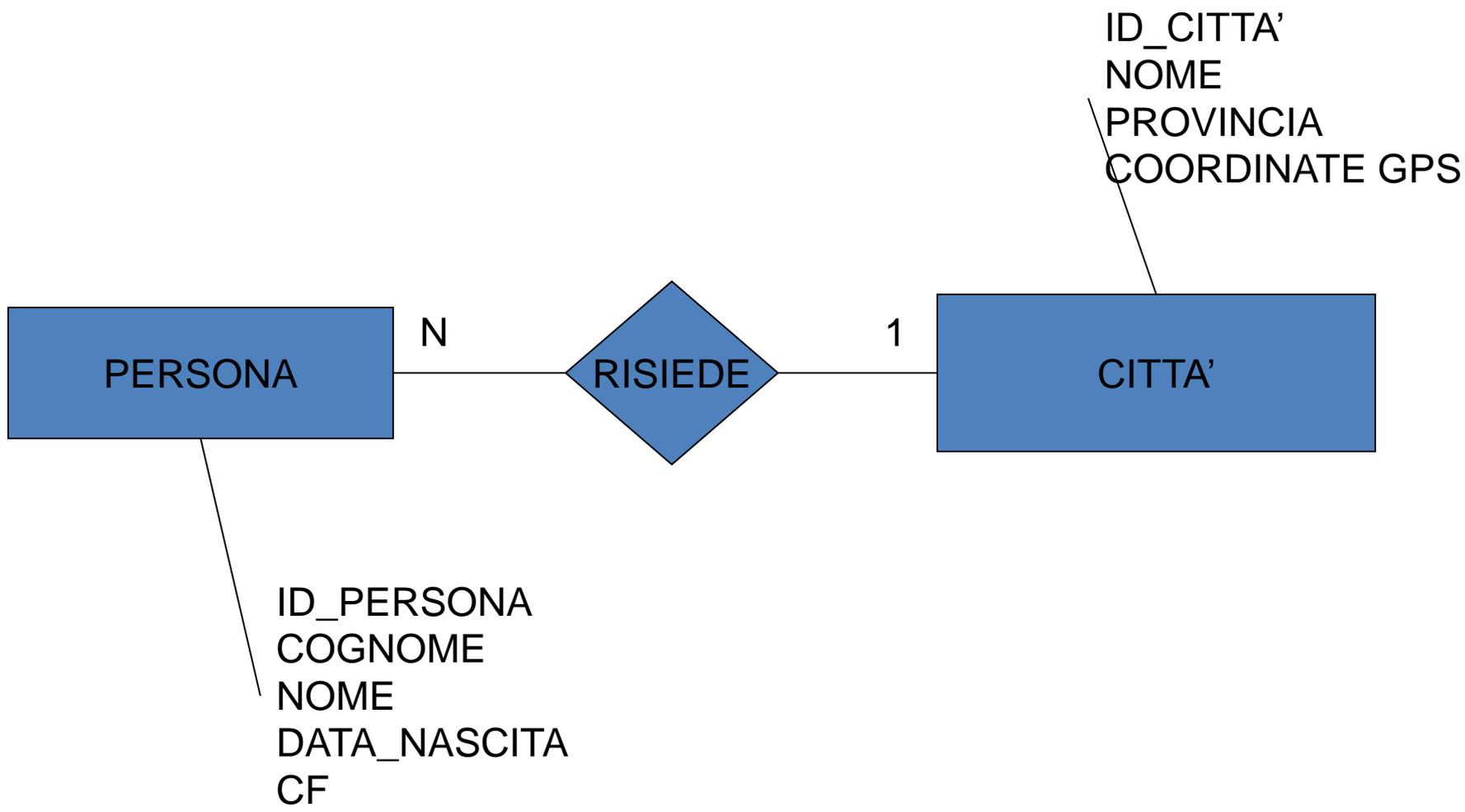
<u>ID_CLIENTE</u>	COGNOME	NOME	INDIRIZZO	CF
-------------------	---------	------	-----------	----

TAB: ORDINE

<u>ID_ORDINE</u>	DATA	DESCRIZIONE	NUM_FATTURA	FK_CLIENTE
------------------	------	-------------	-------------	------------



# MODELLO E-R



# MODELLO RELAZIONALE

TAB: PERSONA

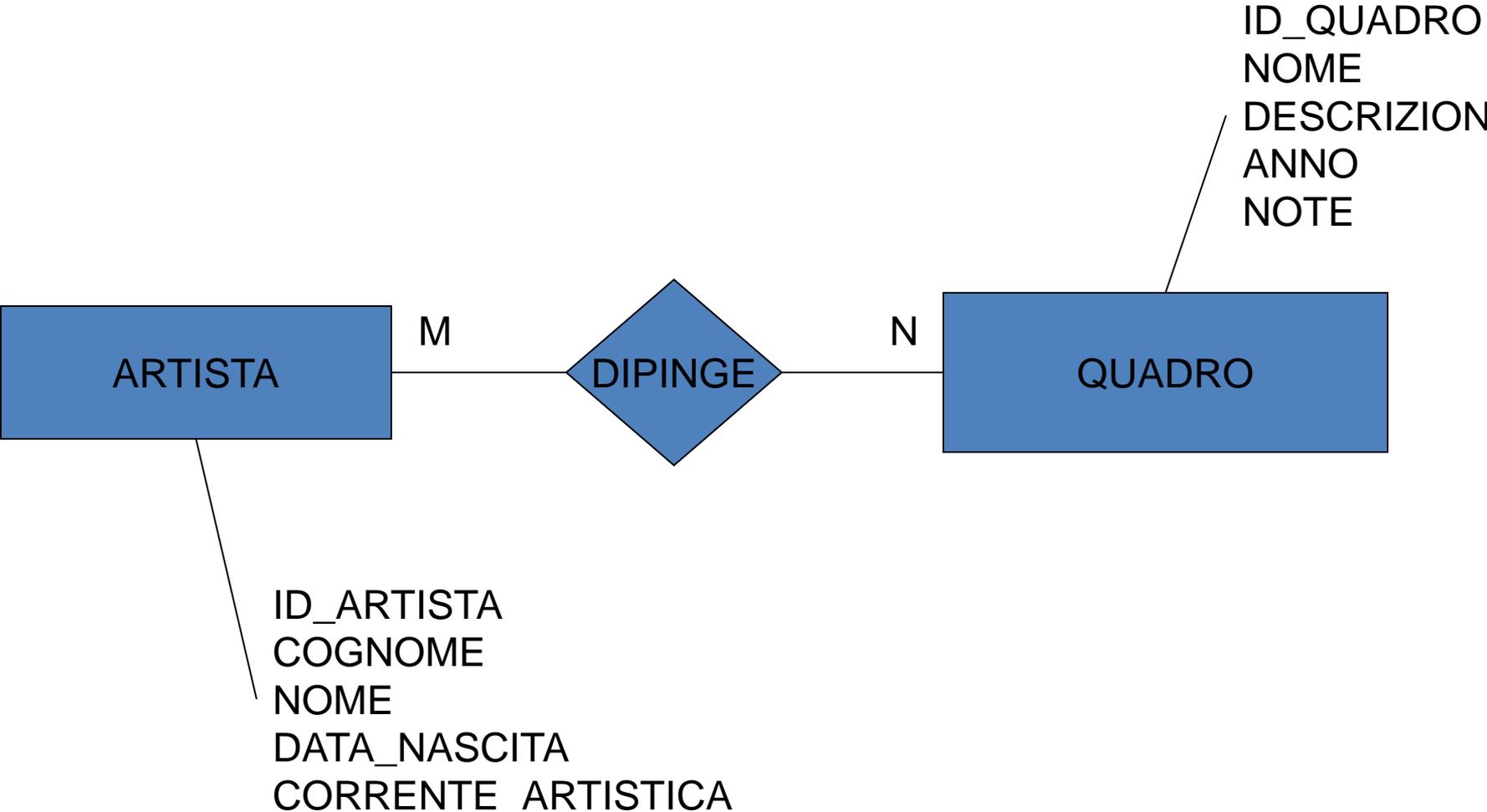
<u>ID_PERSONA</u>	COGNOME	NOME	DATA_NASCITA	CF	FK_CITTA'
-------------------	---------	------	--------------	----	-----------

TAB: CITTA'

<u>ID_CITTA'</u>	NOME	PROVINCIA	COOR_GPS
------------------	------	-----------	----------



# MODELLO E-R



# MODELLO RELAZIONALE

TAB: ARTISTA

<u>ID_ARTISTA</u>	COGNOME	NOME	DATA_NASCITA	CORRENTE_ARTISTICA
-------------------	---------	------	--------------	--------------------

TAB: DIPINGE

<u>ID_DIPINGE</u>	FK_ARTISTA	FK_QUADRO
-------------------	------------	-----------

TAB: QUADRO

<u>ID_QUADRO</u>	NOME	DESCRIZIONE	ANNO	NOTE
------------------	------	-------------	------	------

# Microsoft ACCESS

# ACCESS e Database

- Un **DataBase** (DB) è una raccolta di dati riguardanti un determinato argomento
- Raccolta di informazioni alfanumeriche
  - Numeri
  - Tabelle
  - Testo
  - Immagini
- **Le informazioni sono riunite in tabelle diverse.**
- **In genere ogni database è formato da più tabelle.**

# Database

- In ogni tabella sono presentate variabili con valori diversi
- Le *variabili* sono associate a *campi* nella tabella
- Tra le tabelle si possono stabilire relazioni
- Le informazioni vanno ricercate nelle tabelle attraverso
  - Interrogazioni (query) sulle tabelle del Database.

# Tabelle e Record

- Una **tabella**
  - è un contenitore per dati
  - ogni tabella rappresenta un raccolta di informazioni su uno specifico argomento

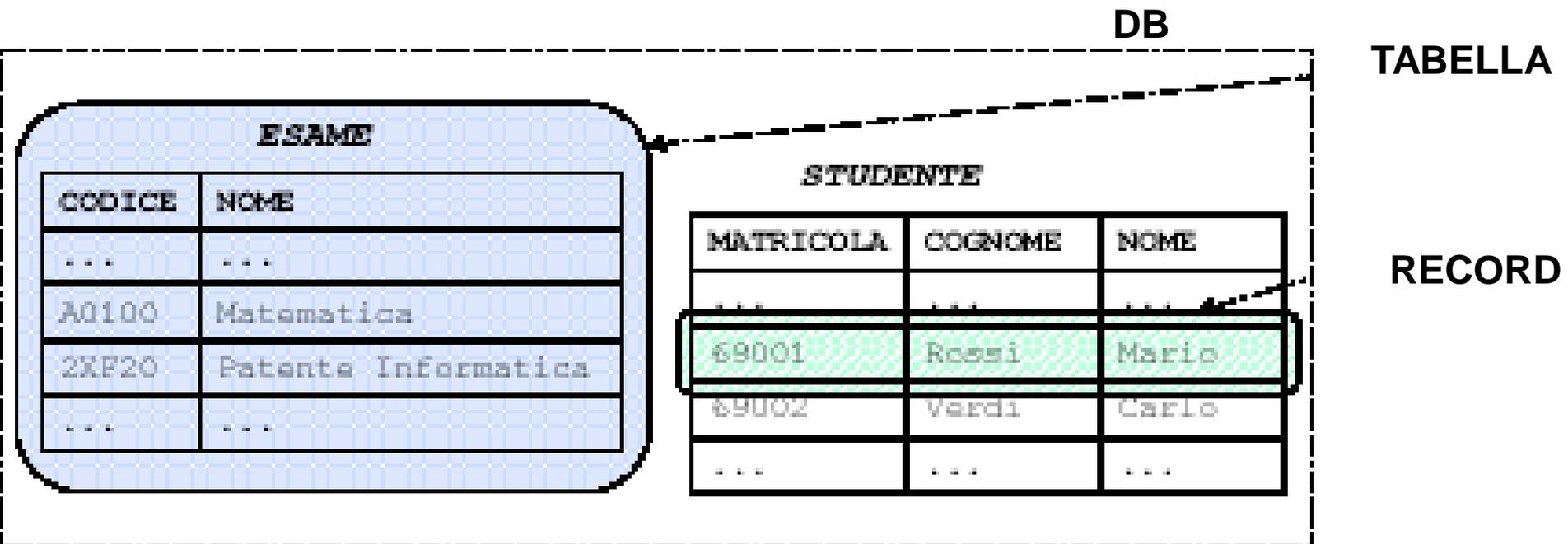
ad esempio possiamo avere una tabella per gli PAZIENTE  
ed una per le PATOLOGIE

- Un **record**
  - è una singola riga di una tabella
  - ci permette di identificare un preciso insieme di dati, all'interno di tutti quelli contenuti nella tabella

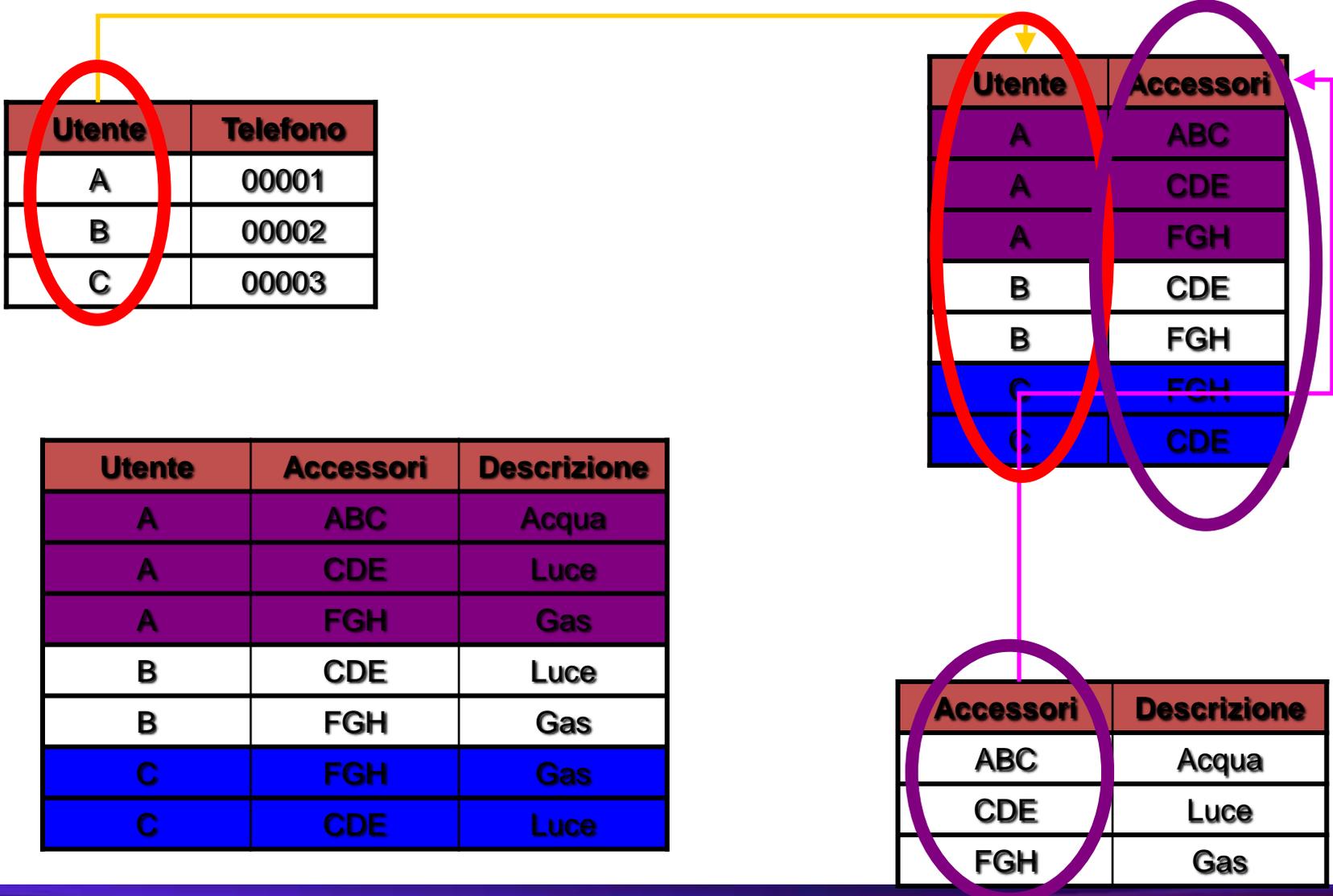
ad esempio nella tabella PAZIENTE ci sarà  
il record relativo a “MARIO ROSSI”

# DB – tabelle – record

- Un **DB** è composto da diverse tabelle
- Una **tabella** è composta da record omogenei
- Un **record** è composto da elementi



# Relazioni con Access



# Database relazionale

- Per recuperare dai memorizzati nelle tabelle si usano le interrogazioni
- Il risultato di un'interrogazione è una tabella che...

*Seleziona i dati presenti nelle tabelle  
se soddisfano al criterio di selezione*

# Microsoft Access

Microsoft Access

File Edit View Insert Tools Window Help

esempi : Database

Open Design New

Tables

Queries

Forms

Reports

Pages

Macros

Modules

Groups

Favorites

Create table in Design view

Create table by using wizard

Create table by entering data

Table1 : Table

Field Name	Data Type	Description
Num_cc	Number	
Nome	Text	
Indirizzo	Text	
Saldo	Number	

Attributi

Dettagli della definizione degli attributi

Field Properties

General Lookup

Field Size Long Integer

Format

Decimal Places Auto

Input Mask

Caption

Default Value 0

Validation Rule

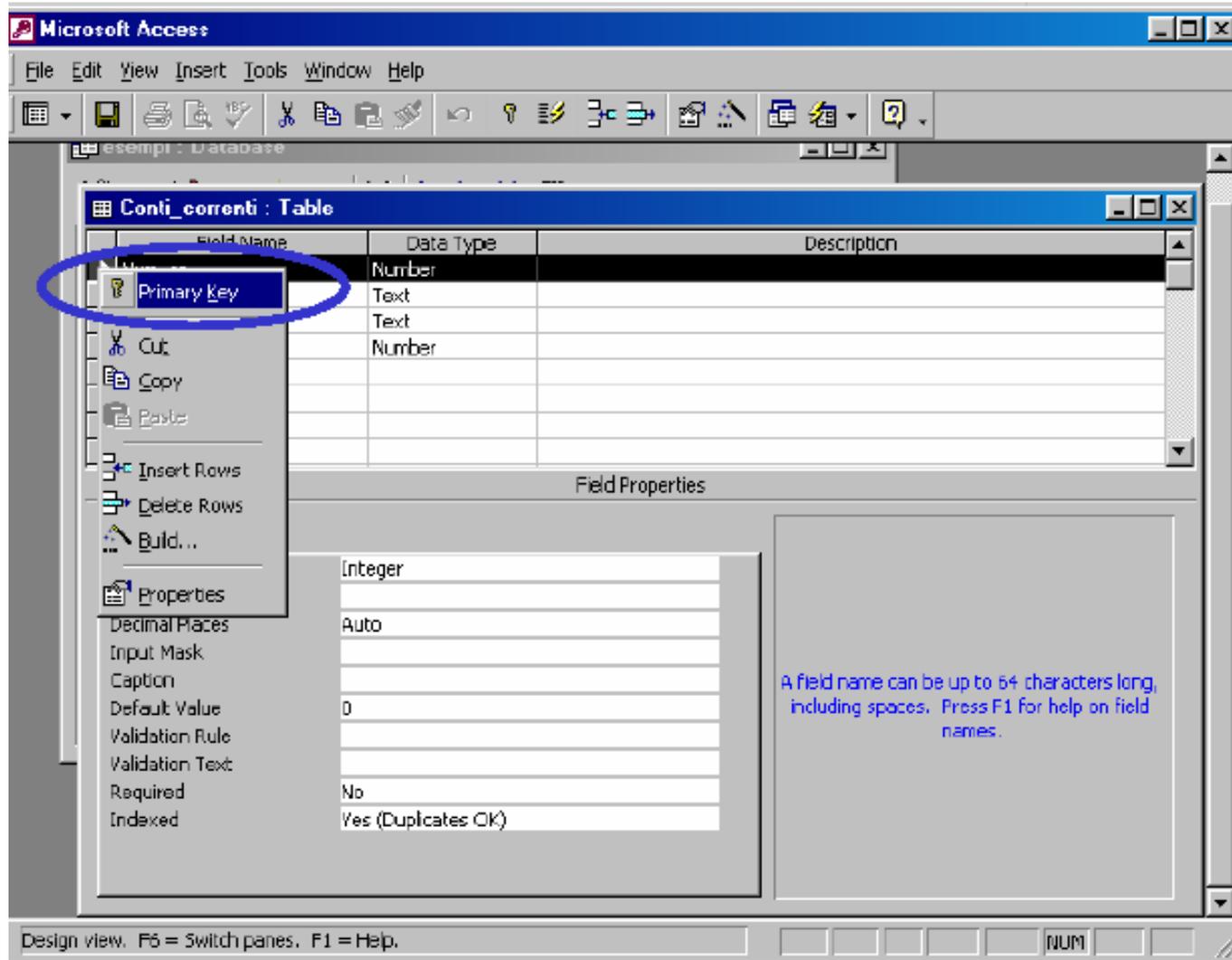
Validation Text

Required No

Indexed No

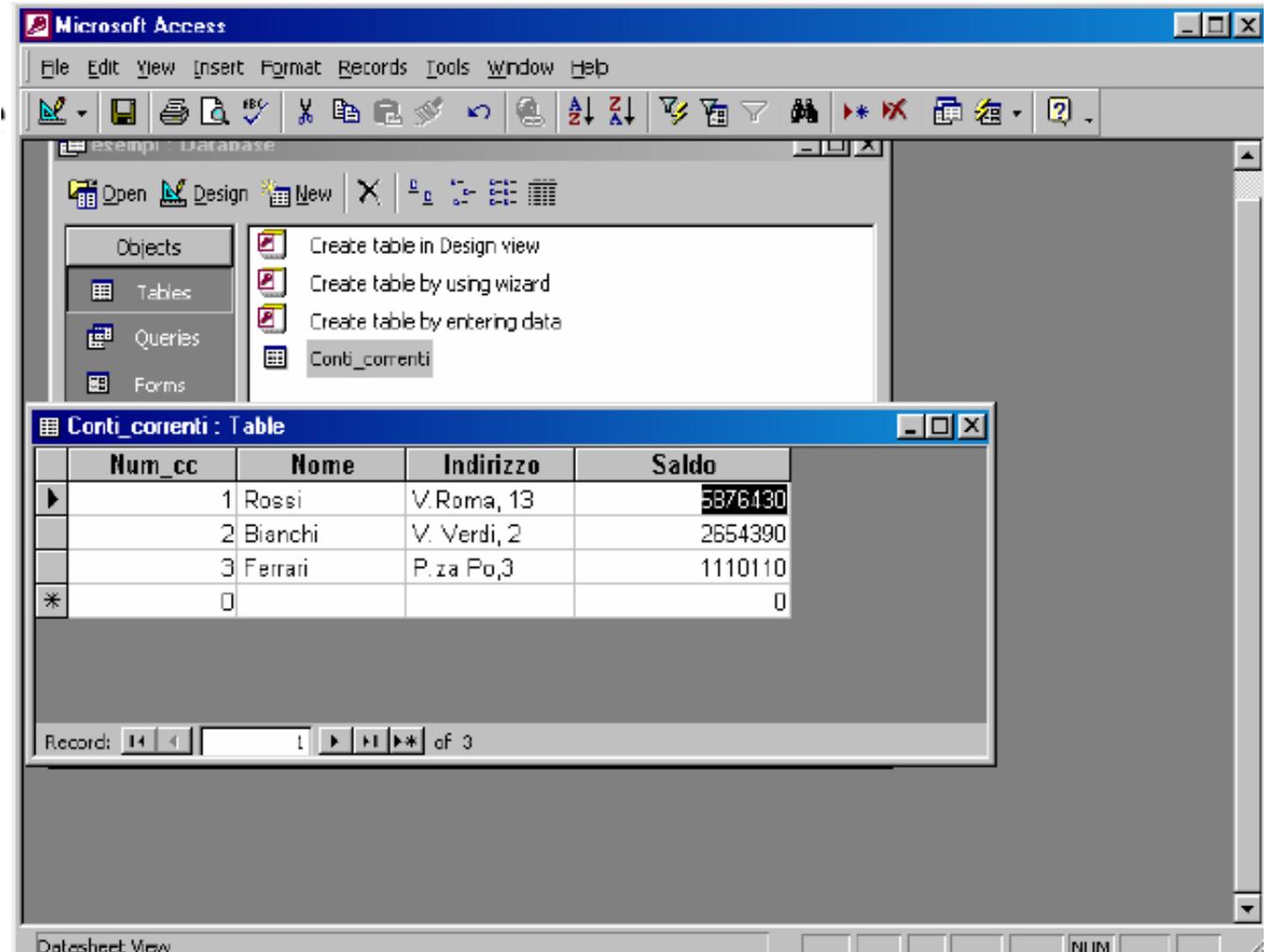
A field name can be up to 64 characters long, including spaces. Press F1 for help on field names.

# Definizione della chiave primaria



# Riempimento delle Tabelle

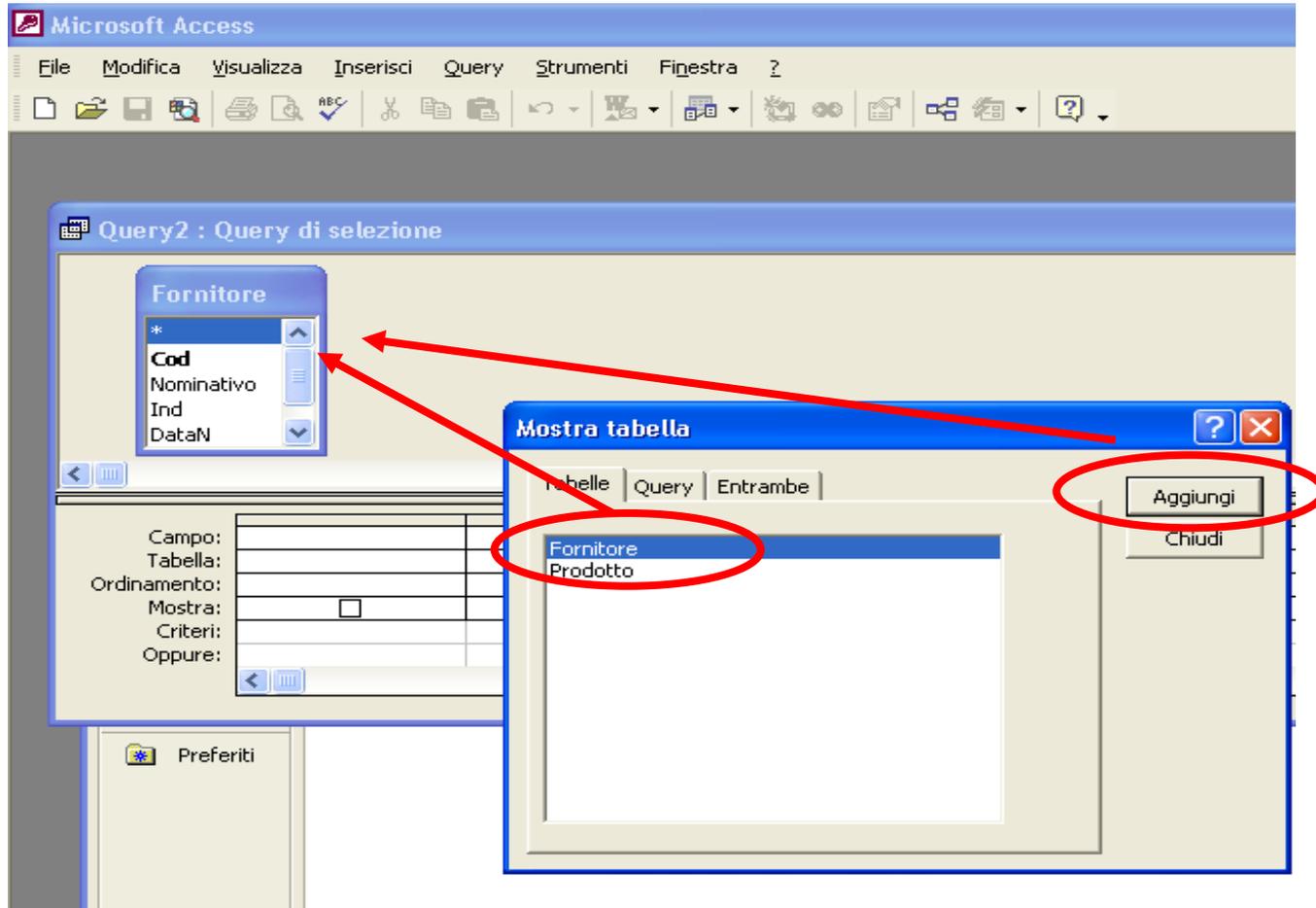
- Le tabelle si possono vedere come fogli di excel editabili direttamente



# Creazione di query



# Creazione di query



# Selezione

Selezione  
Criterio  
della  
selezione  
Occorre  
specificare  
gli attributi  
che fanno  
parte del  
risultato

Microsoft Access

Modifica Visualizza Inserisci Query Strumenti Finestra ?

Magazzino : Database (formato file di Access 2000)

Apri Struttura Nuovo

Oggetti

- Tabelle
- Query
- Maschere
- Report
- Pagine
- Macro
- Moduli

Gruppi

- Preferiti

Query1

Query1 : Query di selezione

Fornitore

- \* Cod
- Nominativo
- Ind

Prodotto

- \* Cod
- Descrizione
- Quantità

Campo:	Nominativo	Descrizione	Quantità
Tabella:	Fornitore	Prodotto	Prodotto
Ordinamento:			
Mostra:	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Criteri:			
Oppure:			

# Risultato della selezione

The screenshot shows the Microsoft Access interface. The main window is titled 'Magazzino : Database (formato file di Access 2000)'. The 'Oggetti' pane on the left shows 'Query1' selected and circled in red. The 'Query1 : Query di selezione' window displays a table with the following data:

	Nominativo	Descrizione	Quantità
▶	Claudia	Pane	10
	Claudia	Latte	12
	Stella	Pantaloni	22
*			

At the bottom of the window, the record navigation bar shows 'Record: 1 di 3'.

# Definizione delle relazioni

The screenshot shows the Microsoft Access interface for a database named "Magazzino : Database (formato file di Access 2000)". The "Relazioni" (Relationships) window is open, displaying two tables: "Fornitore" and "Prodotto".

- The "Fornitore" table has fields: Cod (primary key), Nominativo, Ind, and DataN.
- The "Prodotto" table has fields: Cod, Descrizione, Quantità, Prezzo, and CodF.

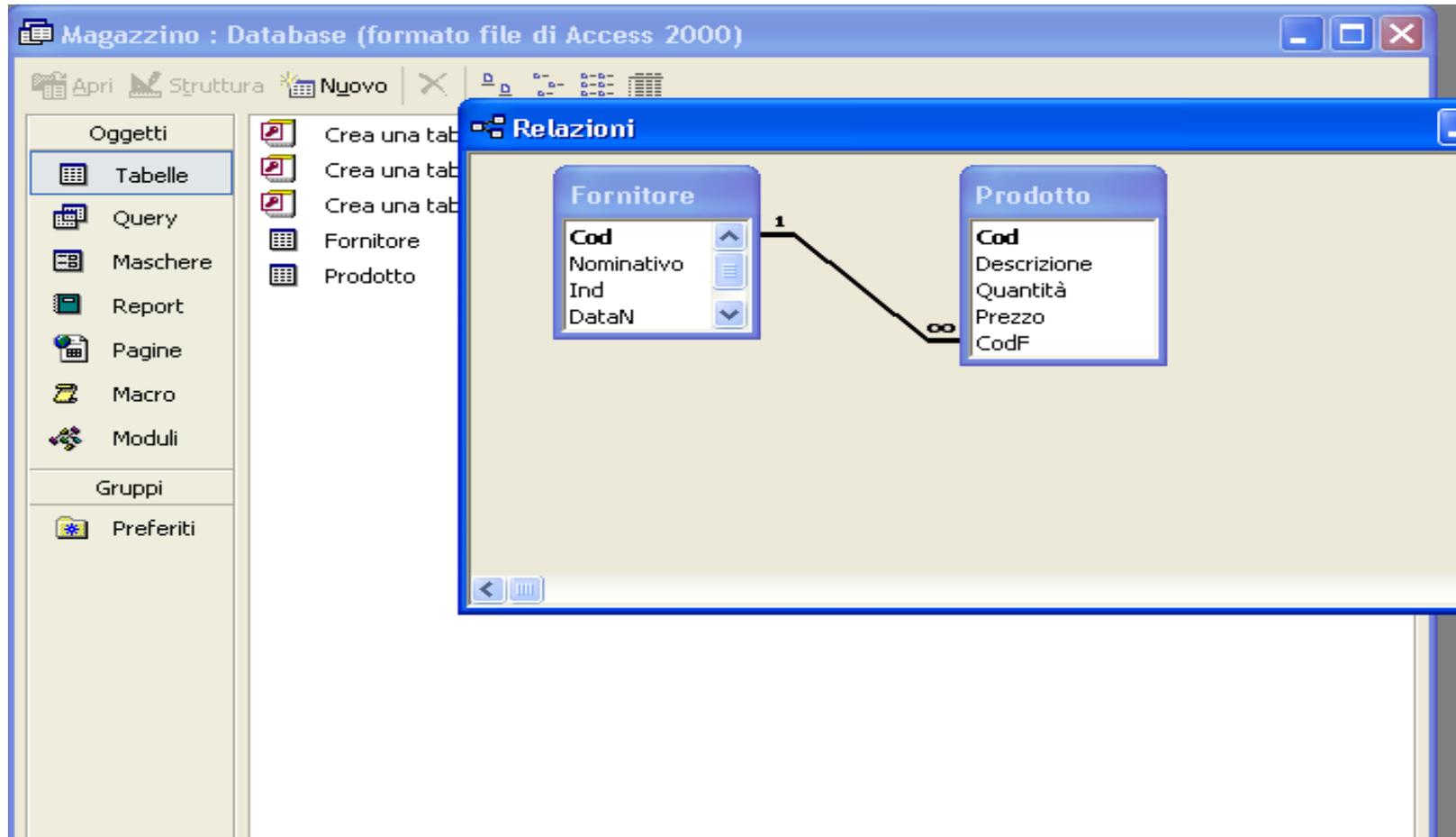
A relationship is established between the "Cod" field in the "Fornitore" table and the "CodF" field in the "Prodotto" table. The relationship is labeled "1" on the "Fornitore" side and "∞" on the "Prodotto" side, indicating a one-to-many relationship. Red circles highlight the "Cod" field in "Fornitore" and the "CodF" field in "Prodotto". A red arrow points from this relationship to the "Modifica relazioni" (Modify Relationships) dialog box.

The "Modifica relazioni" dialog box shows the following configuration:

- Tabella/query: Fornitore
- Tabella/query correlata: Prodotto
- Field: Cod
- Field correlata: CodF
- Applica integrità referenziale
- Aggiorna campi correlati a catena
- Elimina record correlati a catena
- Tipo relazione: Uno-a-molti

Buttons on the right include: OK, Annulla, Tipo join..., and Crea nuova..

# Diagramma delle relazioni

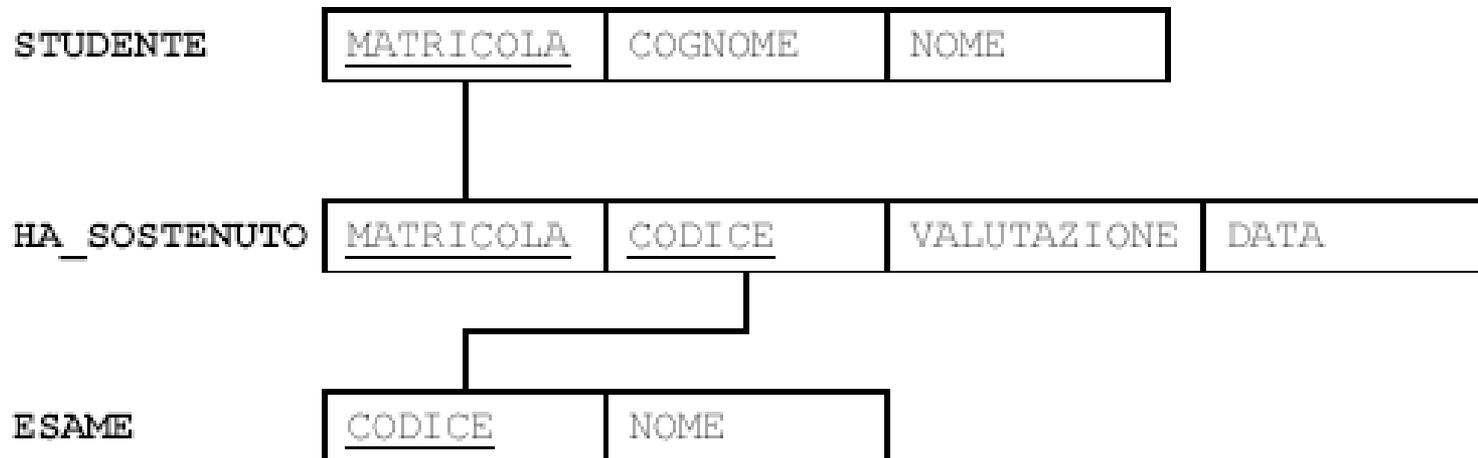


# Altre caratteristiche di Access

- Oltre alle caratteristiche viste, Access permette di fare molte altre cose, tra cui:
  - Definire delle **form**, ovvero delle interfacce per l'input/output dei dati (MASCHERE)
  - Definire dei **report**, cioè degli output adatti alla stampa
  - Importare/esportare dati in formato excel
  - ...

# Esempio

- Nell'esempio dell'archivio universitario la relazione è diventata una tabella ponte
- I campi sottolineati rappresentano le chiavi



# Esercizio 1: Tabelle

- Si vuole costruire un DataBase che consenta di gestire gli studenti che sono iscritti ad una università

I dati da memorizzare sono:

- **STUDENTE** (matricola, cognome, nome, annonascita)
- **FACOLTA'** (.....)
- **CORSO DI LAUREA** (.....)
  - Creare il DB università (universita.mdb)
  - Creare la struttura delle tabelle e le necessarie relazioni

## Esercizio 2: Relazioni

- **Dopo aver definito le diverse tabelle, bisogna indicare come le informazioni sono collegate tra loro**
- **Per aprire la finestra delle relazioni**
  - icona nella barra degli strumenti, oppure
  - menu “strumenti”, comando “relazioni...”
- **Scegliere le tabelle che vogliamo collegare**
- **Trascinare il campo di una tabella sul campo collegato della seconda tabella**

## Struttura query

- **Generazione query tramite “visualizzazione struttura”:**
- **Scegliere le tabelle interessate nell’interrogazione**
- **Trascinare i campi su cui si vuole operare**
- **Inserire eventuali criteri (es. data > 01/01/2002);**
- **Evidenziare i campi che si desidera visualizzare (es. non selezionare la data)**
- **Salvare la query**

**Data Base Relazionali**  
**- FINE -**

**Ing. Maria Grazia Celentano**  
**[www.mariagraziacelentano.it](http://www.mariagraziacelentano.it)**