
INTRODUZIONE AI DATABASE

Archivi

- Un **archivio** è una **raccolta strutturata di informazioni**.
 - Dizionario
 - Pagine gialle
 - Elenco libri di una biblioteca
- Le informazioni sono **strutturate**
 - Per ogni nominativo nelle pagine gialle troviamo una serie di dati fissi (Nome, cognome, telefono, indirizzo...)
- Un archivio può essere cartaceo o elettronico
- Un archivio elettronico è spesso chiamato **database** o **base di dati**.

Basi di Dati

- Lo scopo di una Base di Dati (BD) è quello di memorizzare informazioni in modo strutturato e di permetterne la modifica e il reperimento da parte di utenti e applicazioni
- Usiamo quotidianamente le BD quando:
 - accediamo al Sistema Informativo della azienda o istituzione in cui lavoriamo
 - accediamo al nostro conto corrente via Internet
 - prenotiamo un volo o albergo via Internet

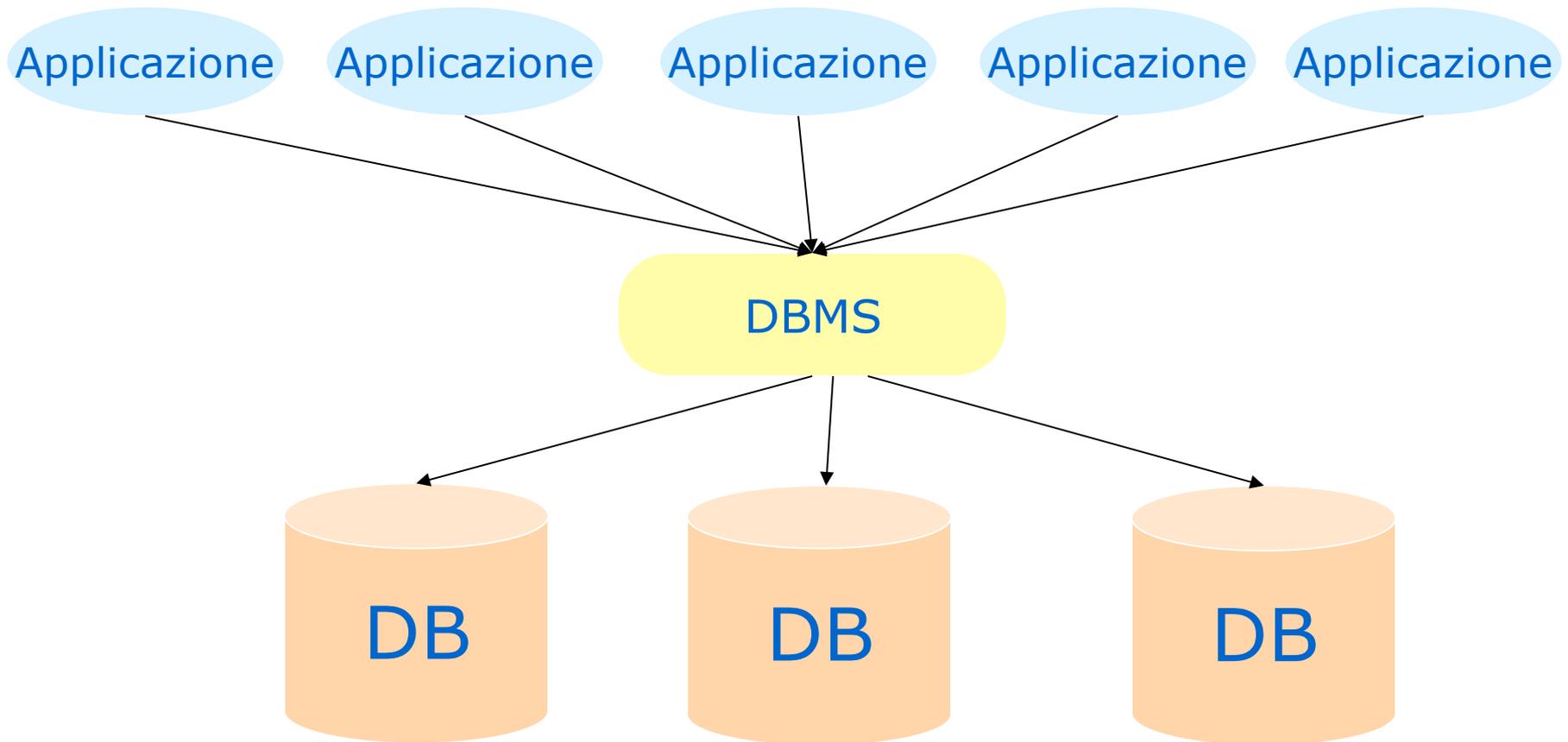
Basi di dati

- Le basi di dati hanno numerosi vantaggi rispetto agli archivi tradizionali
 - Ricerche più versatili e veloci
 - Sulle pagine gialle posso fare solo ricerche per nome
 - Sulle pagine gialle elettroniche posso fare ricerche anche per vicinanza o numeri di telefono
 - Possibilità di ordinare e filtrare i dati
 - Minore occupazione di spazio

DBMS

- un Data Base Management System (**DBMS**) è una applicazione che permette di **creare e gestire** delle **Basi di Dati**
- l'effettivo **utilizzo** delle **BD** avviene invece per mezzo di **applicativi** che **sfruttano** il **DBMS** per accedere alle **BD** stesse

DBMS



DBMS

- il **DBMS** realizza una **separazione** tra **Base Dati** e le **applicazioni** che la utilizzano
- il **contenuto** della BD **varia** normalmente molto spesso (es. prenotazioni di voli)
- la **struttura** della BD, invece, tende a essere **molto stabile** (es. tipi di informazioni con cui descriviamo una prenotazione)
- le **applicazioni** che utilizzano la BD variano normalmente meno frequentemente del suo contenuto e più della sua struttura

Indipendenza Fisica

- la separazione tra applicazioni e DB realizzata dal **DBMS** è detta **indipendenza fisica**
- se vale l'indipendenza fisica è possibile apportare modifiche al modo in cui i dati sono memorizzati senza modificare le applicazioni
- ad es. i dati vengono spostati su un altro disco, o segmentati su più dischi o indicizzati per velocizzare il reperimento

Dati e informazioni

- Per poter rappresentare le informazioni in un database occorre prima capirne la struttura in modo da distinguere i dati che la compongono.
 - Informazioni:
 - Il libro *Harry Potter e la pietra filosofale*, dell'autrice Joanne K. Rowling, è di genere fantasy, si compone di 293 pagine e il suo prezzo è di € 16.80.
 - Il libro *Dieci Piccoli Indiani*, di Agatha Christie, è un giallo, si compone di 210 pagine, e costa € 7.70.
 - Struttura
 - Ogni libro è caratterizzato da un titolo, un autore, un genere, un numero di pagine e un prezzo.
 - Inoltre... numero di pagine è un valore numerico intero, il prezzo è un valore numerico con due cifre dopo la virgola, titolo e autori sono sequenze di caratteri

Dati e informazioni

■ Informazione

- Il libro *Dieci Piccoli Indiani*, di Agata Christie, è un giallo, si compone di 210 pagine, e costa € 7.70.

Dati e informazioni

■ Informazione

- Il libro *Dieci Piccoli Indiani*, di Agata Christie, è un giallo, si compone di 210 pagine, e costa € 7.70.

- Dieci Piccoli Indiani

- Agatha Christie

- Giallo

- 210

- 7.70

Modello Relazionale

- la **quasi totalità** dei moderni **DBMS** permette la gestione di BD basate sul **Modello Relazionale (MR)**
- la proposta del MR risale al **1970** ed è opera del matematico **E.F. Codd**
- dato l'elevato livello di astrazione del MR, le prime realizzazioni commerciali risalgono alla fine degli anni '70

Tabelle

- Un *database* (DB) registra dati
- Un *database relazionale* utilizza tabelle e descrive le relazioni fra i differenti tipi di dati
- In un database le tabelle sono una lista di valori con dei **vincoli**
 - i vincoli **limitano** il tipo di dati(e.g., interi, razionali con virgola mobile, ecc.) che possono essere inseriti nel DB, ma rendono possibile eseguire operazioni molto più complesse e potenti

Tabelle

- Nei database relazionali i dati sono strutturati in tabelle.
 - Ogni **riga** (chiamata anche **entità**, **record** o **istanza**) della tabella è una singola informazione che vogliamo memorizzare.
 - Ogni **colonna** (o **attributo**, **campo**) è un singolo dato che vogliamo memorizzare per tutte le informazioni
- Ecco la rappresentazione tabellare delle informazioni viste precedentemente

Titolo	Genere	Pagine	Prezzo	Autore
Harry Potter e la pietra filosofale	Fantasy	293	16,80	J. K. Rowling
Dieci piccoli indiani	Giallo	530	29,90	A. Christie

Tabelle (2)

- Qualsiasi cosa possa essere identificata da un numero prefissato di caratteristiche (*attributi*)
 - Gli attributi hanno un nome e un valore
 - I valori rappresentano i dati memorizzati nella tabella
- Per creare una tabella si specifica il **nome**, i suoi **attributi** e il **tipo di valore** che si può inserire in ogni attributo

Proprietà delle entità

■ *Istanze di tabelle:*

- Una tabella è inizialmente vuota. Ha un nome e intestazioni di colonna (i nomi degli attributi) ma le righe sono vuote
- Ogni riga rappresenta un'entità
- Un'*istanza* di database è una tabella con un insieme specifico di righe

Proprietà delle entità (cont.)

- Strutture, contenuti e metadati
 - separiamo la struttura dell'informazione dall'informazione stessa
 - i *metadati* di una tabella includono almeno il suo nome, i nomi degli attributi, il tipo di valori che ogni attributo può assumere e la chiave primaria
 - alcune proprietà non sono metadati:
l'ordine delle righe ad esempio non è importante

Tabelle

- per esempio un insieme di studenti (entità reali) può essere rappresentato da una tabella con gli opportuni attributi

Tabella Studenti

Matricola	Cognome	Nome	Sesso	DataDiNascita	LuogoDiNascita
300000	Rossi	Mario	M	01/01/1987	Torino
300001	Verdi	Chiara	F	01/02/1987	Roma
300002	Rossi	Daria	F	01/01/1987	Torino
300003	Cancelli	Chiara	F	10/10/1987	Milano
300004	Votantonio	Raffaella	F	10/02/1985	Roma

Record - entità

campo

Employee

Empl Id	Name	Address	SSN
25X15	Joe E. Baker	33 Nowhere St.	111223333
34Y70	Cheryl H. Clark	563 Downtown Ave.	999009999
23Y34	G. Jerry Smith	1555 Circle Dr.	111005555
•	•	•	•
•	•	•	•
•	•	•	•

Nome	<u>GRIGIA</u>
Cibo	Crostacei
Peso	36.000 kg
Genere	<i>Eschrichtius</i>
Specie	<i>robustus</i>
Lunghezza	13,7-14 m
Figura	

Nome	<u>ORCA</u>
Peso	5.400 kg
Genere	<i>Orcinus</i>
Specie	<i>orca</i>
Cibo	Salmoni, Foche
Lunghezza	6,7-8,2 m
Figura	

Genere	<i>Balaenoptera</i>
Specie	<i>musculus</i>
Cibo	Krill
Nome	<u>BLU</u>
Lunghezza	23-24,5 m
Peso	100.000 Kg
Figura	

Lunghezza	15-18 m
Nome	<u>CAPODOGLIO</u>
Cibo	Calamari
Peso	40.000 Kg
Genere	<i>Physeter</i>
Specie	<i>macrocephalus</i>
Figura	

Nome	<u>MEGATTERA</u>
Cibo	Krill
Peso	36.000 Kg
Genere	<i>Megaptera</i>
Specie	<i>novaeangliae</i>
Lunghezza	13-15 m
Figura	

Figura 12.3 Alcune entità di tipo balena.

Campi

- Cosa può essere usato come valore per un campo?
 - Per ora siamo stati molto vaghi a proposito
- Puntualizziamo di nuovo un aspetto fondamentale:
 - Ogni campo può contenere **un solo dato elementare**.
Nel caso dei libri:
 - Il campo Titolo contiene solo il titolo del libro
 - Il campo Prezzo contiene solo il prezzo
 - Violare questa regola genera problemi
 - e.g., un solo campo che contiene sia il titolo che il prezzo, rendere impossibile ricercare tutti i libri che costano meno di 10 €

Tipi di dato

- Cosa può essere un dato elementare?
- In fase di creazione della tabella, i sistemi DBMS obbligano a scegliere un **tipo di dato**
 - Il tipo di dato limita i valori che possono essere messi in un campo.
 - Ci sono 5 categorie principali per i tipi di dato
 - **Numerico**: a sua volta divisi in
 - Intero, Virgola mobile e Virgola fissa
 - **Stringa**
 - **Temporale**
 - **Booleano**

Tipi Numerici (1)

■ INTEGER: valori interi

- Non si possono rappresentare tutti i numeri, solo quelli non troppo piccoli né troppo grandi. Di solito, da $-2^{31}-1$ a 2^{31} .
 - Vengono anche rese disponibili varianti che supportano intervalli più piccoli (**SMALLINT** da $-2^{15}-1$ a 2^{15}) o più grandi (**BIGINT** da $-2^{63}-1$ a 2^{63})
- **DECIMAL/NUMERIC**: valori non interi, con un numero di cifre dopo la **virgola fissato**
 - Ottimo per esempio per un campo destinato a contenere dei prezzi, poiché di solito si considerano solo 2 o 3 cifre dopo la virgola
 - Esempio con 2 cifre dopo la virgola: 12.45, -9.10

Tipi Numerici (2)

- **REAL**: valori non interi, in notazione scientifica.
 - Esempio: 3.5E-4 che vuol dire $3.5 \cdot 10^{-4}$ ovvero 0.00035
 - Il numero di cifre significative e l'intervallo di valori possibili per l'esponente è fissato.
 - I calcoli sono approssimati, non esatti e questo può causare spiacevoli conseguenze.

**Da evitare a meno che non sia
assolutamente necessario**

Tipi Stringa (1)

- Per **stringa** si intende una sequenza di caratteri alfanumerici
 - “luke” = { l, u, k, e } o “R2D2” = { r, 2, d, 2 }
- **CHAR**: stringa a lunghezza fissa
 - Se si tenta di inserire una stringa più corta, viene allungata con spazio
 - se la lunghezza fissata è 4
luke = { l, u, k, e }, han = { h, a, n, ' ' }, **obi-wan= ?**
 - Utile per campi come il codice fiscale che hanno una lunghezza costante

Tipi Stringa (2)

- **VARCHAR**: stringa a lunghezza variabile
 - Bisogna comunque specificare un lunghezza massima
 - Meno efficiente (il computer è lento.. per modo di dire.. a trattare questi dati)
 - Utile per campi che possono essere molto corti o molto lunghi (interi documenti, per esempio)

Altri tipi

■ Tipi temporali

- **DATE**: una data
 - dal 1 Gennaio 1000 al 31 Dicembre 9999
- **TIME**: un'ora del giorno, in ore, minuti e secondi
- **TIMESTAMP**: ora e data assieme
 - dalle 00:00:01 del 1 Gennaio 1970 alle 03:14:07 del 19 Gennaio 2038

■ Tipi booleani

- **BOOLEAN**: può assumere solo due valori (sì/no, vero/falso, 0/1 a seconda dei casi)

Il valore NULL (1)

- In alcuni casi, il valore di un campo può non essere noto.
 - Per rappresentare un valore non noto si potrebbero usare dei valori particolari per ogni tipi di dato:
 - numero civico: 0; codice fiscale: stringa vuota; data di nascita: 1/1/9999.

L'uso di “magic number” genera **SEMPRE**
problemi

Il valore NULL (2)

- Esiste un valore speciale, chiamato **NULL**, che indica proprio la mancanza di dati
 - Quando si specifica un campo, oltre al tipo bisogna indicare se accetta o meno valori NULL.

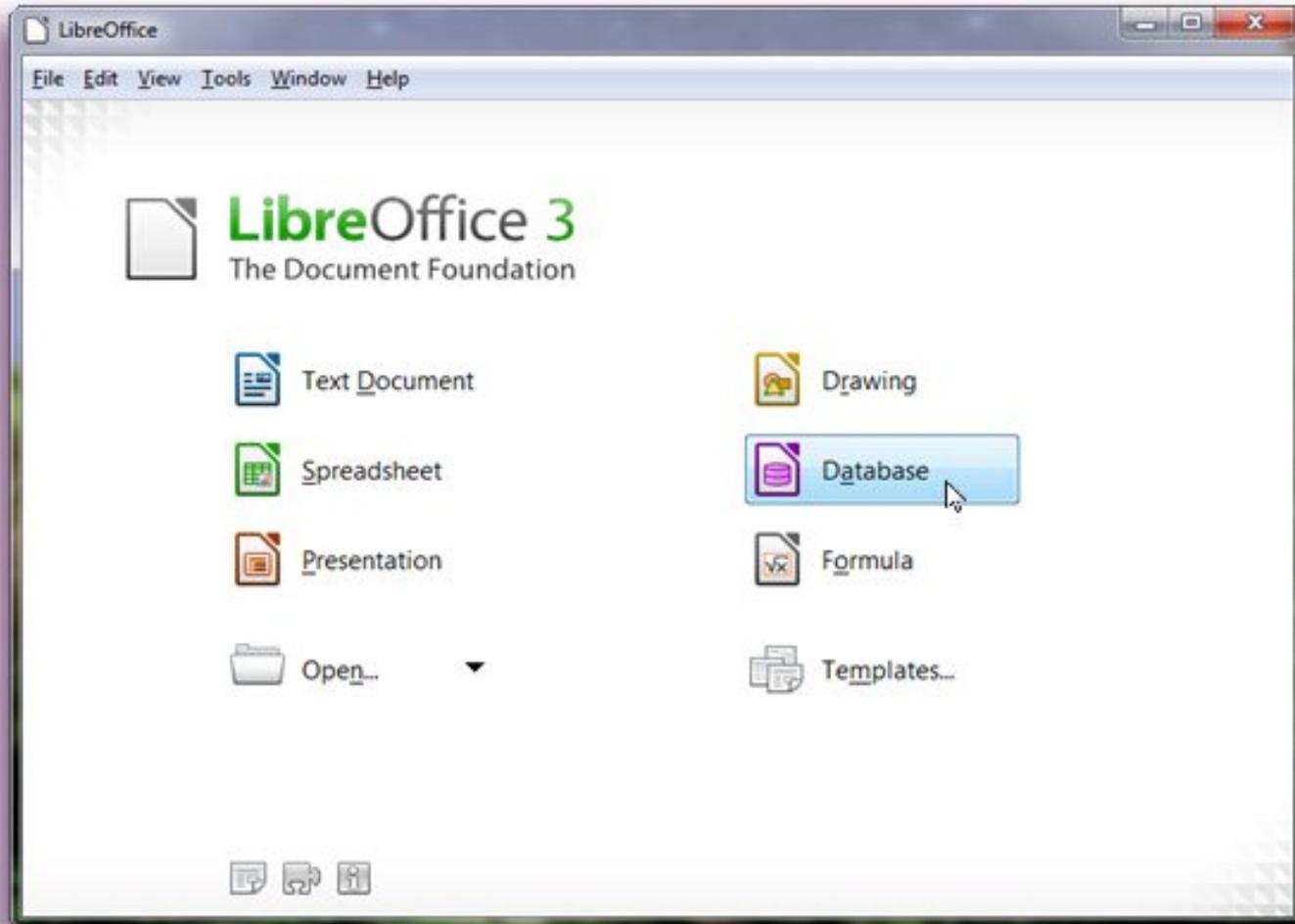
Balene

nome	Character, 15	<i>Nome comune</i>
genere	Character, 15	<i>Prima parte del nome scientifico</i>
specie	Character, 15	<i>Seconda parte del nome scientifico</i>
cibo	Character, 25	<i>Fonte primaria di cibo</i>
peso	Integer	<i>Peso medio di un adulto</i>
lunghezza	Character, 20	<i>Lunghezza tipica di un adulto</i>
figura	GIF	<i>Immagine di un esemplare adulto</i>

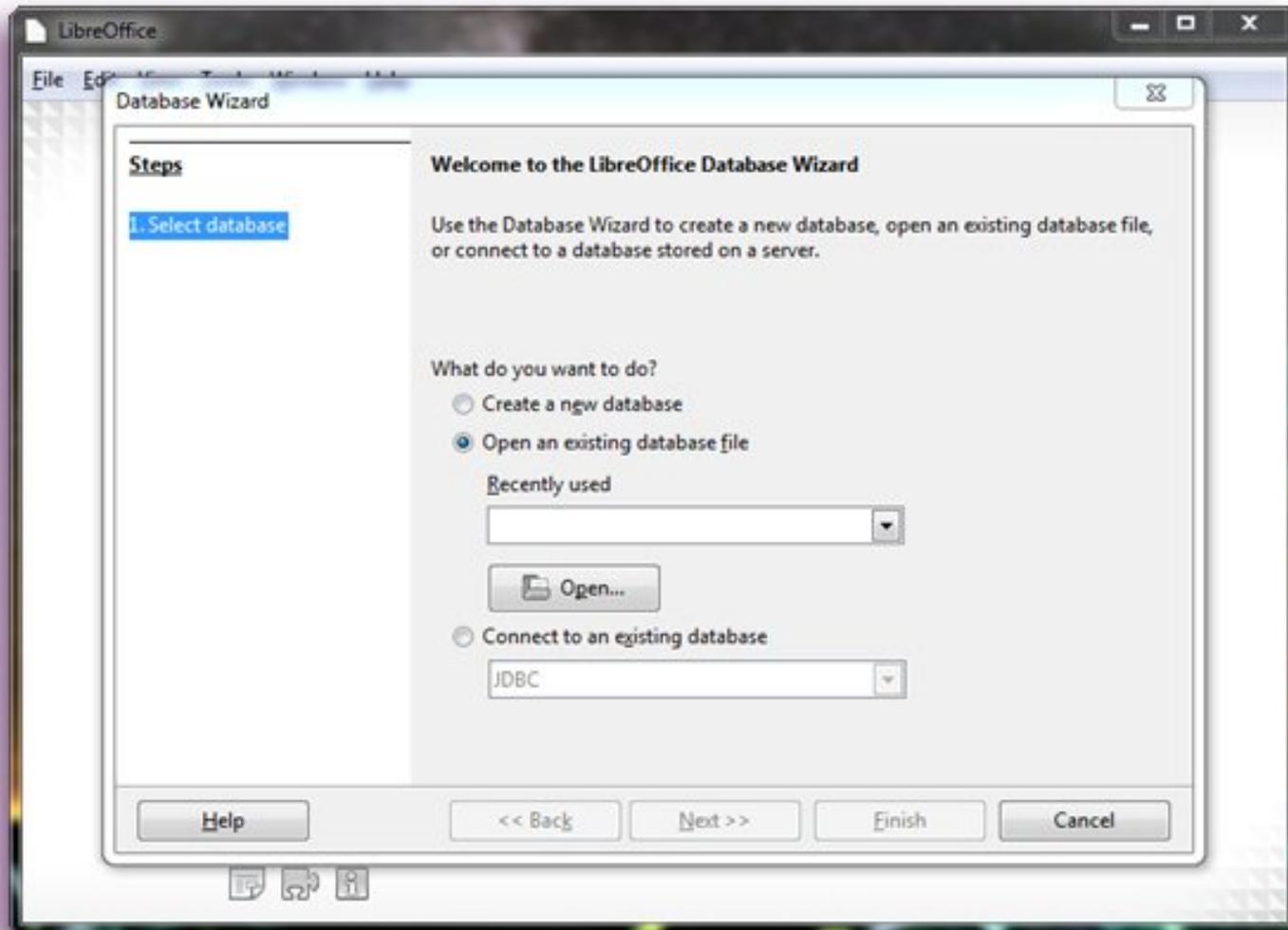
Chiave primaria: nome

Figura 12.4 La specifica della tabella **Balene**.

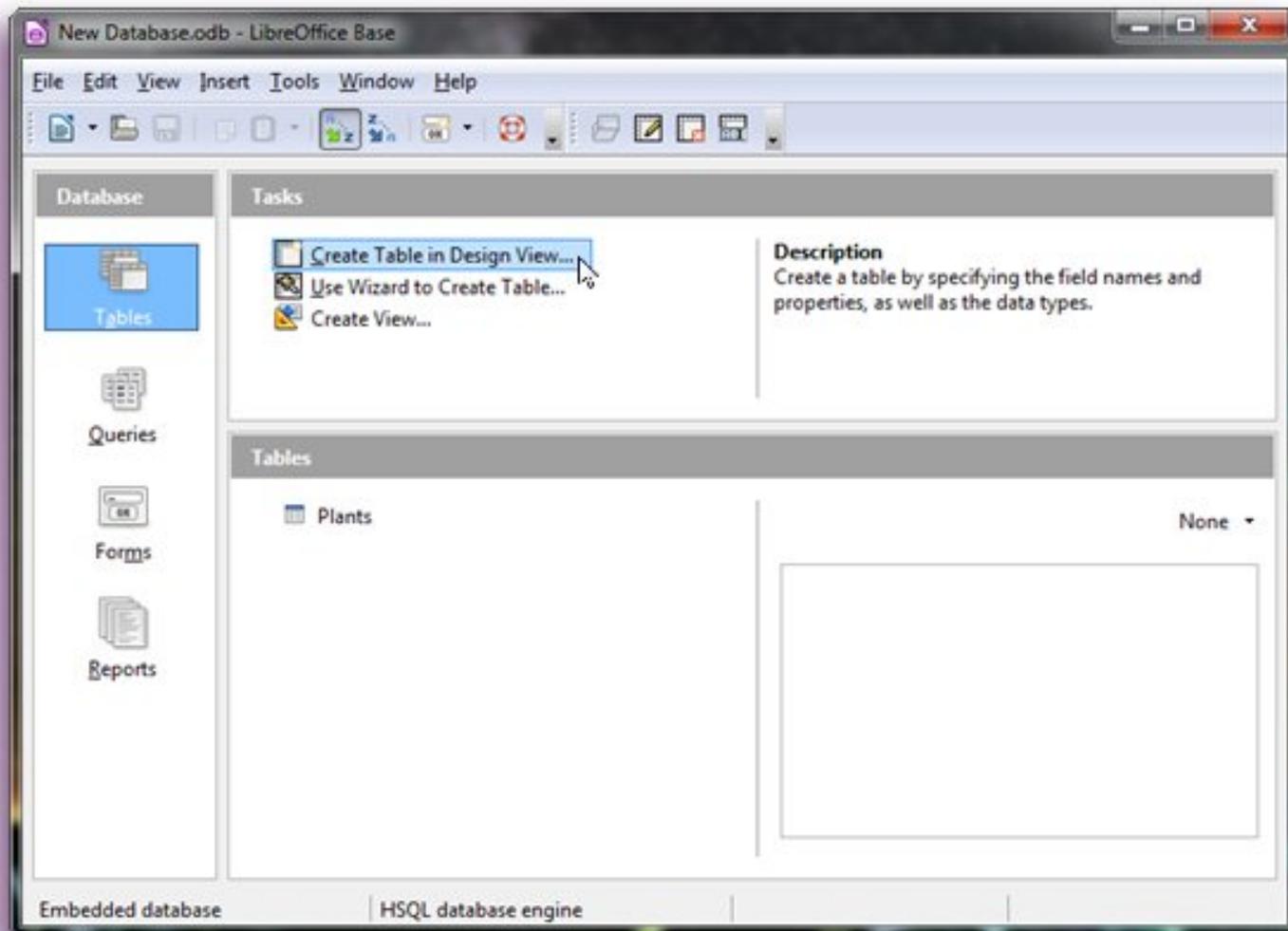
LibreOffice Base: creare un DB



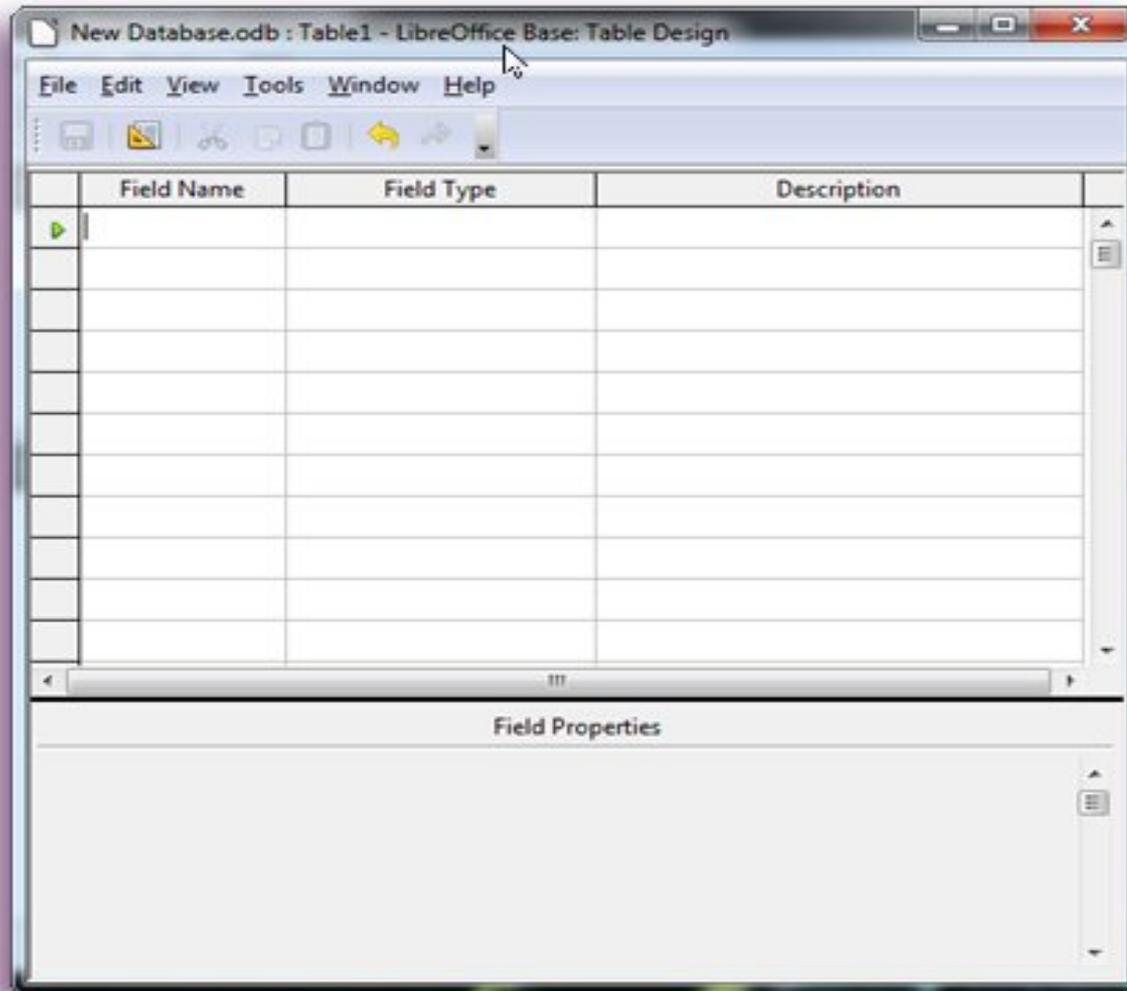
LibreOffice Base: creare un DB



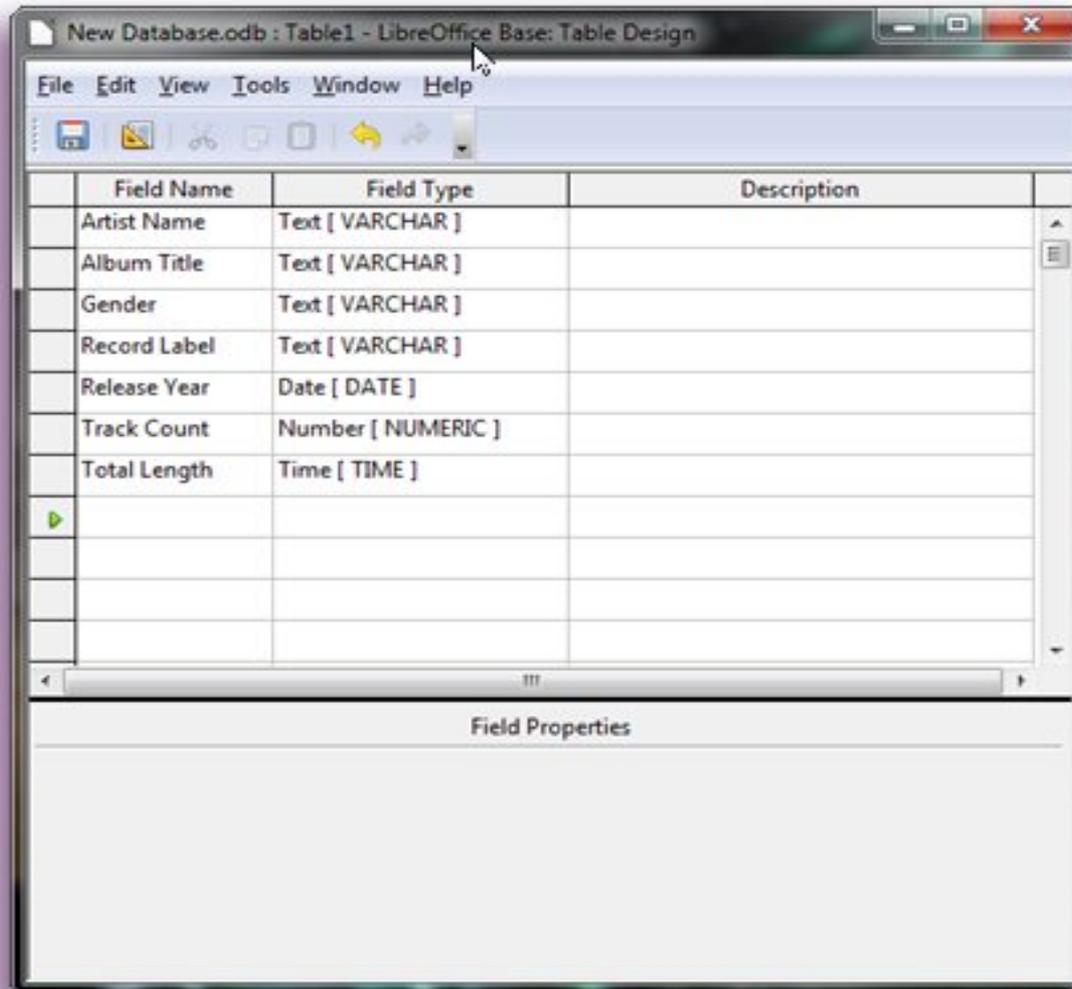
LibreOffice Base: Creare Tabella



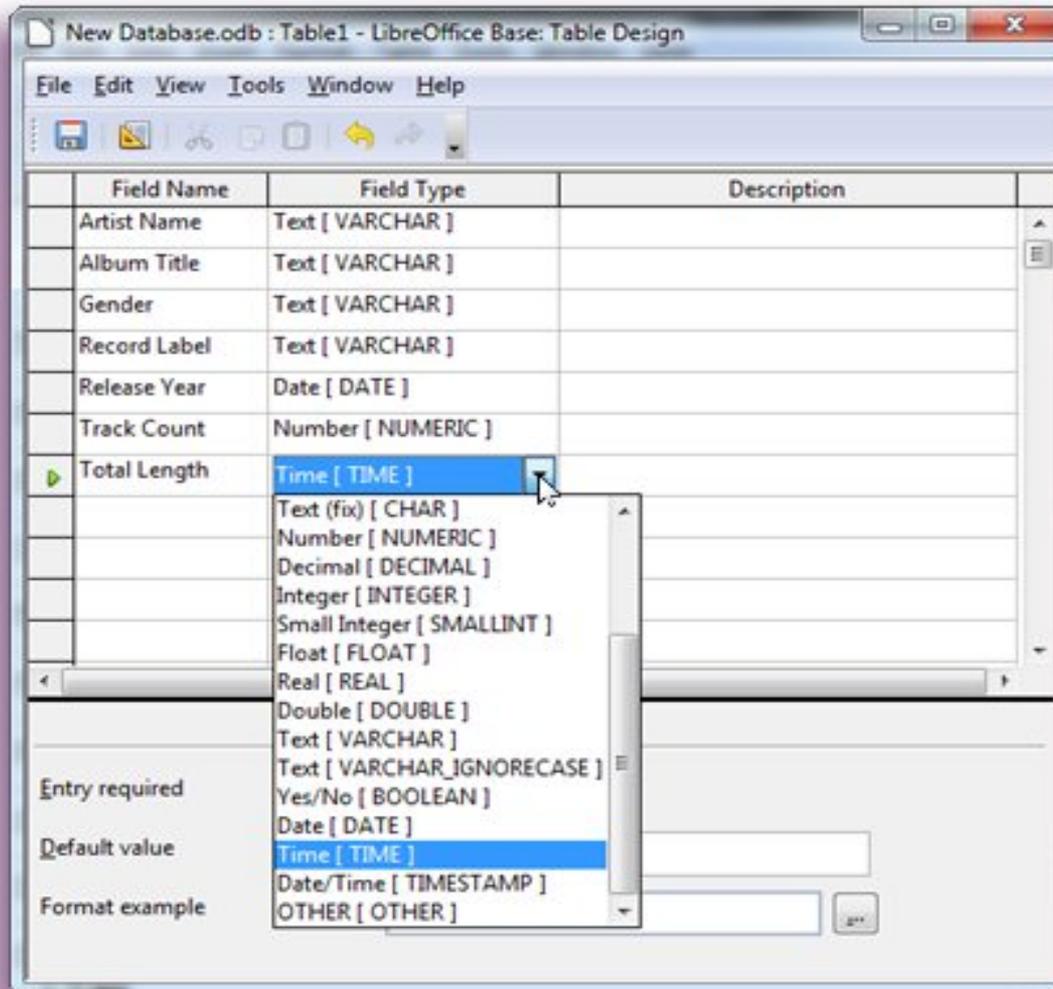
LibreOffice Base: Creare Tabella



LibreOffice Base



LibreOffice Base



LibreOffice Base: Form per creare istanze

The image shows a screenshot of a LibreOffice Base form titled "Fuel Purchases". The form is set against a dark blue background with a light blue cloud pattern. It contains several input fields and a dropdown menu. The fields are arranged in a grid-like fashion. The "FuelID" field contains the value "0". The "Date" field contains "Thursday, April 12, 2007" and has a small calendar icon to its right. The "FuelCost" field contains "\$26.45". The "FuelQuantity" field contains "11.650". The "Odometer" field contains "90997.9". The "PaymentType" field is a dropdown menu with "Kevin" selected. The form title "Fuel Purchases" is centered above the input fields.

FuelID	Date
0	Thursday, April 12, 2007
Fuel Purchases	
FuelCost	FuelQuantity
\$26.45	11.650
Odometer	PaymentType
90997.9	Kevin

Proprietà delle entità (cont.)

■ Unicità delle entità

- ❑ una tabella di database non può avere righe duplicate
- ❑ dev'esserci **almeno una caratteristica distintiva** per ogni entità (e.g., il codice fiscale)
- ❑ alcune caratteristiche possono essere uguali, ma non tutte

Proprietà delle entità (cont.)

■ Chiavi

- qualsiasi **insieme di attributi** per cui le entità sono tutte diverse si chiama *chiave candidata*, ne scegliamo una e la chiamiamo *chiave primaria*
- una chiave dev'essere in grado di **distinguere tutte le possibili entità**, non solo quelle presenti nel particolare stato corrente della tabella

Chiave Primaria (1)

- se nessuna combinazione di attributi si qualifica chiave candidata, occorre assegnare a ogni entità un **codice numerico distinto** (ID)
 - pensiamo ad esempio ai numeri di matricola assegnati agli studenti dall'università. Nella tabella Studenti l'insieme di attributi {Matricola} può essere la **chiave primaria**

Tabella Studenti



Matricola	Cognome	Nome	Sesso	DataDiNascita	LuogoDiNascita
300000	Rossi	Mario	M	01/01/1987	Torino
300001	Verdi	Chiara	F	01/02/1987	Roma
300002	Rossi	Daria	F	01/01/1987	Torino
300003	Cancelli	Chiara	F	10/10/1987	Milano
300004	Votantonio	Raffaella	F	10/02/1985	Roma

Chiave Primaria (2)

- L'insieme di attributi {Cognome, Nome} non è una buona chiave primaria
- se si iscrive un altro Mario Rossi dobbiamo **modificare** la **struttura** del DB

Tabella Studenti

Matricola	Cognome	Nome	Sesso	DataDiNascita	LuogoDiNascita
300000	Rossi	Mario	M	01/01/1987	Torino
300001	Verdi	Chiara	F	01/02/1987	Roma
300002	Rossi	Daria	F	01/01/1987	Torino
300003	Cancelli	Chiara	F	10/10/1987	Milano
300004	Votantonio	Raffaella	F	10/02/1985	Roma

Libreoffice Base: impostare la chiave primaria

	Field Name	Field Type	Description
	FuelID	Integer [INTEGER]	

Context menu options:

- Cut
- Copy
- Delete
- Insert Rows
- Primary Key

Field Properties

AutoValue: Yes

Auto-increment state: IDENTITY

Length: 10

Format example: 0

Operazioni sulle tabelle (1)

Nazioni

Nome	Character, 15	<i>Nome del paese</i>
Dominio	Character, 2	<i>Dominio Internet associato</i>
Capitale	Character, 20	<i>Capitale del paese</i>
Latitudine	Integer	<i>Latitudine della capitale</i>
N_S	Boolean	<i>Latitudine N(ord) o S(ud)</i>
Longitudine	Integer	<i>Longitudine della capitale</i>
E_W	Boolean	<i>Longitudine E (Est) o W (Ovest)</i>
Interesse	Character, 50	<i>Breve descrizione del paese</i>

Chiave primaria: Nome

<u>Nome</u>	<u>Dom</u>	<u>Capitale</u>	<u>Lat</u>	<u>NS</u>	<u>Lon</u>	<u>EW</u>	<u>Interesse</u>
Irlanda	IE	Dublino	52	N	7	W	Storico
Israele	IR	Gerusalemme	32	N	35	E	Storico
Italia	IT	Roma	42	N	12	E	Arte
Giamaica	JM	Kingston	18	N	77	W	Spiagge
Giappone	JP	Tokyo	35	N	143	E	Kabuki

Figura 12.6 La definizione della tabella **Nazioni** e alcuni elementi di esempio.

Operazioni sulle tabelle (2)

- L'uso principale dei database è la ricerca di informazioni
 - gli utenti specificano cosa vogliono sapere e il database lo trova
- le operazioni fondamentali applicabili alle tabelle sono cinque:
 - Selezione;
 - Proiezione;
 - Unione;
 - Differenza;
 - Prodotto.

Operazioni sulle tabelle - SELEZIONE

- l'operazione di **SELEZIONE** prende alcune **righe** di una tabella per crearne un'altra
 - si deve specificare la **tabella** da cui prelevare le righe e il **test** di selezione con la sintassi

Select Test From Tabella

- il **Test** è applicato a ogni riga per decidere se includerla o no nella tabella risultato. Il **Test** usa nomi degli attributi, costanti numeriche e operatori relazionali (=, ≠, <, >, ≤, ≥);
- se il **Test** è verificato, la riga è aggiunta nella tabella risultato, altrimenti è ignorata

Operazioni sulle tabelle - SELEZIONE

Select Interesse = “Spiagge”
From Nazioni

Nome	Dom	Capitale	Lat	NS	Lon	EW	Interesse
Australia	AU	Canberra	37	S	148	E	Spiagge
Bahamas	BS	Nassau	25	N	78	W	Spiagge
Barbados	BB	Bridgetown	13	N	59	W	Spiagge
Belize	BZ	Belize	17	N	89	W	Spiagge
Bermuda	BM	Hamilton	32	N	64	W	Spiagge

Figura 12.7 Parte della tabella creata selezionando nazioni con un test che impone che Interesse sia uguale a Spiagge.

Operazioni sulle tabelle - PROIEZIONE

- L'operazione di **PROIEZIONE** prende alcune **colonne** di una tabella esistente e ne crea un'altra
 - si deve specificare il nome della tabella esistente e le colonne (la **lista di attributi**) da includere nella nuova tabella con la sintassi:

Project Lista Attributi From Tabella

- la nuova tabella avrà il tante colonne quante definite nella lista di attributi e lo stesso numero di righe di quella originale.
- **Attenzione**: se la nuova tabella elimina un campo chiave eventuali righe duplicate nella nuova tabella **saranno eliminate**

Operazioni sulle tabelle - PROIEZIONE

Project (Nome, Dominio, Interesse)
From Nazioni

<u>Nome</u>	<u>Dom</u>	<u>Interesse</u>
Nauru	NR	Spiagge
Nepal	NP	Montagne
Olanda	NL	Arte
Nuova Caledonia	NC	Spiagge
Nuova Zelanda	NZ	Avventura

Figura 12.8 Un frammento del risultato di un'operazione di proiezione sulle **Nazioni**.

Operazioni sulle tabelle – UNIONE, DIFFERENZA E PRODOTTO

- L'**UNIONE** combina due tabelle che hanno gli stessi attributi. Sintassi:

Tabella1 + Tabella2

- La **DIFFERENZA** rimuove da una tabella tutte le righe contenute in una seconda tabella. Sintassi:

Tabella1 – Tabella2

- Il **PRODOTTO** crea una super-tabella che ha tutti gli attributi di delle tabelle originali combinando tutte le righe della prima tabella con tutte quelle della seconda. Sintassi:

Tabella1 x Tabella2

L'operazione di JOIN (1)

- La **JOIN** (\bowtie) combina due tabelle come il prodotto, ma non produce necessariamente tutte le combinazioni di righe.
- Se le tabelle hanno un campo che contiene lo stesso tipo di dato, la nuova tabella **concatena solo le righe in cui c'è una corrispondenza** su quel campo.
- Sintassi: **Tabella1 JOIN Tabella2 ON corrispondenza**

L'operazione di JOIN (2)

- La **corrispondenza** è un confronto che considera un particolare attributo di ogni tabella
- Il sistema verifica se i campi corrispondenti nelle due tabelle contengono lo stesso dato; in tal caso viene creata una riga nella tabella risultato

L'operazione di JOIN (3)

<u>Nome</u>	<u>Dom</u>	<u>Lat</u>
Finlandia	FI	64
Groenlandia	GL	72
Islanda	IS	65
Norvegia	NO	62

Figura 12.9 **Northland**, la tabella delle nazioni che hanno le capitali più a nord.

<u>Nome</u>	<u>Amico</u>
Ciad	Wen
Cile	Isabella
Cina	Wen
Isole Christmas	Clare
Isole Cocos	Brian

Figura 12.13 Un frammento della tabella **Master**, con le responsabilità di tutoring assegnate ai diversi amici.

L'operazione di JOIN (4)

Master

Nome	Amico
Afghanistan	Wen
...	...
Finlandia	Wen
...	...
Groenlandia	Clare
...	..
Islanda	Clare
...	...
Norvegia	Wen
...	..
Zimbabwe	Brian

Master JOIN Northland

ON Master.Nome = Northland.Nome

Northland

Nome	Dom	Lat
Finlandia	FI	64
Groenlandia	GL	72
Islanda	IS	65
Norvegia	NO	62

Nome	Amico	Dom	Lat
Finlandia	Wen	FI	64
Groenlandia	Clare	GL	72
Islanda	Clare	IS	65
Norvegia	Wen	NO	62

Risultato

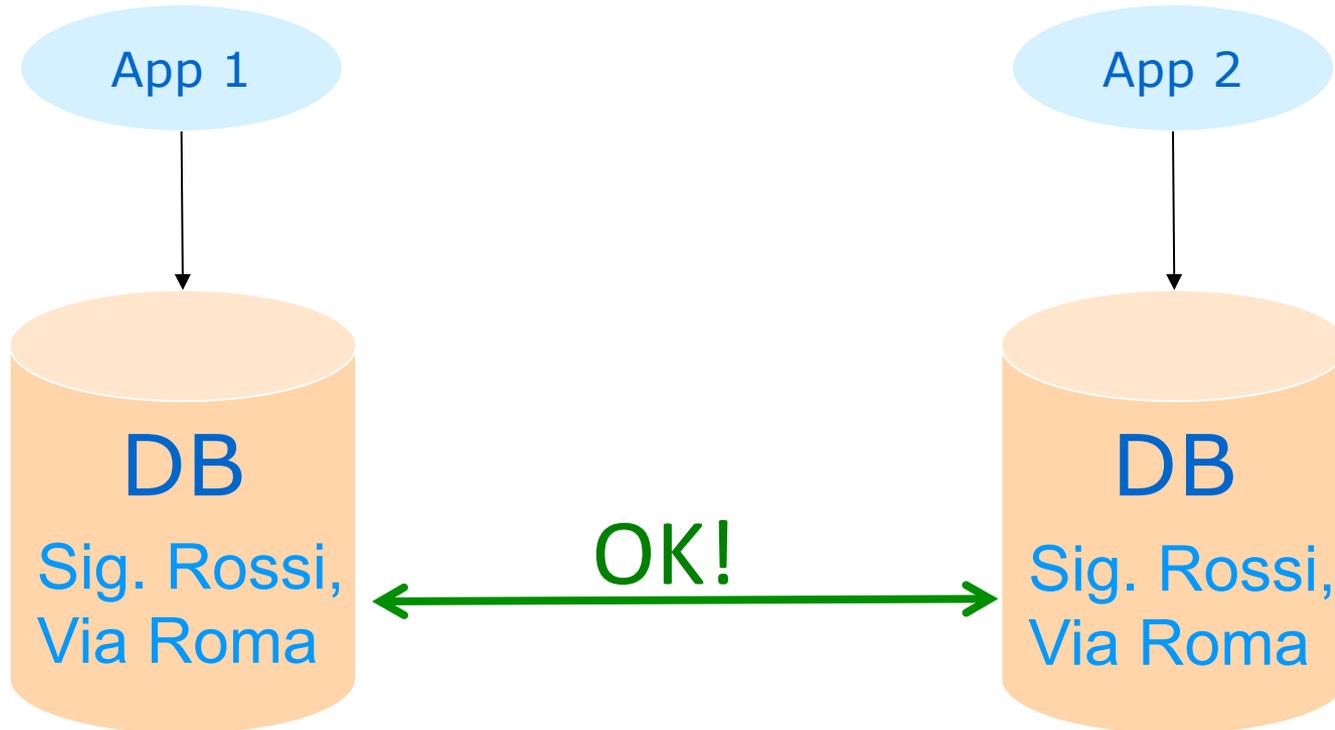
Il DataBase Fisico

Il database fisico (1)

- Ogni dato dovrebbe essere memorizzato univocamente nel DB.
- Più copie dello stesso dato richiedono una costante sincronizzazione per preservarne la coerenza.
- I dati incoerenti vengono chiamati “*garbage*” e la loro presenza è peggiore della mancanza del dato stesso

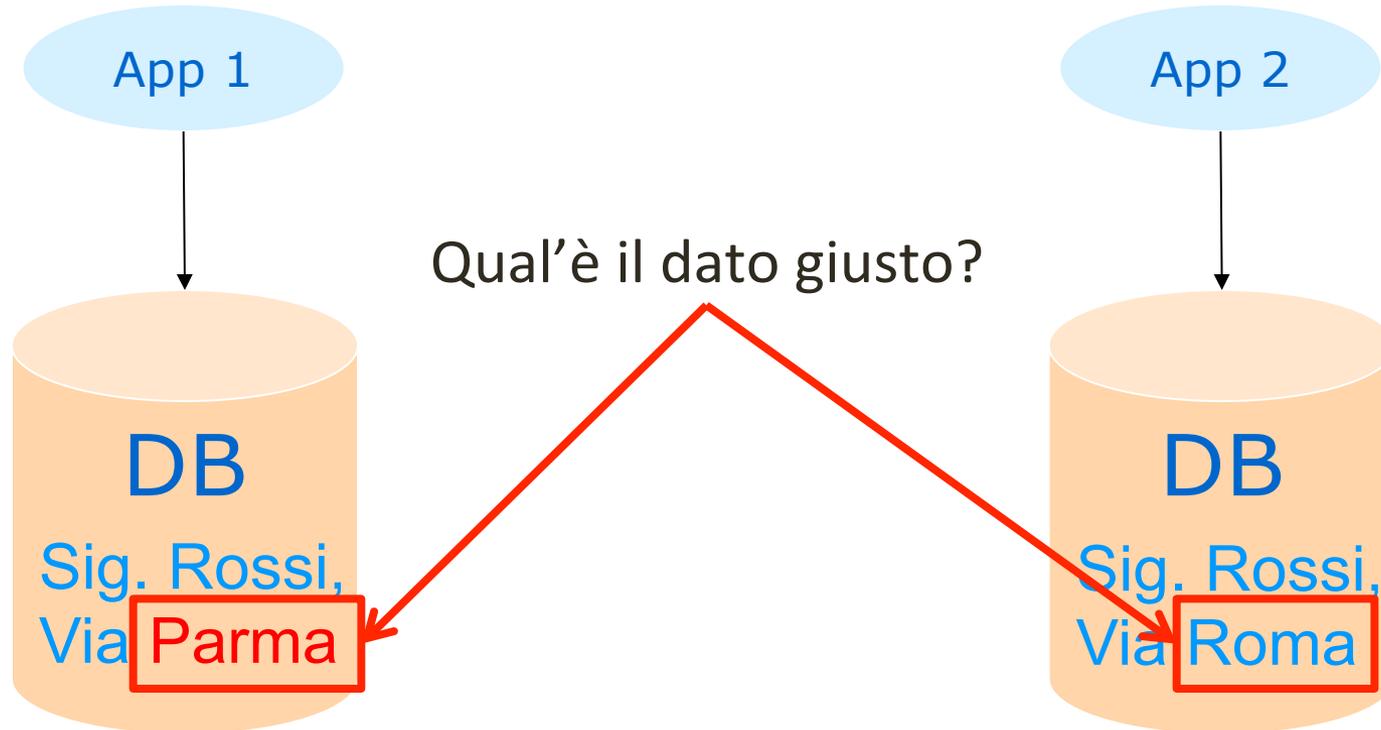
Ridondanza (1)

- ridondanza **indesiderabile**



Ridondanza (2)

- ridondanza **indesiderabile**

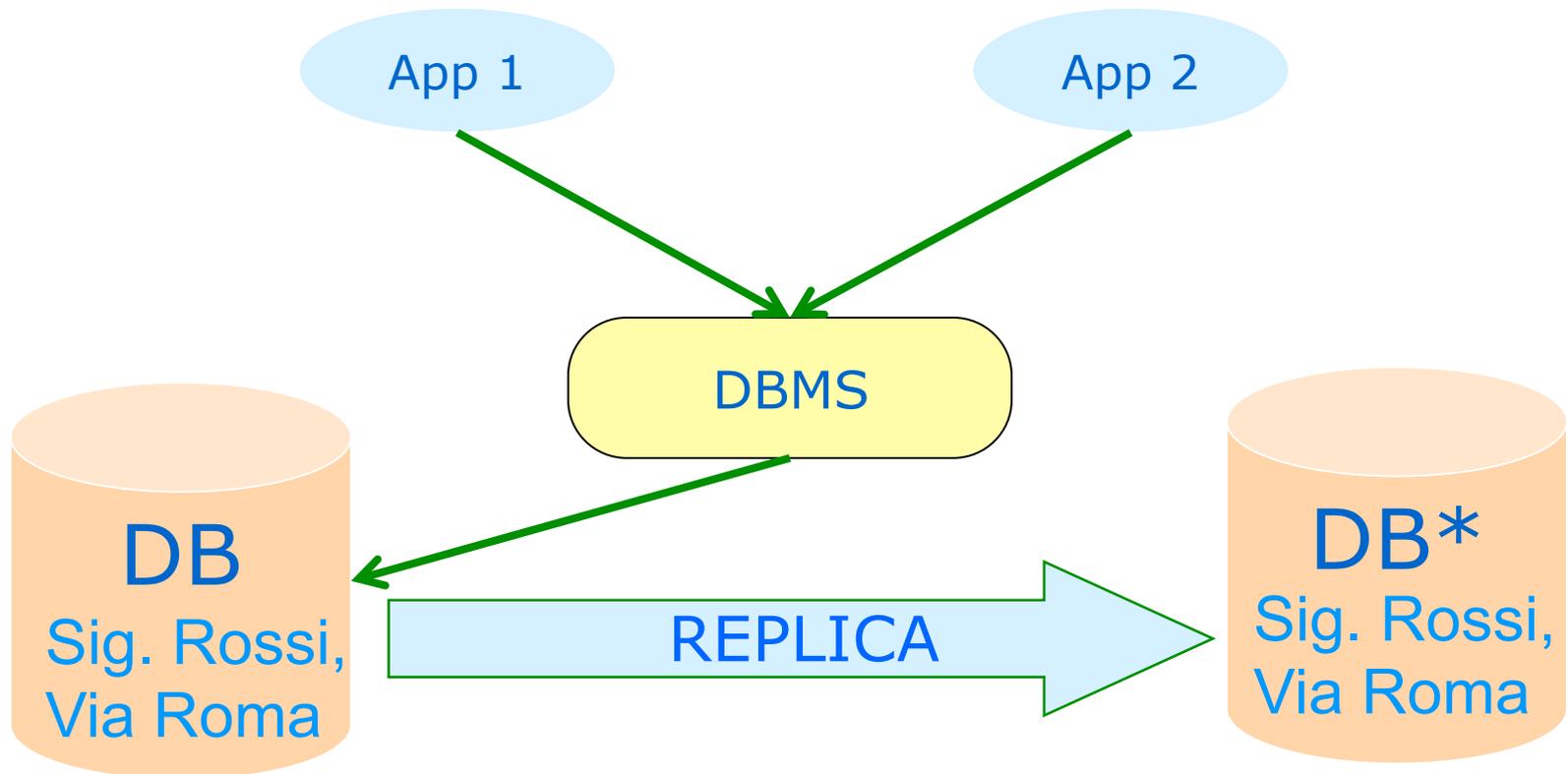


Il database fisico (2)

- Una sola copia di ogni dato
 - evitare le duplicazioni favorisce la coerenza interna, ma non assicura che i dati siano corretti
- Se l'informazione è necessaria in più parti del DB
 - è meglio tenere una lista singola e permettere agli altri di accedervi

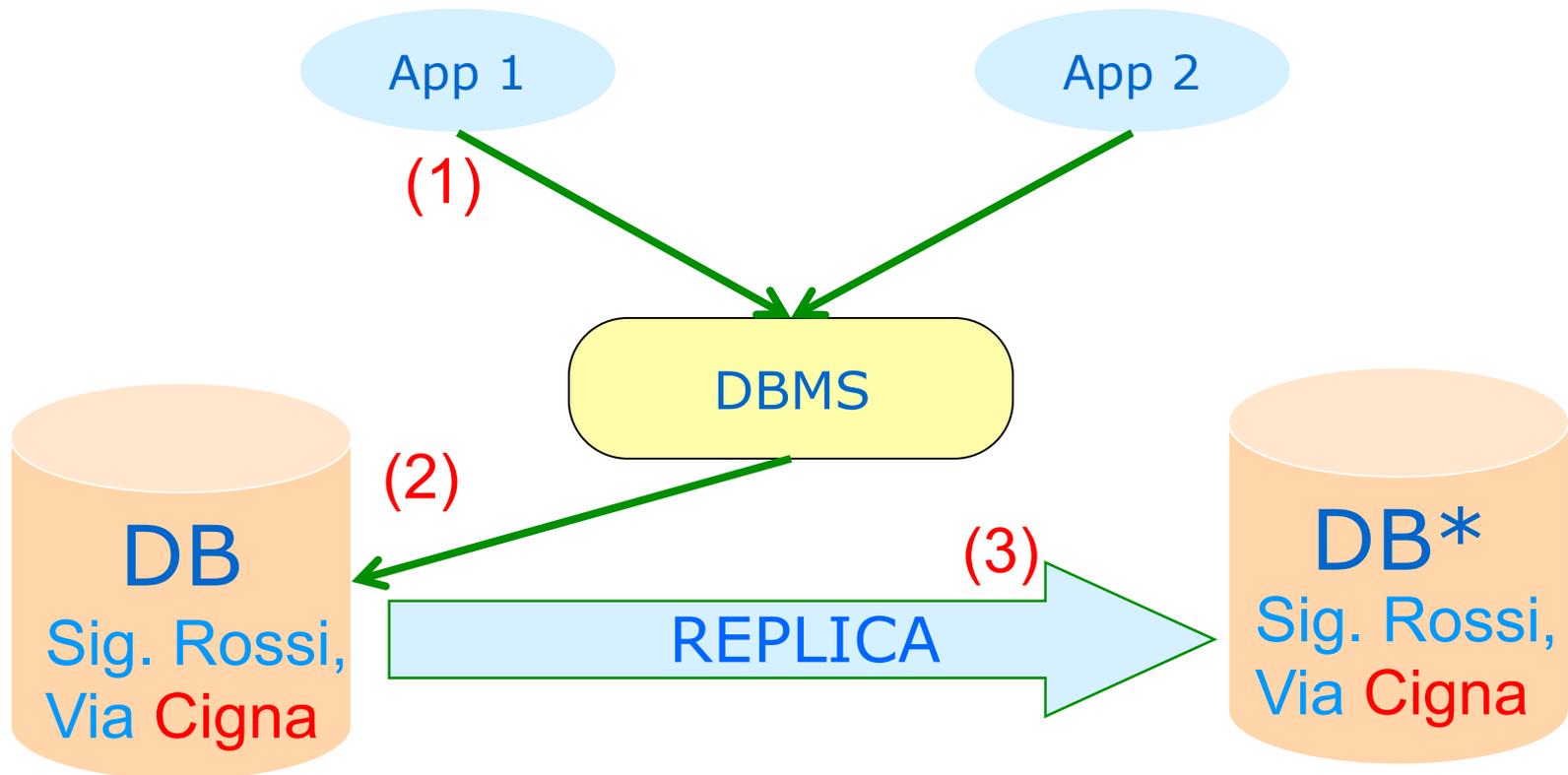
Ridondanza (3)

- ridondanza **utile** per **affidabilità**



Ridondanza (4)

- ridondanza **utile** per **affidabilità**



Ridondanza (5)

- la condivisione di un DB da parte di molte applicazioni **riduce** drasticamente i problemi legati alla ridondanza
- In generale, è desiderabile eliminare (minimizzare) la ridondanza anche all'interno di un singolo DB

Ridondanza (6)

- Riprendiamo la tabella dei libri e supponiamo di avere anche delle informazioni sull'autore, come nazionalità e anno di nascita

Titolo	Genere	Pagine	Prezzo	Autore	Nazionalità	Nascita
Dieci piccoli indiani	Giallo	530	29,90	A. Christie	Britannica	1890
Io non ho paura	Romanzo	219	9,50	N. Ammanniti	Italiana	1966
Harry Potter e la pietra filosofale	Fantasy	293	16,80	J. K. Rowling	Britannica	1965
Harry Potter e il prigioniero di Azkaban	Fantasy	366	14,50	J. K. Rowling	Britannica	1965

- Inaccettabile **ridondanza!**

Ridondanza (7)

- Soluzione: creare una tabella degli autori

Tabella libri

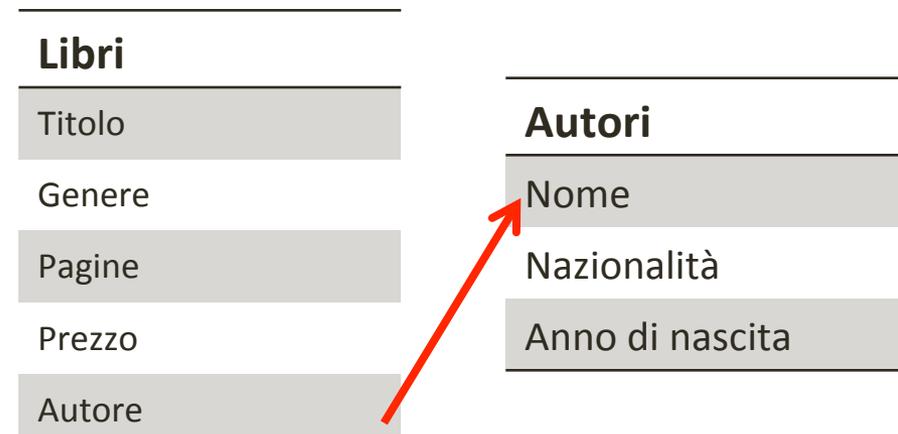
Titolo	Genere	Pagine	Prezzo	Autore
Dieci piccoli indiani	Giallo	530	29,90	A. Christie
Io non ho paura	Romanzo	219	9,50	N. Ammanniti
Harry Potter e la pietra filosofale	Fantasy	293	16,80	J. K. Rowling
Harry Potter e il prigioniero di Azkaban	Fantasy	366	14,50	J. K. Rowling

Tabella autori

Nome	Nazionalità	Nascita
A. Christie	Britannica	1980
N. Ammanniti	Italiana	1966
J. K. Rowling	Britannica	1965

Schema logico

- Dal punto di vista logico si crea una **relazione** tra la tabella libri e quella autori: l'attributo autore della tabella libri si riferisce all'attributo nome della tabella autori
- Nomi delle tabelle, attributi e loro relazioni si possono rappresentare graficamente tramite uno **schema logico**.
- Quando in una base di dati non ci sono ridondanze, si dice che è **normalizzata**.



Ridondanza (8)

- una tabella con dati ridondanti

Tabella - Esami

CodiceCorso	Voto	Data	Matricola	Cognome	Nome	Sesso	DataDiNascita	LuogoDiNascita
L0155	25	15/01/2007	300001	Verdi	Chiara	F	01/02/1987	Roma
L0507	30	01/06/2006	300001	Verdi	Chiara	F	01/02/1987	Roma
L0014	24	20/02/2007	300002	Rossi	Daria	F	01/01/1987	Torino
L0014	28	10/02/2007	300003	Cancelli	Chiara	F	10/10/1987	Milano
L0507	21	10/09/2006	300003	Cancelli	Chiara	F	10/10/1987	Milano
L0016	27	10/09/2006	300004	Votantonio	Raffaella	F	10/02/1985	Roma

Ridondanza (9)

- possiamo distribuire i dati su **due** tabelle

Matricola	Cognome	Nome	Sesso	DataDiNascita	LuogoDiNascita
300000	Rossi	Mario	M	01/01/1987	Torino
300001	Verdi	Chiara	F	01/02/1987	Roma
300002	Rossi	Daria	F	01/01/1987	Torino
300003	Cancelli	Chiara	F	10/10/1987	Milano
300004	Votantonio	Raffaella	F	10/02/1985	Roma

Tabella - Studenti

Tabella - Esami

CodiceCorso	Voto	Data	Matricola
L0014	24	20/02/2007	300002
L0014	28	10/02/2007	300003
L0016	27	10/09/2006	300004
L0155	25	15/01/2007	300001
L0507	30	01/06/2006	300001
L0507	21	10/09/2006	300003

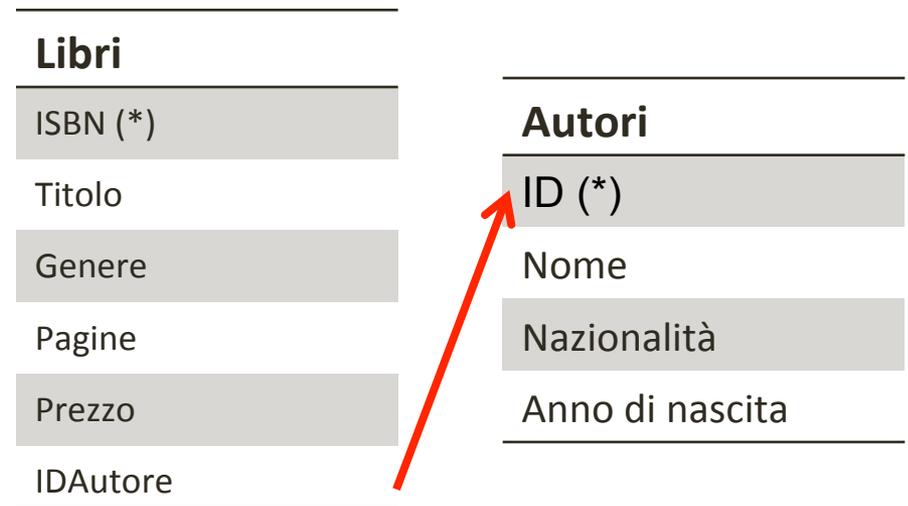
Chiavi esterne (1)

- Il legame tra le tabelle Studenti ed Esami si esprime con un **vincolo di integrità referenziale**
- L'attributo Matricola della tabella Esami è detto **chiave esterna** di Esami su Studenti
- L'integrità referenziale richiede che ogni valore della chiave esterna Matricola di Esami **corrisponda a un valore della chiave (primaria) Matricola di Studenti**

Chiavi esterne (2)

- I campi di una tabella che si riferiscono alle chiavi primarie di un'altra tabella prendono il nome di **chiavi esterne**.

- Esempio libri - autori

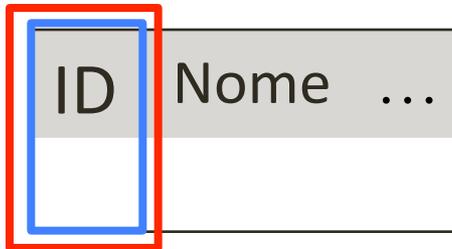


- I campi con l'asterisco (*) sono le chiavi primarie.
- IDAutore è una chiave esterna che si riferisce alla tabella Autori.

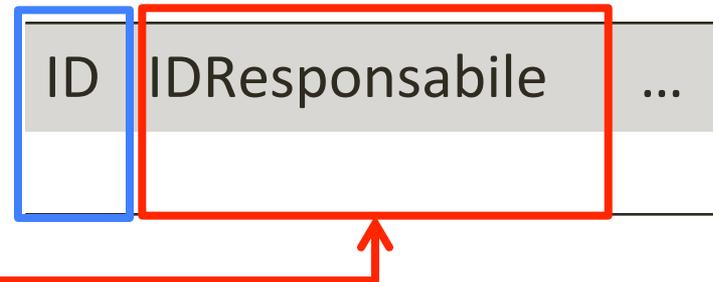
Chiavi Esterne (3)

- Il **nome** di un attributo **chiave esterna** può essere **diverso** dal **nome** dell'attributo **chiave (primaria)** cui si riferisce

Impiegati



Progetti



Chiavi Esterne (4)

- una chiave esterna può essere **composta** da **più attributi**

Stazioni

Città	Nome	...
-------	------	-----

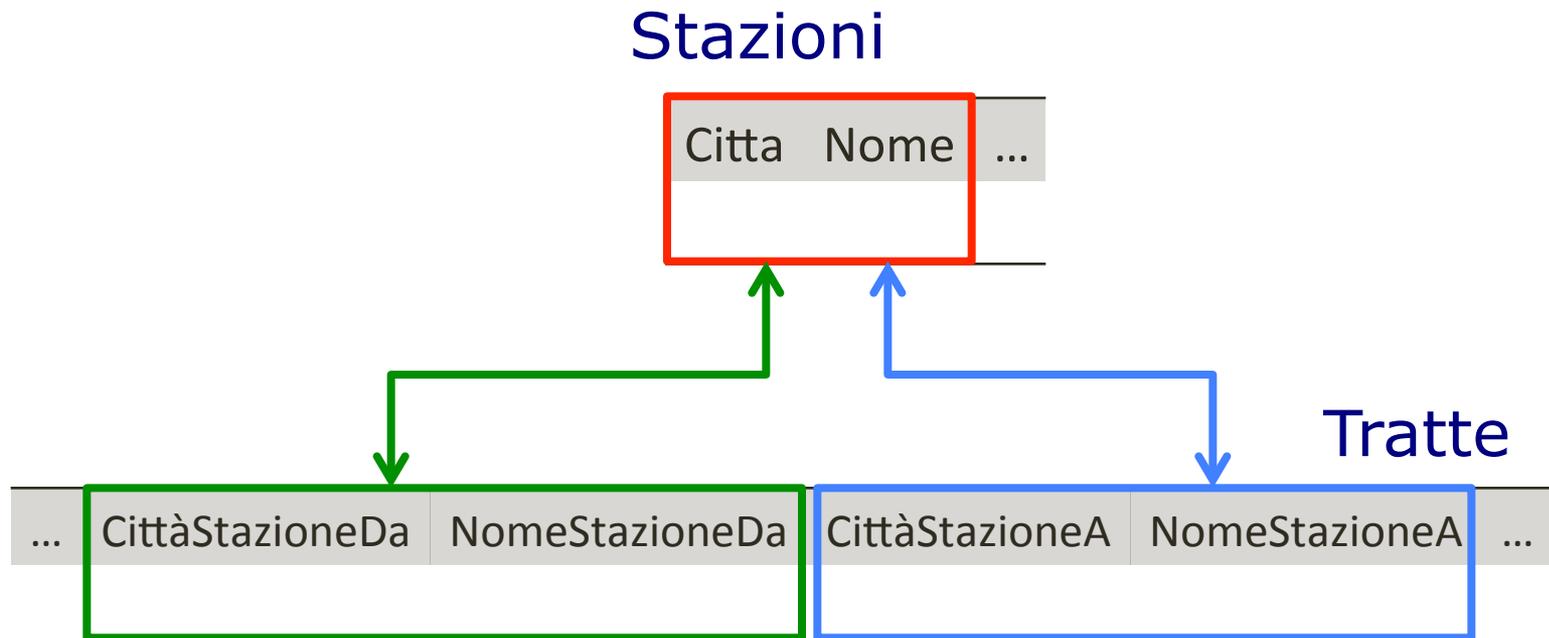
OrarioFermate

...	CittàStazione	NomeStazione	...
-----	---------------	--------------	-----



Chiavi Esterne (5)

- Una tabella può definire più chiavi esterne

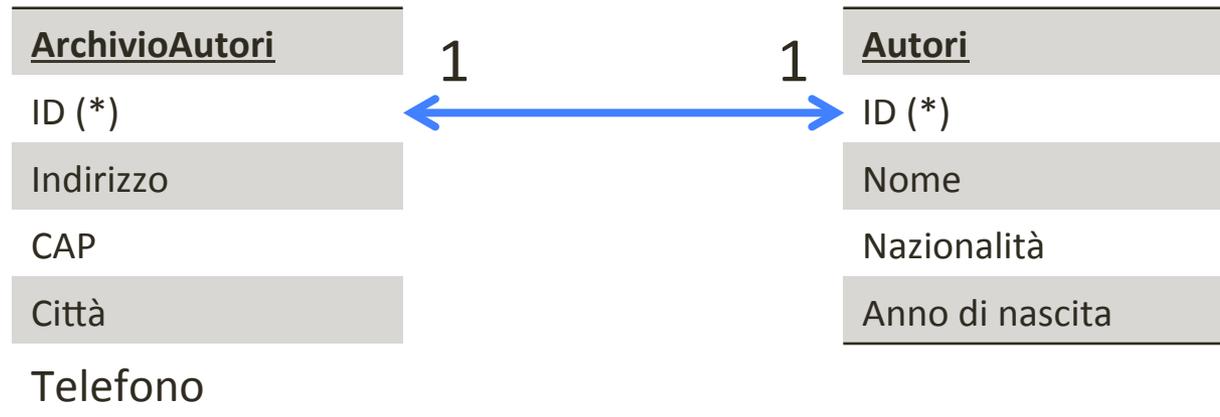


Tipi di relazioni

- Chiavi esterne e chiavi primarie creano delle relazioni
- Ci sono vari tipi di relazione
 - Relazioni **uno ad uno** (1-1)
 - Relazioni **uno a molti** (1-n)
 - Relazioni **molti a molti** (n-n)
- Alcune si possono realizzare direttamente in un database, altre necessitano di **tabelle ausiliarie**.
- Nello schema logico si possono annotare le linee tra le tabelle per chiarire di che tipo di relazione si tratti.

Relazioni uno ad uno (1-1)

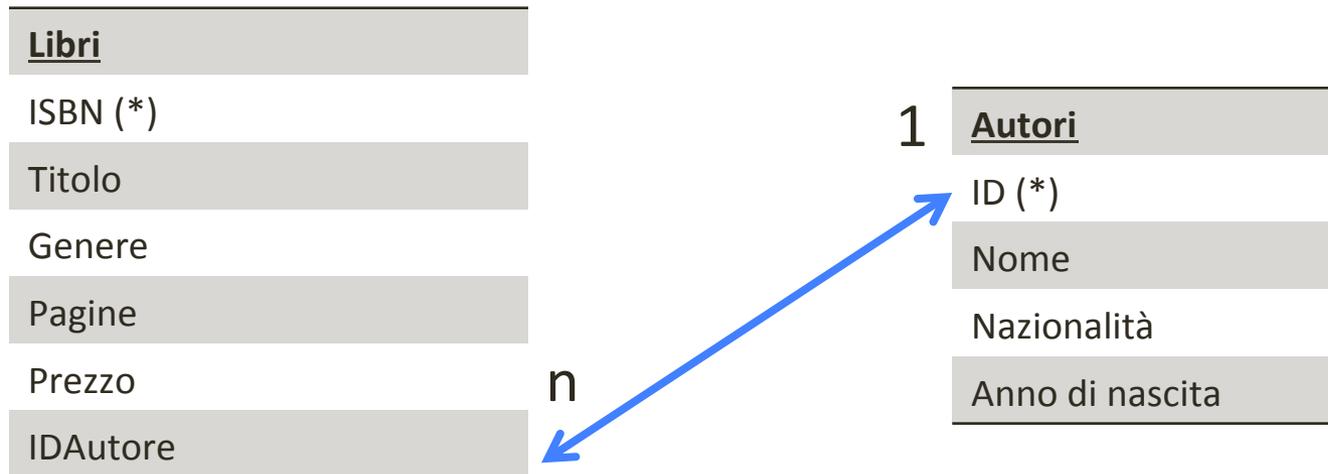
- Ad ogni riga di una tabella corrisponde una sola riga dell'altra e viceversa
 - Esempio: se vogliamo memorizzare dei dati privati degli autori, per dividerli meglio da quelli pubblici possiamo usare una sola tabella.



- Non molto comune.
- La chiave esterna coincide con la chiave primaria.

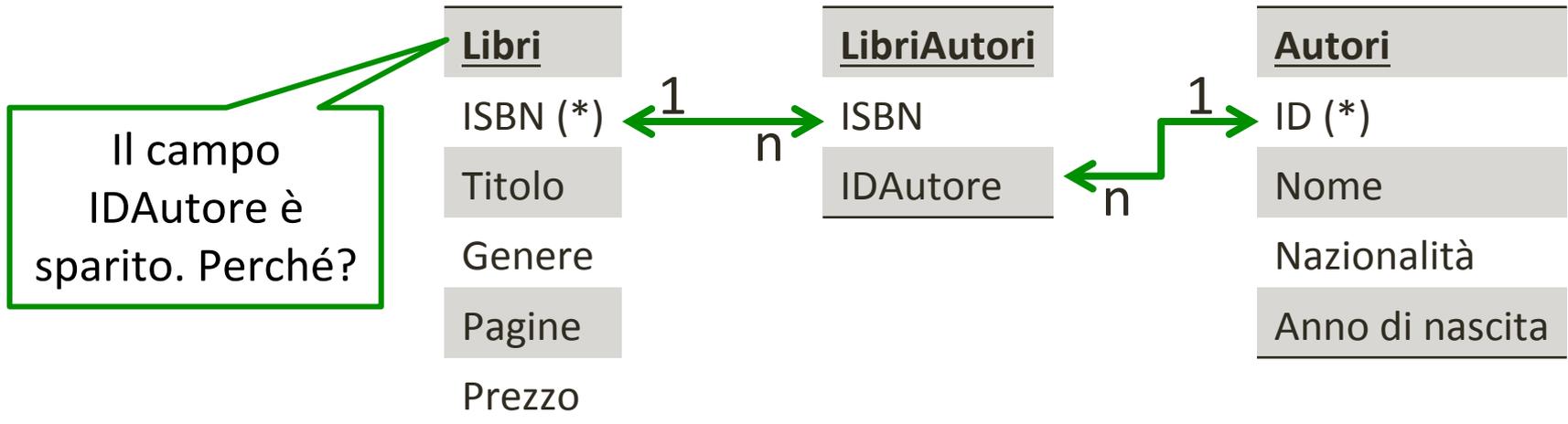
Relazioni uno a molti

- Ad ogni riga della prima tabella corrisponde una riga della seconda tabella, ma nella direzione opposta, ad ogni riga della seconda corrispondono più righe della prima.
 - Esempio: la relazione tra autori e libri che abbiamo visto prima. Ad ogni libro corrisponde un autore (ovviamente stiamo semplificando la realtà), ma ad ogni autore corrispondono molti libri.



Relazioni molti a molti (1)

- Ad ogni riga della prima tabella corrispondono più righe della seconda tabella e viceversa.
 - Non si può realizzare direttamente, serve una tabella ausiliaria.
 - Esempio: nel mondo reale, un libro può essere scritto da più autori.



Relazioni molti a molti (2)

■ Chi ha scritto il “Libro inesistente” ?

Autori

Nome	Nazionalità	Nascita	ID
J. K. Rowling	Britannica	1965	1
A. Christie	Britannica	1980	2
N. Ammanniti	Italiana	1966	3
J. K. Rowling	Statunitense	1983	4

LibriAutori

IDAutore	ISBN
1	3858-38-4923
2	x1
1	x2
3	x3
4	x3

Libri

Titolo	Prezzo	ISBN
Harry Potter e la pietra filosofale	16,80	3858-38-4923
Dieci piccoli indiani	29,90	x1
Harry Potter e il prigioniero di Azkaban	14,50	x2
Libro inesistente	0,4	x3

Relazioni molti a molti (3)

■ Chi ha scritto il “Libro inesistente” ?

Autori

Nome	Nazionalità	Nascita	ID
J. K. Rowling	Britannica	1965	1
A. Christie	Britannica	1980	2
N. Ammanniti	Italiana	1966	3
J. K. Rowling	Statunitense	1983	4

LibriAutori

IDAutore	ISBN
1	3858-38-4923
2	x1
1	x2
3	x3
4	x3

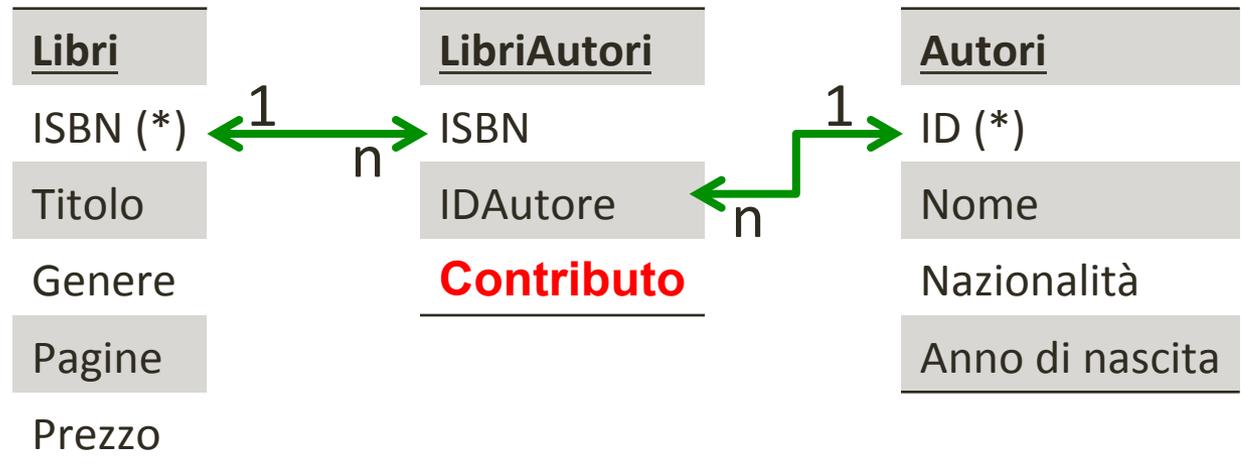
Libri

Titolo	Prezzo	ISBN
Harry Potter e la pietra filosofale	16,80	3858-38-4923
Dieci piccoli indiani	29,90	x1
Harry Potter e il prigioniero di Azkaban	14,50	x2
Libro inesistente	0,4	x3

Dati aggiuntivi di una relazione (1)

- Nel caso di relazioni molti a molti, è possibile **specificare dei dati aggiuntivi** nella tabella ausiliaria.
 - Utile se ci sono informazioni inerenti la relazione tra le due tabelle principali.
 - Esempio: supponiamo di voler memorizzare il **contributo** di ogni autore ad ogni libro che ha scritto.

- Non ha senso mettere questa informazione né nella tabella autori, né in quella libri.



Dati aggiuntivi di una relazione (2)

Autori

Nome	Nazionalità	Nascita	ID
J. K. Rowling	Britannica	1965	1
A. Christie	Britannica	1980	2
N. Ammanniti	Italiana	1966	3
J. K. Rowling	Statunitense	1983	4

LibriAutori

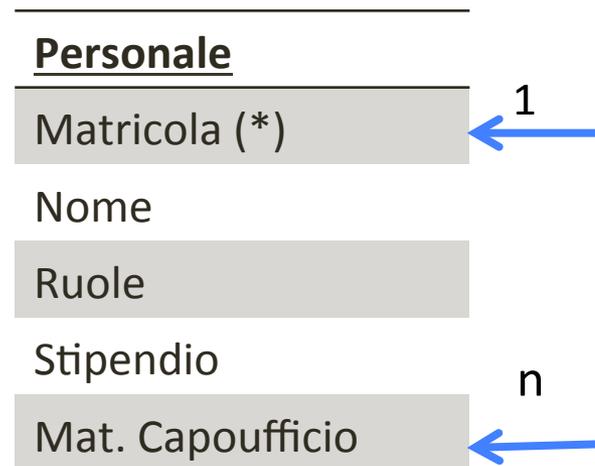
IDAutore	ISBN	Contributo
1	3858-38-4923	Tutto
2	x1	Tutto
1	x2	Tutto
3	x3	Cap. 1-5
4	x3	Cap. 6-12

Libri

Titolo	Prezzo	ISBN
Harry Potter e la pietra filosofale	16,80	3858-38-4923
Dieci piccoli indiani	29,90	x1
Harry Potter e il prigioniero di Azkaban	14,50	x2
Libro inesistente	0,4	x3

Relazioni gerarchiche

- Talvolta una chiave esterna fa riferimento alla sua stessa tabella.
 - utile per rappresentare relazione gerarchiche (capo di..., progetto derivato da... , etc..)
 - Esempio: tabella con il personale di una azienda



Integrità Referenziale

Relazioni e integrità

- In una coppia chiave esterna – chiave primaria, tutti i valori per la chiave esterna dovrebbero corrispondere a valori validi per la chiave primaria.

ID	Nome	Nazionalità	Nascita
1	J. K. Rowling	Britannica	1965
2	A. Christie	Britannica	1980
3	N. Ammanniti	Italiana	1966
4	J. K. Rowling	Statunitense	1983

Chi è l'autore di questo libro?

ISBN	Titolo	Prezzo	IDAutore
3858-38-4923	Harry Potter e la pietra filosofale	80	1
xxxxx	Dieci piccoli indiani	29,9	2
xxxxx	Harry Potter e il prigioniero di Azkaban	14,50	1
xxxxx	Io non ho paura	9,50	5

Integrità referenziale

- Per **integrità referenziale** si intende la proprietà delle base di dati “buone” per cui ogni valore per una chiave esterna corrisponde ad un valore della relativa chiave primaria.
- Il DBMS tenta di mantenere l’integrità referenziale evitando di effettuare operazioni che la violino.
 - Per far ciò il DBMS deve conoscere quali sono le coppie chiavi esterne – chiavi private.
 - Ogni DBMS ha degli strumenti che servono ad istruirlo sulle relazioni presenti nel database.

Cancellazione

- Cosa fare se si vuole cancellare l'autore 4?

ID	Nome	Nazionalità	Nascita
1	J. K. Rowling	Britannica	1965
2	A. Christie	Britannica	1980
3	N. Ammanniti	Italiana	1966
4	J. K. Rowling	Statunitense	1965

ISBN	Titolo	Prezzo	IDAutore
3858-38-4923	Harry Potter e la pietra filosofale	16,80	1
xxxxx	Dieci piccoli indiani	29,90	2
xxxxx	Harry Potter e il prigioniero di Azkaban	14,50	1
xxxxx	Io non ho paura	9,50	4



Cancellazione

- 1^a possibilità: annullare l'operazione

ID	Nome	Nazionalità	Nascita
1	J. K. Rowling	Britannica	1965
2	A. Christie	Britannica	1980
3	N. Ammanniti	Italiana	1966
4	J. K. Rowling	Statunitense	1983

ISBN	Titolo	Prezzo	IDAutore
3858-38-4923	Harry Potter e la pietra filosofale	16,80	1
xxxxx	Dieci piccoli indiani	29,90	2
xxxxx	Harry Potter e il prigioniero di Azkaban	14,50	1
xxxxx	Io non ho paura	9,50	4

Cancellazione

- 2[^] possibilità: **cancellazione a cascata**

ID	Nome	Nazionalità	Nascita
1	J. K. Rowling	Britannica	1965
2	A. Christie	Britannica	1980
3	N. Ammanniti	Italiana	1966
4	J. K. Rowling	Statunitense	1983

ISBN	Titolo	Prezzo	IDAutore
3858-38-4923	Harry Potter e la pietra filosofale	16,80	1
xxxxx	Dieci piccoli indiani	29,90	2
xxxxx	Harry Potter e il prigioniero di Azkaban	14,50	1
xxxxx	Le non ho paura	9,50	1

Cancellazione

- 3^a soluzione: impostare a NULL i corrispondenti valori delle chiavi esterne (se il valore NULL è ammesso)

ID	Nome	Nazionalità	Nascita
1	J. K. Rowling	Britannica	1965
2	A. Christie	Britannica	1980
3	N. Ammanniti	Italiana	1966
4	J. K. Rowling	Statunitense	1965

ISBN	Titolo	Prezzo	IDAutore
3858-38-4923	Harry Potter e la pietra filosofale	16,80	1
xxxxx	Dieci piccoli indiani	29,90	2
xxxxx	Harry Potter e il prigioniero di Azkaban	14,50	1
xxxxx	Io non ho paura	9,50	NULL

Cancellazione

- Notare che non c'è problema a cancellare una riga dal lato “molti” della relazione.

ID	Nome	Nazionalità	Nascita
1	J. K. Rowling	Britannica	1965
2	A. Christie	Britannica	1980
3	N. Ammanniti	Italiana	1966
4	J. K. Rowling	Statunitense	1983

ISBN	Titolo	Prezzo	IDAutore
3858-38-4923	Harry Potter e la pietra filosofale	16,80	1
xxxxx	Dieci piccoli indiani	29,90	2
xxxxx	Harry Potter e il prigioniero di Azkaban	14,50	1
xxxxx	Io non ho paura	9,50	4

Modifiche

- Problemi analoghi nel caso di modifiche della chiave primaria

ID	Nome	Nazionalità	Nascita
1	J. K. Rowling	Britannica	1965
2	A. Christie	Britannica	1980
3	N. Ammanniti	Italiana	1966
5	J. K. Rowling	Statunitense	1983

4 rimpiazzato
con 5

ISBN	Titolo	Prezzo	IDAutore
3858-38-4923	Harry Potter e la pietra filosofale	16,80	1
xxxxx	Dieci piccoli indiani	29,90	2
xxxxx	Harry Potter e il prigioniero di Azkaban	14,50	1
xxxxx	Io non ho paura	9,50	4



Modifiche

■ Soluzioni

- Annullare l'operazione
- Effettuare le modifiche in cascata sulla chiave esterna
- Mettere a NULL i valori corrispondenti per la chiave esterna.

Interrogazioni

Interrogazioni

la **strutturazione** di un DB in **tabelle**, a loro volta strutturate in record con **attributi** omogenei, permette **interrogazioni** molto **sofisticate**

- Un database è inutile se non si possono recuperare da esso le informazioni che servono
 - l'operazione di recupero informazioni da un database si chiama **interrogazione**.

Interrogazioni

- Le interrogazioni possono essere più o meno semplici:
 - Dammi tutte le informazioni sul libro *Harry Potter e la Pietra Filosofale*
 - Dammi l'elenco degli autori che hanno scritto più di uno libro e per il quale la differenza di prezzo tra il libro più economico e quello più costoso è superiore a 20 €.
- È possibile avere risposta ad una interrogazione solo se il database contiene le informazioni rilevanti

Interrogazioni

- A quali di queste interrogazioni si può rispondere nel nostro caso?
 - Tutti i libri scritti da J. K. Rowling
 - Il numero di libri scritti da ogni autore
 - Gli autori che hanno scritto più di un libro e per il quale la differenza di prezzo tra il libro più economico e quello più costoso è superiore a 20 €
 - Gli autori che hanno scritto almeno un libro ogni 5 anni

Nome	Nazionalità	Nascita
J. K. Rowling	Britannica	1965
A. Christie	Britannica	1980
N. Ammanniti	Italiana	1966

Titolo	Prezzo	Autore
Harry Potter e la pietra filosofale	16,80	J. K. Rowling
Dieci piccoli indiani	29,90	A. Christie
Harry Potter e il prigioniero di Azkaban	14,50	J. K. Rowling
Io non ho paura	9,50	N. Ammanniti

Un linguaggio di interrogazione: SQL

- SQL (Structured Query Language):
 - è un (IL) linguaggio standard di uso dei DB
 - ogni venditore usa un proprio dialetto SQL, ma le interrogazioni semplici sono più o meno uguali:

SELECT *lista di campi*

FROM *tabella/e*

WHERE *vincoli sulle righe*

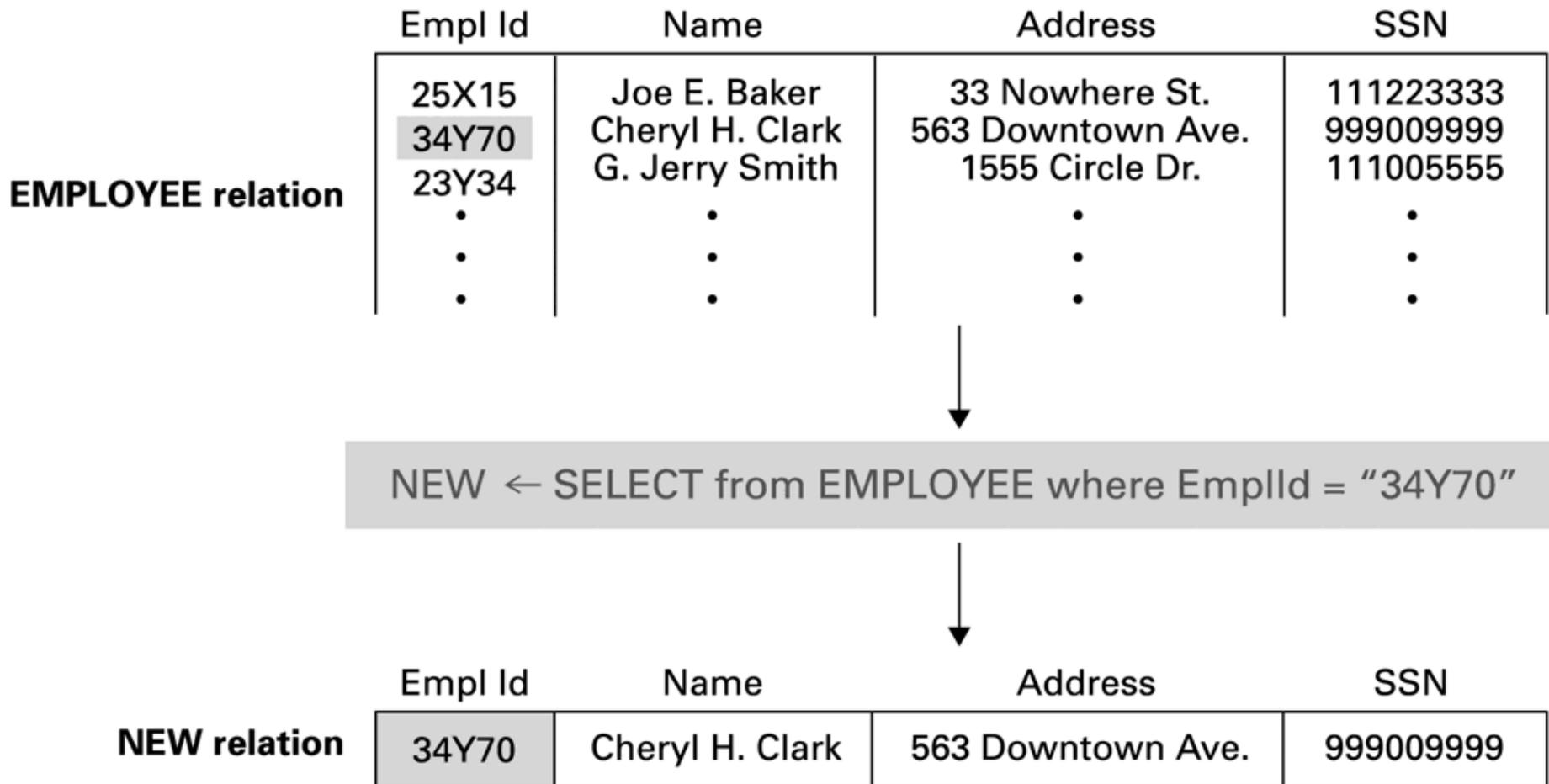
La clausola ON di SQL

- Dopo il SELECT c'è una lista di campi
- Il FROM specifica le tabelle da riunire nel Join

La clausola WHERE di SQL

- La clausola WHERE permette di specificare condizioni sulle righe:
 - il Preside ad esempio vuole vedere solo gli studenti con una media uguale o superiore al 28

L'operazione di SELECT



L'operazione di PROIEZIONE

	Empl Id	Name	Address	SSN
EMPLOYEE relation	25X15	Joe E. Baker	33 Nowhere St.	111223333
	24Y70	Cheryl H. Clark	563 Downtown Ave.	999009999
	23Y34	G. Jerry Smith	1555 Circle Dr.	111005555
	•	•	•	•
	•	•	•	•

MAIL ← PROJECT Name, Address from EMPLOYEE

	Name	Address
MAIL relation	Joe E. Baker	33 Nowhere St.
	Cheryl H. Clark	563 Downtown Ave.
	G. Jerry Smith	1555 Circle Dr.
	•	•
	•	•

Esempio (da Web)

FILM	<u>Codice</u>	Titolo	Anno	Durata	Regista
	LWR62	Lawrence d'Arabia	1962	180	LNEDVD
	PSG85	Passaggio in India	1985	150	LNEDVD

PERSONE	<u>Codice</u>	Cognome	Nome	Data_Nascita
	LNEDVD	Lean	David	1908
	OTLPTR	O'Toole	Peter	2/8/1932
	GNNALC	Guinness	Alec	2/4/1914
	HCKJCK	Hawkins	Jack	1910
	SHROMR	Sharif	Omar	10/4/1932
	DVSJDU	Davis	Judy	1956

RECITA	<u>Attore</u>	<u>Film</u>	<u>Parte</u>
	OTLPTR	LWR62	T.Lawrence
	SHROMR	LWR62	Sceriffo Ali
	GNNALC	LWR62	Re Faisal
	GMMGLN	LWR62	gen. Allenby
	DVSJDU	PSG85	Adela
	GNNALC	PSG85	prof. Godbole

Esempio (da Web)

```
SELECT Personaggio, Nome, Cognome
FROM Film, Recita, Persone
WHERE Film.Codice = Recita.Film
AND Recita.Attore = Persone.Codice
AND Film.Titolo LIKE "%Lawrence%"
```

SELECT ...	Personaggio	Nome	Cognome
	T.Lawrence	Peter	O'Toole
	Sceriffo Alì	Omar	Sharif
	Re Faisal	Alec	Guinness
	gen. Allenby	Jack	Hawkins

FILM	Codice	Titolo	Anno	Durata	Regista
	LWR62	Lawrence d'Arabia	1962	180	LNEDVD
	PSG85	Passaggio in India	1985	150	LNEDVD

PERSONE	Codice	Cognome	Nome	Data_Nascita
	LNEDVD	Lean	David	1908
	OTLPTR	O'Toole	Peter	2/8/1932
	GNNALC	Guinness	Alec	2/4/1914
	HCKJCK	Hawkins	Jack	1910
	SHROMR	Sharif	Omar	10/4/1932
	DVSJDU	Davis	Judy	1956

RECITA	Attore	Film	Parte
	OTLPTR	LWR62	T.Lawrence
	SHROMR	LWR62	Sceriffo Alì
	GNNALC	LWR62	Re Faisal
	GMMGLN	LWR62	gen. Allenby
	DVSJDU	PSG85	Adela
	GNNALC	PSG85	prof. Godbole

Interrogazioni

- in un DB relazionale, il **risultato** di una **interrogazione** è una **tabella** temporanea (persiste fintanto che il risultato viene analizzato)
- in una interrogazione specifichiamo:
 - gli **attributi** da includere nella tabella risultato
 - i **criteri di selezione** che devono essere soddisfatti dai **record** della tabella risultato

Interrogazioni

- consideriamo le tabelle Studenti ed Esami viste prima

Matricola	Cognome	Nome	Sesso	DataDiNascita	LuogoDiNascita
300000	Rossi	Mario	M	01/01/1987	Torino
300001	Verdi	Chiara	F	01/02/1987	Roma
300002	Rossi	Daria	F	01/01/1987	Torino
300003	Cancelli	Chiara	F	10/10/1987	Milano
300004	Votantonio	Raffaella	F	10/02/1985	Roma

Studenti

CodiceCorso	Voto	Data	Matricola
L0014	24	20/02/2007	300002
L0014	28	10/02/2007	300003
L0016	27	10/09/2006	300004
L0155	25	15/01/2007	300001
L0507	30	01/06/2006	300001
L0507	21	10/09/2006	300003

Esami

Interrogazioni

- vogliamo spedire delle congratulazioni agli **studenti** che hanno **superato** l'esame **L0507**

CodiceCorso	Voto	Data	Matricola
L0014	24	20/02/2007	300002
L0014	28	10/02/2007	300003
L0016	27	10/09/2006	300004
L0155	25	15/01/2007	300001
L0507	30	01/06/2006	300001
L0507	21	10/09/2006	300003

Esami

Matricola	Cognome	Nome	Sesso	DataDiNascita	LuogoDiNascita	Domicilio
300000	Rossi	Mario	M	01/01/1987	Torino	v. Fiore 2, Torino
300001	Verdi	Chiara	F	01/02/1987	Roma	v. Casa 31, Roma
300002	Rossi	Daria	F	01/01/1987	Torino	v. Fiore 2, Torino
300003	Cancelli	Chiara	F	10/10/1987	Milano	v. Tetti 1, Torino
300004	Votantonio	Raffaella	F	10/02/1985	Roma	p. Cane 1, Roma

Studenti

CodiceCorso	Voto	Data	Matricola	Cognome	Nome	Sesso	DataDiNascita	LuogoDiNascita
L0155	25	15/01/2007	300001	Verdi	Chiara	F	01/02/1987	Roma
L0507	30	01/06/2006	300001	Verdi	Chiara	F	01/02/1987	Roma
L0014	24	20/02/2007	300002	Rossi	Daria	F	01/01/1987	Torino
L0014	28	10/02/2007	300003	Cancelli	Chiara	F	10/10/1987	Milano
L0507	21	10/09/2006	300003	Cancelli	Chiara	F	10/10/1987	Milano
L0016	27	10/09/2006	300004	Votantonio	Raffaella	F	10/02/1985	Roma

SQL

- vogliamo spedire delle congratulazioni agli studenti che hanno superato l'esame L0507

```
SELECT *  
FROM Studenti S, Esami E  
WHERE S.Matricola = E.Matricola AND  
       CodiceCorso = "L0507"
```

Interrogazioni

- vogliamo spedire delle congratulazioni agli **studenti** che hanno **superato** l'esame **L0507** con **voto ≥ 27**

CodiceCorso	Voto	Data	Matricola
L0014	24	20/02/2007	300002
L0014	28	10/02/2007	300003
L0016	27	10/09/2006	300004
L0155	25	15/01/2007	300001
L0507	30	01/06/2006	300001
L0507	21	10/09/2006	300003

Esami

CodiceCorso	Voto	Data	Matricola	Cognome	Nome	Sesso	DataDiNascita	LuogoDiNascita
L0155	25	15/01/2007	300001	Verdi	Chiara	F	01/02/1987	Roma
L0507	30	01/06/2006	300001	Verdi	Chiara	F	01/02/1987	Roma
L0014	24	20/02/2007	300002	Rossi	Daria	F	01/01/1987	Torino
L0014	28	10/02/2007	300003	Cancelli	Chiara	F	10/10/1987	Milano
L0507	21	10/09/2006	300003	Cancelli	Chiara	F	10/10/1987	Milano
L0016	27	10/09/2006	300004	Votantonio	Raffaella	F	10/02/1985	Roma

Matricola	Cognome	Nome	Sesso	DataDiNascita	LuogoDiNascita
300000	Rossi	Mario	M	01/01/1987	Torino
300001	Verdi	Chiara	F	01/02/1987	Roma
300002	Rossi	Daria	F	01/01/1987	Torino
300003	Cancelli	Chiara	F	10/10/1987	Milano
300004	Votantonio	Raffaella	F	10/02/1985	Roma

Studenti

SQL

- vogliamo spedire delle congratulazioni agli **studenti** che hanno **superato** l'esame **L0507** con **voto ≥ 27**

```
SELECT *  
FROM Studenti S, Esami E  
WHERE S.Matricola = E.Matricola AND  
       CodiceCorso = "L0507" AND  
       Voto >= 27
```

Interrogazioni

- vogliamo spedire delle congratulazioni agli **studenti** che hanno **superato** l'esame **L0014** o **L0507** con **voto ≥ 27**

CodiceCorso	Voto	Data	Matricola
L0014	24	20/02/2007	300002
L0014	28	10/02/2007	300003
L0016	27	10/09/2006	300004
L0155	25	15/01/2007	300001
L0507	30	01/06/2006	300001
L0507	21	10/09/2006	300003

Esami

CodiceCorso	Voto	Data	Matricola	Cognome	Nome	Sesso	DataDiNascita	LuogoDiNascita
L0155	25	15/01/2007	300001	Verdi	Chiara	F	01/02/1987	Roma
L0507	30	01/06/2006	300001	Verdi	Chiara	F	01/02/1987	Roma
L0014	24	20/02/2007	300002	Rossi	Daria	F	01/01/1987	Torino
L0014	28	10/02/2007	300003	Cancelli	Chiara	F	10/10/1987	Milano
L0507	21	10/09/2006	300003	Cancelli	Chiara	F	10/10/1987	Milano
L0016	27	10/09/2006	300004	Votantonio	Raffaella	F	10/02/1985	Roma

Matricola	Cognome	Nome	Sesso	DataDiNascita	LuogoDiNascita
300000	Rossi	Mario	M	01/01/1987	Torino
300001	Verdi	Chiara	F	01/02/1987	Roma
300002	Rossi	Daria	F	01/01/1987	Torino
300003	Cancelli	Chiara	F	10/10/1987	Milano
300004	Votantonio	Raffaella	F	10/02/1985	Roma

Studenti

SQL

- vogliamo spedire delle congratulazioni agli **studenti** che hanno **superato** l'esame **L0014** o **L0507** con **voto ≥ 27**

```
SELECT *  
FROM Studenti S, Esami E  
WHERE S.Matricola = E.Matricola AND  
      (CodiceCorso = "L0507" OR  
       CodiceCorso = "L0014") AND  
      Voto >= 27
```

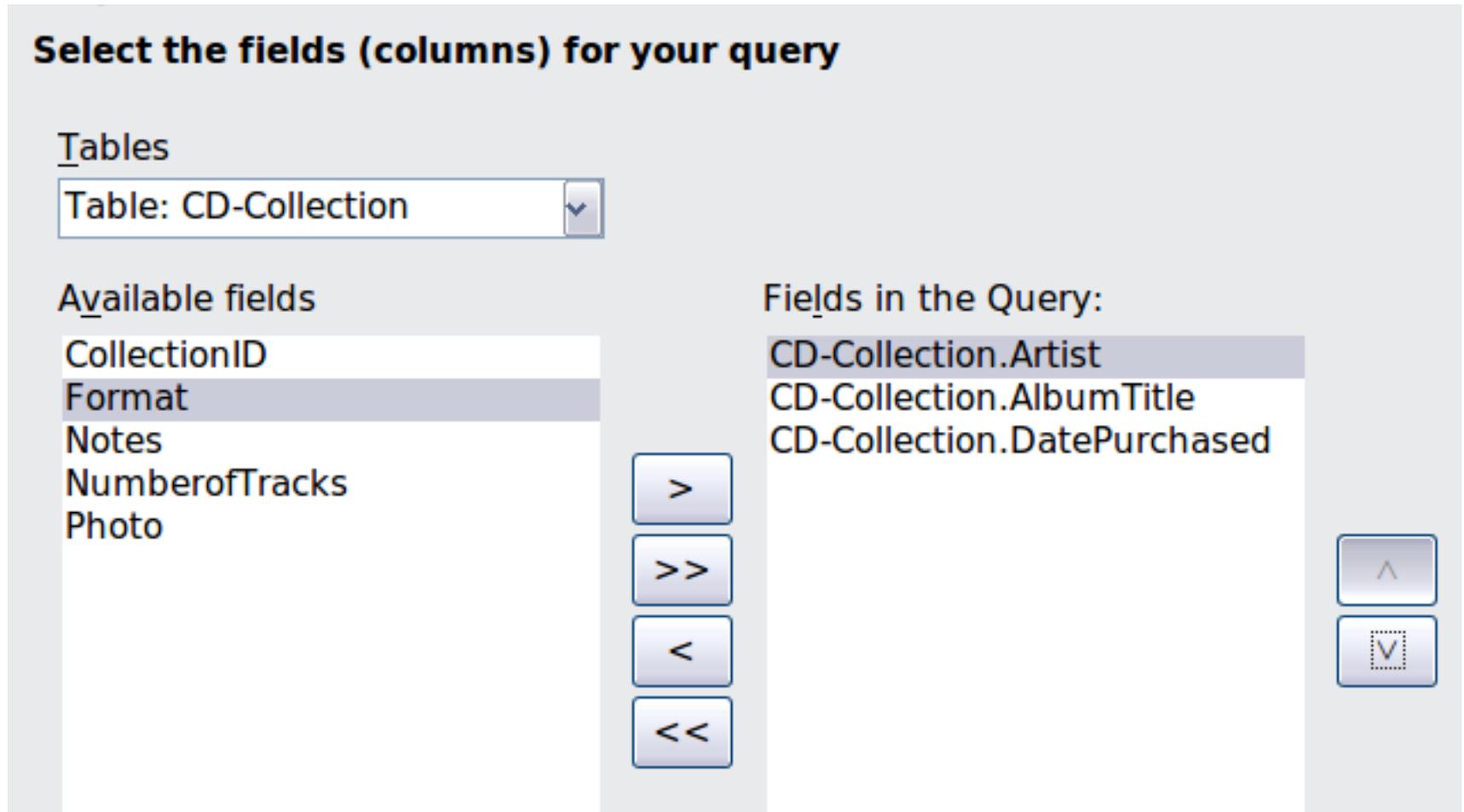
LibreOffice Base: tool per creare Query in maniera visuale

Select the fields (columns) for your query

Tables
Table: CD-Collection

Available fields
CollectionID
Format
Notes
NumberofTracks
Photo

Fields in the Query:
CD-Collection.Artist
CD-Collection.AlbumTitle
CD-Collection.DatePurchased



Esempio

- Il DB di una azienda contiene le seguenti tabelle:

Impiegati

Cognome, Nome, ID, Indirizzo, Città, Provincia, CAP, Telefono, Data_ass

Stipendi

ID, Paga, Deduzioni, Sanità, Vita

Risorse_Umane

ID, Dipa, Data_ass, Commenti, Supervisor, Progetti

Calcetto

ID, Piede, Presenze, Goal, Posizione

Esempio – Chiavi Primarie

- Esplicitiamo le chiavi primarie:

Impiegati

Cognome, Nome, **ID**, Indirizzo, Città, Provincia, CAP, Telefono, Data_ass

Stipendi

ID, Paga, Deduzioni, Sanità, Vita

Risorse_Umane

ID, Dipa, Data_ass, Commenti, Supervisor, Progetti

Calcetto

ID, Piede, Presenze, Goal, Posizione

Esempio - Associazioni

■ Consideriamo ora le associazioni

Impiegati

Cognome, Nome, ID, Indirizzo, Città, Provincia, CAP,
Telefono, Data_ass

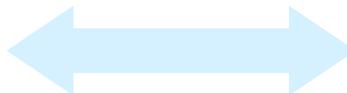
Stipendi

ID, Paga, Deduzioni, Sanità, Vita

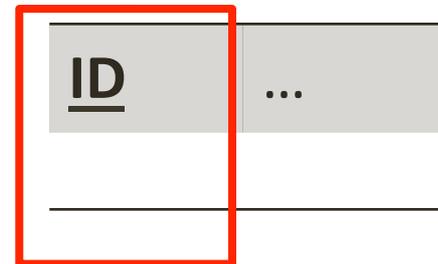
Impiegati



uno-a-uno



Stipendi



Esempio - Associazioni

■ Consideriamo ora le associazioni

Impiegati

Cognome, Nome, ID, Indirizzo, Città, Provincia, CAP,
Telefono, Data_ass

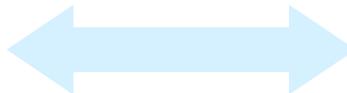
Risorse_Umane

ID, Dipa, Data_ass, Commenti, Supervisor, Progetti

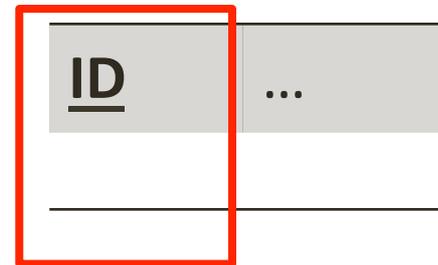
Impiegati



uno-a-uno



Risorse_Umane



Esempio - Associazioni

■ Consideriamo ora le associazioni

Impiegati

Cognome, Nome, ID, Indirizzo, Città, Provincia, CAP,
Telefono, Data_ass

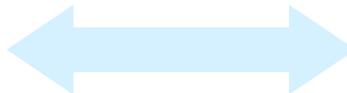
Calcetto

ID, Piede, Presenze, Goal, Posizione

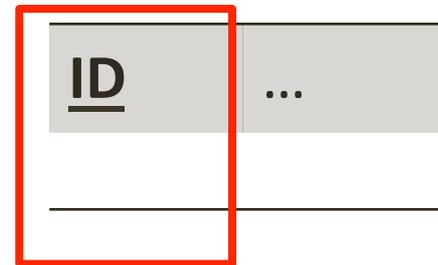
Impiegati



uno-a-uno



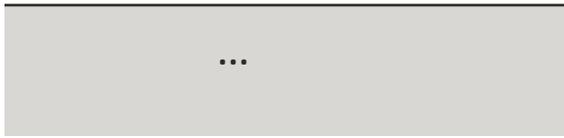
Calcetto



Esempio - Associazioni

- Abbiamo individuato le associazioni:

Impiegati



Risorse_Umane



uno-a-uno



uno-a-uno



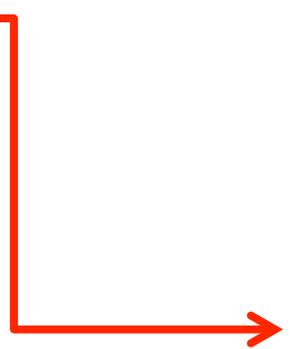
Stipendi



Calcetto



uno-a-uno



Esempio - Associazioni

■ Ci sono altre associazioni?

Impiegati

Cognome, Nome, ID, Indirizzo, Città, Provincia, CAP, Telefono, Data_ass

Stipendi

ID, Paga, Deduzioni, Sanità, Vita

Risorse_Umane

ID, Dipa, Data_ass, Commenti, Supervisor, Progetti

Calcetto

ID, Piede, Presenze, Goal, Posizione

Esempio - Associazioni

- C'è un'altra associazione tra Impiegati e Risorse_Umane

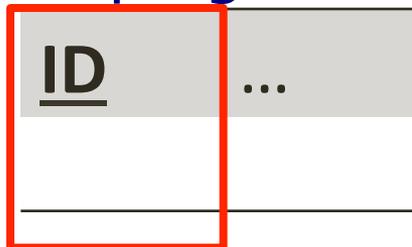
Impiegati

Cognome, Nome, ID, Indirizzo, Città, Provincia, CAP, Telefono, Data_ass

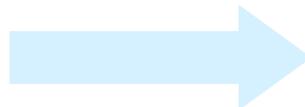
Risorse_Umane

ID, Dipa, Data_ass, Commenti, Supervisor, Progetti

Impiegati



uno-a-molti

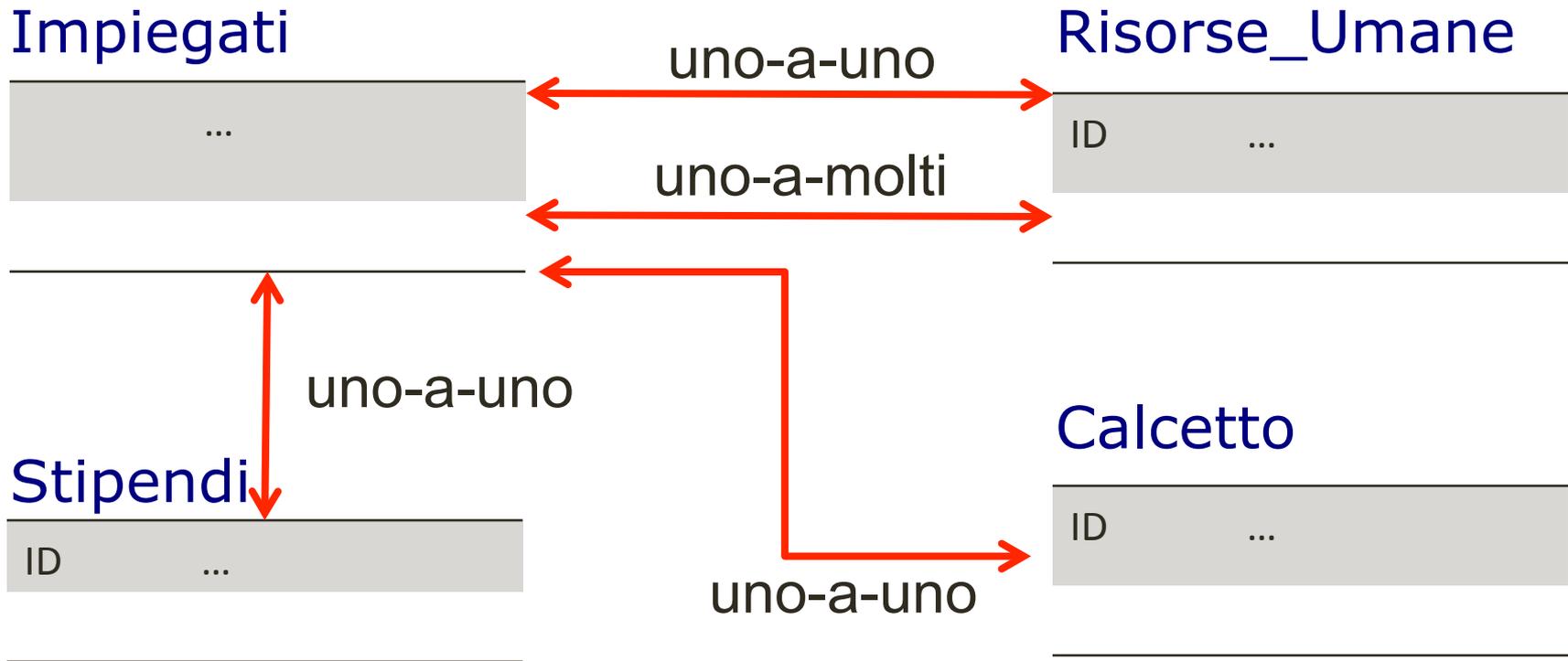


Risorse_Umane

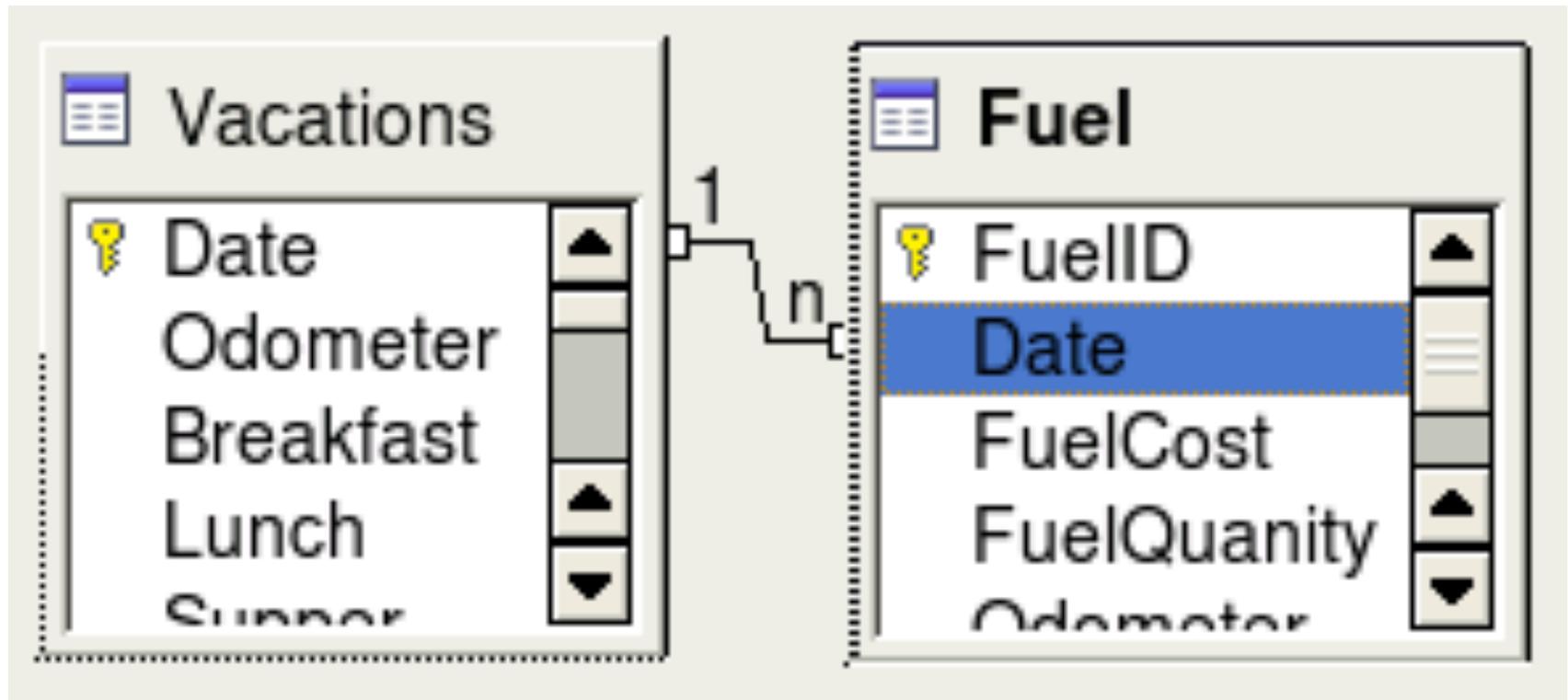


Esempio - Associazioni

- Abbiamo individuato le associazioni:



LibreOffice Base: relazioni



Esempio - Ridondanza

- La data di assunzione **Data_ass** è ridondante
- Per rimuovere la ridondanza possiamo **rimuovere** l'attributo **Data_ass** dalla tabella **Impiegati** oppure dalla tabella **Risorse_Umane**
- In questo caso sembra preferibile rimuovere **Data_ass** da **Risorse_Umane** perché altrimenti, il DB potrebbe non specificare la data di assunzione per qualche impiegato

Esempio - View

- Consideriamo il DB da diversi punti di vista (**view**)
- L'allenatore di calcio non dovrà (necessariamente) avere accesso a tutte le tabelle **Impiegati**, **Stipendi**, **Risorse_Umane** e **Calcetto**
- Gli sarà sufficiente avere accesso alla tabella **Calcetto** e a (parte) dei dati nella tabella **Impiegati** relativi a impiegati che giocano a calcio

Esempio - View

- La **view** dell'allenatore di calcetto potrebbe essere il **risultato** di una **query**.

Impiegati

<u>ID</u>	...
-----------	-----

Calcetto

<u>ID</u>	...
-----------	-----

ViewAllenatore

ID	Attributi di Calcetto	Attributi di Impiegati tranne Data_ass
----	-----------------------	--
