DATABASE

Marco Bertini
Università degli Studi di Firenze

Obiettivi della lezione

- Concetti base
  - dati e informazioni
  - cos’ è un database
  - terminologia
- Modelli organizzativi
  - flat file
  - database relazionali
- Principi e linee guida nel progetto
  - normalizzazione
  - esempi di progetto

Bibliografia

- Testi sul progetto di database:
  - P. Atzeni, S. Ceri, S. Paraboschi, R. Torlone. “Basi di dati”, Mc Graw Hill 2002.

- Web e database:
  - J: Engels “Database con MySql’ ’ , Jackson 2002.

Concetti base e linee guida nel progetto

Dati e informazioni

- Nel seguito affronteremo essenzialmente il problema di memorizzare ed organizzare dati.
  - Un dato è un qualunque tipo di valore statico. I suoi valori possono assumere forme diverse, da una semplice scelta si/no, ad un valore numerico, ad un blocco di testo.
- Il problema di trasformare dati in informazioni è completamente diverso da quello di costruire dei "contenitori" per i dati.
- Comunque, esistono metodi per progettare la struttura del database che rendono questa elaborazione più facile da compiere.

Cos'è un database

- Un database è fondamentalmente una collezione di registrazioni (records) che concernono qualche sorta di azione od oggetto.
- Più formalmente, un database è qualsiasi collezione di "fatti" organizzati in modo sistematico.
  - Es. Le schede del catalogo di una biblioteca. Ogni scheda, fisica o virtuale, contiene fatti (autore, titolo, ISBN, etc.) che sono organizzati (alfabeticamente per titolo o autore, numericamente per ISBN) al fine di rendere la loro ricerca più semplice rispetto ad una scansione di tutti i libri della biblioteca.

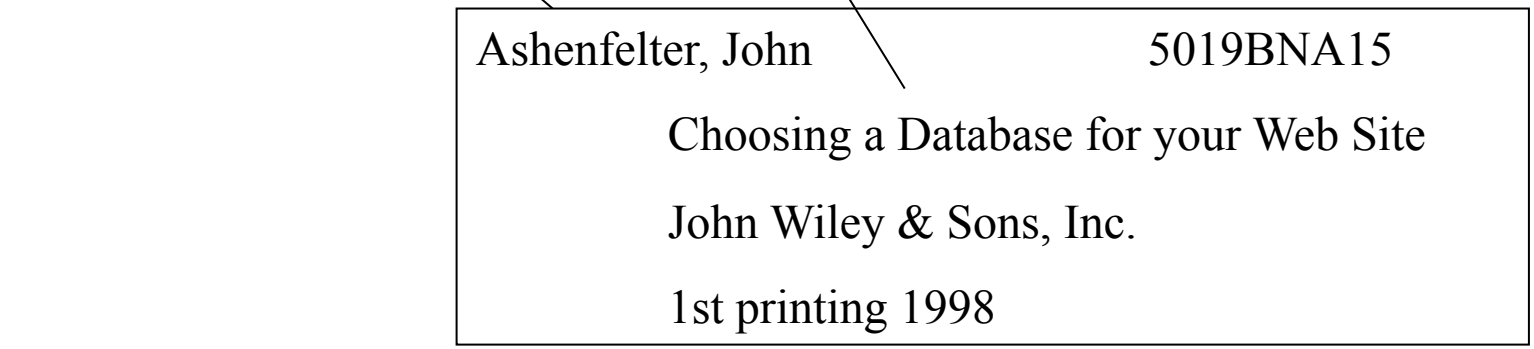
- Esistono diversi tipi di database:

- Database gerarchici
- Database a rete
- Database relazionali
- Database a oggetti

- Access, MySQL, SQL Server etc. sono database relazionali

Es. Scheda e Rappresentazione Tabellare

Autore	Titolo	Numero di codice
Ashenfelter, John	Choosing a Database for your Web Site	5019BNA15
Botto, Francis	Multimedia and Internet Applications	2212MKV26
Cintron, Dave	Fast Track Web Programming	7682QYT91



Terminologia

- Un certo numero di termini descrivono le parti di un database. Sono termini standard, indipendenti dal particolare software con cui è implementato il database.

  - Campo - è l'unità strutturale di base del database. È il contenitore per un dato o per ogni sua parte logica.
    Ad esempio, un indirizzo di posta potrebbe essere il campo ma, in un diverso contesto, potrebbero essere campi separati la via, il numero civico, la città, lo stato ed il codice postale.

Terminologia

- Record - è un insieme di campi che descrivono un'unità più grande. È detto anche riga. I campi in un record forniscono una descrizione completa di ogni elemento in una collezione. Un record è un'istanza unica di dati che riguardano un oggetto o un evento.

- Tabella - è il nome formale dato ad un gruppo di records che contengono gli elementi di una collezione. Di norma una tabella rappresenta un oggetto distinto (es. tabella libri in una biblioteca) o un evento (es. tabella degli ordini per un prodotto).

Terminologia

- Chiave - è un identificatore univoco per ogni riga (record) in una tabella di dati.
- Anche se un record singolo rappresenta una porzione separata dei dati, alcuni di questi records possono apparire identici.
- La chiave rappresenta un modo non ambiguo per identificare records distinti e serve come puntatore ad un particolare record della tabella.
  - Es.: ogni cittadino italiano è univocamente identificato dal codice fiscale.
  - In molti casi, le chiavi per le tabelle dei dati sono costruite semplicemente aggiungendo un campo al record con funzione di chiave.

Terminologia

Key *Field*

ID	LastName	FirstName	Address	City	State	Zip
2231	Adamson	Robert	1021 B Wheatley	Lawrence	NY	11559
2232	Anderson	John	1024 Santee St.	Owings Mill	MD	21117
2233	Atget	Eugene	103 N Market	Cleveland	OH	44135
2234	Baldus	Denis	1031-1 S Dolfield	Los Angeles	CA	90015

Record

Tabella indirizzi

Terminologia

- Spesso utenti diversi vogliono guardare ai dati di una stessa o di più tabelle in modi diversi.
  - Ad esempio, se fornitori e merce sono due tabelle in un database, può essere utile avere una vista di quali merci sono fornite da un certo fornitore. Questo dato può essere ottenuto dalle due tabelle e combinato in una tabella virtuale.
- Vista - è una tabella "virtuale", nel senso che la tabella non è fisicamente presente nel database nel modo con cui è presentata all'utente. La vista è generata come risposta ad una particolare interrogazione. Permette di presentare in modi diversi i dati contenuti nel database.

Terminologia

- Database - è una collezione di tabelle. Spesso include anche maschere per l'ingresso dei dati, regole per la verifica della correttezza e la validazione dei dati immessi, ed il formato per creare reports informativi dei dati nel database.

Tipi di Database

- Esistono diversi modi in cui i database sono implementati. Si possono individuare tre categorie di base:
  - database flat-file
  - database relazionali
  - database orientati agli oggetti

Database Flat-File

- È il tipo più elementare di organizzazione dei dati in un database. La caratteristica base di un flat-file è che tutti i dati sono memorizzati insieme in una singola tabella. La maggior parte dei database cartacei appartengono a questa categoria.
  - Caratterizzati da semplicità di progetto e implementabili in modo diretto. Un semplice file di testo con dati organizzati può rappresentare un flat-file.
  - È il tipo di organizzazione meno efficiente e più problematica. Il maggior svantaggio è dovuto al fatto che parte dei dati inseriti possono essere ridondanti o inconsistenti.

Flat-File: database di ordini

OrderID	CustID	Name	Shipping Address	ItemCode	Quantity	Price
23455	1521	Velio Cooper	152 W Waterloo St Austin, TX 78752	ZD-552	1	8.95
23456	1567	Francis Cioni	1445 E Delavan Ave Laredo, TX 78043	XT-211	2	16.50
23457	1567	Francis Cioni	1445 E Delavan Ave Laredo, TX 78043	XT-212	2	22.00
23458	1765	Jane Carroll	1551 Westwood Blvd Ocilla, GA 31774	CC-48	15	225.15
23459	0021	Jody Hawes	2664 Woodhill Rd Bronx, NY	A-567	1	19.99
23460	1235	David Hill	742 Hearne Cleveland, OH 44104	XD-234	1	49.95

Database Flat-File

- Il database dell'esempio precedente ha un evidente problema di progetto: ogni record è relativo all'ordine di un solo prodotto per un dato cliente.

- Se un cliente ordina prodotti diversi, si avranno tanti records per il cliente quanti sono gli ordini (come mostrato per il cliente con CustID=1567)
  - questo porta alla duplicazione dei dati relativi al cliente
    (Name, Shipping Address) in ognuno di questi record.

Database Flat-File

- In maniera alternativa, il problema potrebbe essere risolto aggiungendo alla tabella un altro insieme di campi (colonne della tabella) ItemCode, Price e Quantity, numerandoli in sequenza (così da distinguere i campi per il primo prodotto da quelli del secondo e dei successivi).

Name	Shipping Address	ItemCode1	Quantity1	Price1	ItemCode2	Quantity2	Price2
Velio Cooper	152 W Waterloo St Austin, TX 78752	ZD-552	1	8.95			
Francis Cioni	1445 E Delavan Ave Laredo, TX 78043	XT-211	2	16.50	XT-212	2	22.00

Database Flat-File

- Quanti elementi sono sufficienti?
  - se si considera un numero massimo di prodotti ammessi per ogni ordine piccolo, potrebbe accadere che un ordine non possa essere soddisfatto (non è previsto un numero sufficiente di campi);
  - se è previsto un numero massimo ammissibile di prodotti troppo grande si produce un evidente spreco di spazio.

Database relazionali

- Sono stati sviluppati con l'obiettivo di prevenire una non necessaria duplicazione dei dati nel database.
- Il problema, nell'esempio del database di ordini, nasce dal fatto che nella tabella sono rappresentate due entità logicamente distinte:
  - il cliente e i dati ad esso relativi;
  - l'insieme di prodotti che sono stati ordinati.
- In un database relazionale ogni entità logicamente distinta dalle altre è rappresentata in una differente tabella del database.

Database relazionali: database di ordini

- Due tabelle:
  - tabella ordini, con i dati relativi al cliente;
  - tabella prodotti, con i dati relativi ai prodotti di ogni singolo ordine.

Database relazionali: database di ordini

OrderID	CustID	Name	Shipping Address
23455	1521	Velio Cooper	152 W Waterloo St Austin, TX 78752
23457	1567	Francis Cioni	1445 E Delavan Ave Laredo, TX 78043
23458	1765	Jane Carroll	1551 Westwood Blvd Ocilla, GA 31774
23459	0021	Jody Hawes	2664 Woodhill Rd Bronx, NY
23460	1235	David Hill	742 Hearne Cleveland, OH 44104

Tabella ordini

One

Many

Tabella prodotti

OrderID	ItemCode	Quantity	Price
23455	ZD-552	1	8.95
23457	XT-211	2	16.50
23457	XT-212	2	22.00
23458	CC-48	15	225.15
23459	A-567	1	19.99
23460	XD-234	1	49.95

Database relazionali

- La chiave OrderID nella tabella degli ordini descrive univocamente ciascun ordine di un cliente.
- Ogni prodotto ordinato è individualmente e univocamente memorizzato in una tabella separata dei prodotti.
- Il cliente può, nello stesso ordine, richiedere quanti prodotti desidera senza limitazioni a priori sul numero massimo, né spreco di memoria.
- Tra le tabelle è stabilita una relazione tramite le rispettive chiavi.

Database relazionali

- Ogni tabella ha la propria chiave primaria per identificare i suoi elementi.
- Notare che la chiave OrderID della tabella degli ordini è inserita nella tabella dei prodotti per ogni prodotto che è stato ordinato da un cliente.
- Ogni prodotto che è stato ordinato può essere legato al cliente appropriato cercando tutti i records della tabella prodotti che hanno un particolare valore nel campo OrderID.

Database relazionali: relazione uno-a-molti

- Nella tabella dei prodotti OrderID non è un valore univoco.
- La relazione tra la tabella degli ordini e quella dei prodotti è detta relazione uno-a-molti in quanto un record nella tabella degli ordini è legato ad un numero variabile di records nella tabella dei prodotti.
- OrderID serve come chiave esterna nella tabella dei prodotti, in quanto il valore della chiave viene da una diversa tabella (dove i valori sono univoci).

Database relazionali: database degli ordini

- Nota: per evitare la duplicazione degli indirizzi di un cliente tra più ordini successivi sarebbe conveniente creare una ulteriore tabella dei clienti, con solo i dati relativi al cliente, e legarla con una relazione uno-a-molti, stabilita attraverso il CustID, con la tabella degli ordini.
  - In questo modo ogni cliente può effettuare più ordini ed ogni ordine può comprendere più prodotti.

Database relazionale: database di ordini

CustID	Name	Shipping Address
1521	Velio Cooper	152 W Waterloo St Austin, TX 78752
1567	Francis Cioni	1445 E Delavan Ave Laredo, TX 78043
1765	Jane Carroll	1551 Westwood Blvd Ocilla, GA 31774
0021	Jody Hawes	2664 Woodhill Rd Bronx, NY
1235	David Hill	742 Hearne Cleveland, OH 44104

Tabella clienti

OrderID	CustID	Date
23455	1521	2002.02.25
23457	1567	2002.02.13
23458	1765	2002.01.19
23459	1521	2002.02.13
23460	1235	2001.12.21

Tabella ordini

Tabella prodotti

OrderID	ItemCode	Quantity	Price
23455	ZD-552	1	8.95
23457	XT-211	2	16.50
23457	XT-212	2	22.00
23458	CC-48	15	225.15
23459	A-567	1	19.99
23460	XD-234	1	49.95

Database relazionali: tipi di relazioni

- Oltre alla relazione uno-a-molti esistono altri due modi possibili per relazionare due tabelle del database:
  - relazione uno-ad-uno;
  - relazione molti-a-molti.

Database relazionali: relazione uno-a-uno

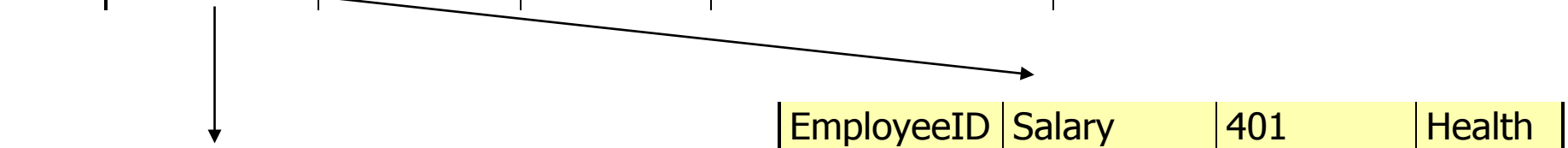
- In una relazione uno-a-uno ogni record in una tabella è collegato ad uno ed uno solo dei record in un'altra tabella.

- Nella maggior parte dei casi, questo tipo di relazione è stabilita tra una tabella che rappresenta un insieme di dati relativi ad un sottoinsieme delle entità in una tabella "principale", e la tabella principale stessa.

Relazione uno-a-uno: database delle risorse umane

EmployeeID	LastName	FirstName	Address	Phone
77620	Southworth	Jerry	496 Laguardia Place	555-2222
77621	Talbot	Gina	426 Lake St	555-8267
77622	Russell	David	318 W 39 Th St	555-4445
77623	Thompson	Linda	2427 Broadway	555-1568
77624	Albertson	Steve	224 Lawrence Ave	555-9872
77625	Davis	Russell	224 Lawrence Ave	555-1213

Tabella impiegati



EmployeeID	Rate	Hours
77620	7.78	30
77625	13.55	40

Tabella impiegati a ore

EmployeeID	Salary	401	Health
77621	41,000	5	BC
77622	55,755	2	BC
77623	22,155	N/A	QS
77624	18,975	2	N/A

Tabella impiegati salariati

Database relazionali: relazione uno-a-uno

- La tabella degli impiegati ha un record per ogni impiegato dell'azienda ed usa EmployeeID come chiave primaria per identificarli in modo univoco.

- Supponiamo che l'azienda preveda due tipi di impiegati:
  - salariati;
  - a ore.

- La tabella impiegati contiene solo i dati di base relativi ad entrambe le categorie: nome, cognome, indirizzo e numero telefonico.

Database relazionali: relazione uno-a-uno

- Dato che per le due categorie di impiegati devono essere mantenuti dati diversi, questi sono inclusi in due tabelle ausiliarie:
  - la tabella degli impiegati a ore. Mantiene solo i dati aggiuntivi che li riguardano, come il costo orario e le ore lavorate;
  - la tabella degli impiegati salariati. La tabella mantiene solo i dati specifici per questa categoria, come il salario complessivo e l'informazione sanitaria.

- Per entrambe le tabelle ausiliarie la chiave primaria è rappresentata da EmployeeID

Database relazionali: relazione uno-a-uno

- Ogni impiegato può apparire una sola volta nella tabella impiegati, ed un record nella tabella degli impiegati a ore rappresenta un solo impiegato.
  Lo stesso vale per la tabella degli impiegati salariati.

- L'uso della relazione uno-a-uno tra la tabella principale e ciascuna delle due tabelle ausiliarie ha permesso di separare i dati che riguardano solo sottoinsiemi di records nella tabella principale. In questo modo non è stato necessario inserire campi aggiuntivi nella tabella principale con conseguente risparmio di memoria.

Database relazionali: relazione molti-a-molti

- La caratteristica distintiva di una relazione molti-a-molti tra due tabelle è che una terza tabella è necessaria per rappresentare la relazione.

- La tabella che descrive la relazione semplicemente mantiene la chiave primaria da una tabella, con la chiave primaria dei record ad essa collegati nella seconda tabella, insieme con ogni altra informazione che contraddistingue la relazione.

Database relazionali: relazione molti-a-molti

- La relazione molti-a-molti permette di rappresentare relazioni in casi in cui più records in una tabella sono in relazione con più records in un' altra tabella.

- Un esempio classico è quello di un database di corsi e studenti. Ogni corso è seguito da più studenti, ma ogni studente segue molti corsi. Perciò è necessaria una relazione molti-a-molti tra la tabella dei corsi e quella degli studenti.

- Un altro esempio è quello della relazione tra impiegati e progetti.

Relazione molti-a-molti: database impiegati e progetti

Tabella impiegati

EmployeeID	LastName	FirstName
77620	Southworth	Jerry
77621	Talbot	Gina
77622	Russell	David
77623	Thompson	Linda
77624	Albertson	Steve
77625	Davis	Russell

Tabella progetti

ProjectID	Manager	Deadline
22356	Jones	4/9
27685	Lewis	12/22
33564	Albertson	8/17

EmployeeID	ProjectID
77620	22356
77621	27685
77624	27685
77622	33564
77624	33564
77622	22356

Tabella relazione impiegati-progetti

Relazione molti-a-molti

- Database di progetti che tiene traccia degli impiegati che lavorano su ciascun progetto.
- Ogni impiegato dell'azienda lavora a molti progetti, ed ogni progetto ha molti impiegati dedicati al suo svolgimento.
- Sono individuate due tabelle per rappresentare le due entità distinte:
  - tabella impiegati;
  - tabella progetti.
- È necessaria un'ulteriore tabella per rappresentare la relazione molti-a-molti tra impiegati e progetti.

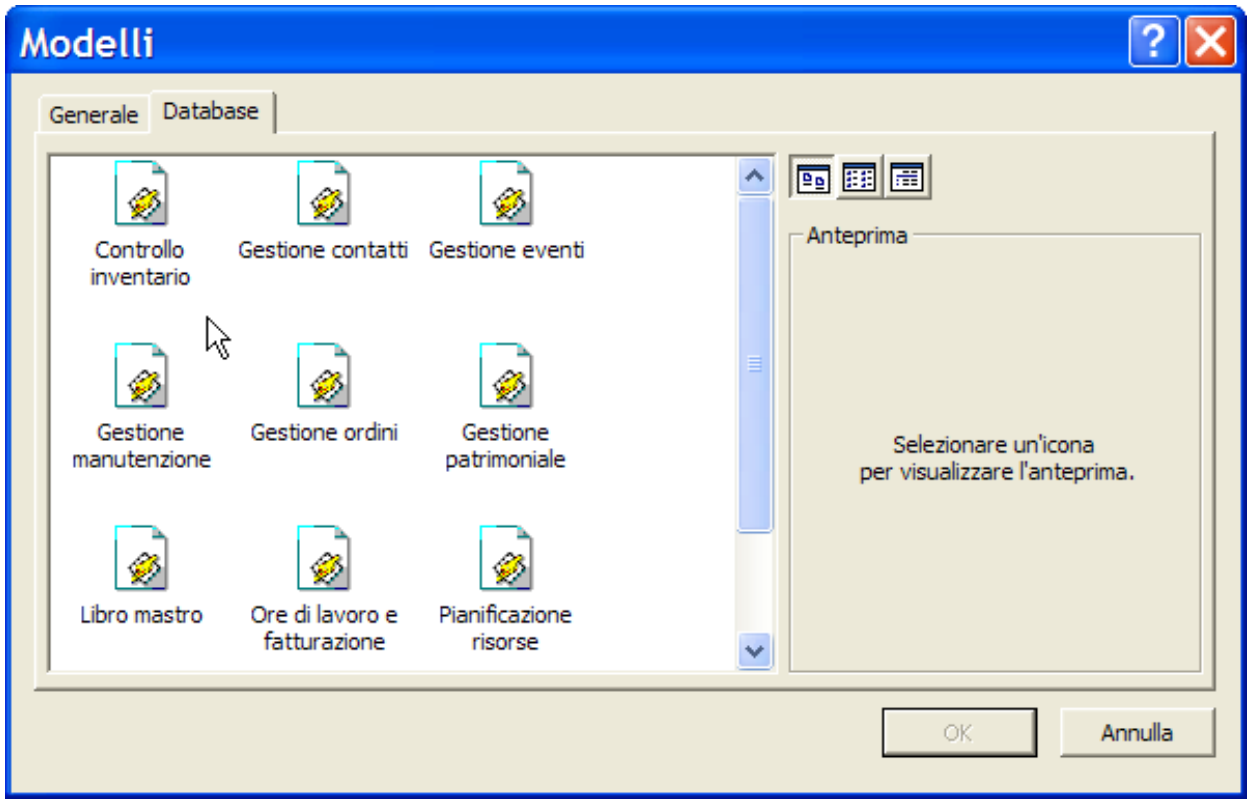
Relazione molti-a-molti

- La tabella di relazione tra impiegati e progetti è costruita usando le chiavi primarie delle tabelle impiegati e progetti.

- Sia i valori della chiave per la tabella degli impiegati (EmployeeID), sia quelli della chiave per la tabella dei progetti (ProjectID), possono essere ripetuti più volte nella tabella della relazione.

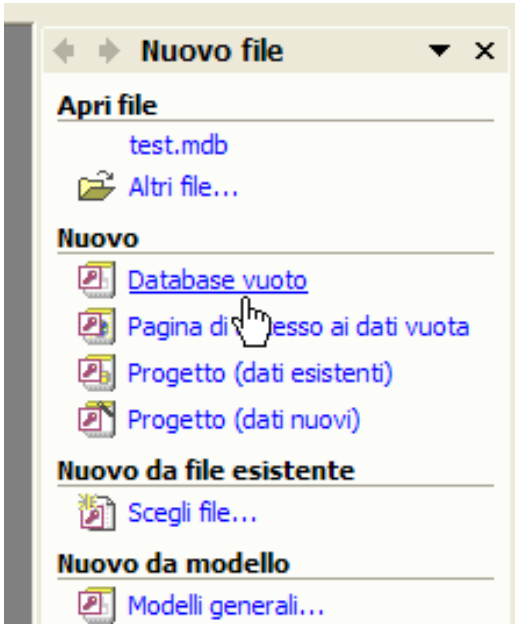
Tabelle e relazioni in Access

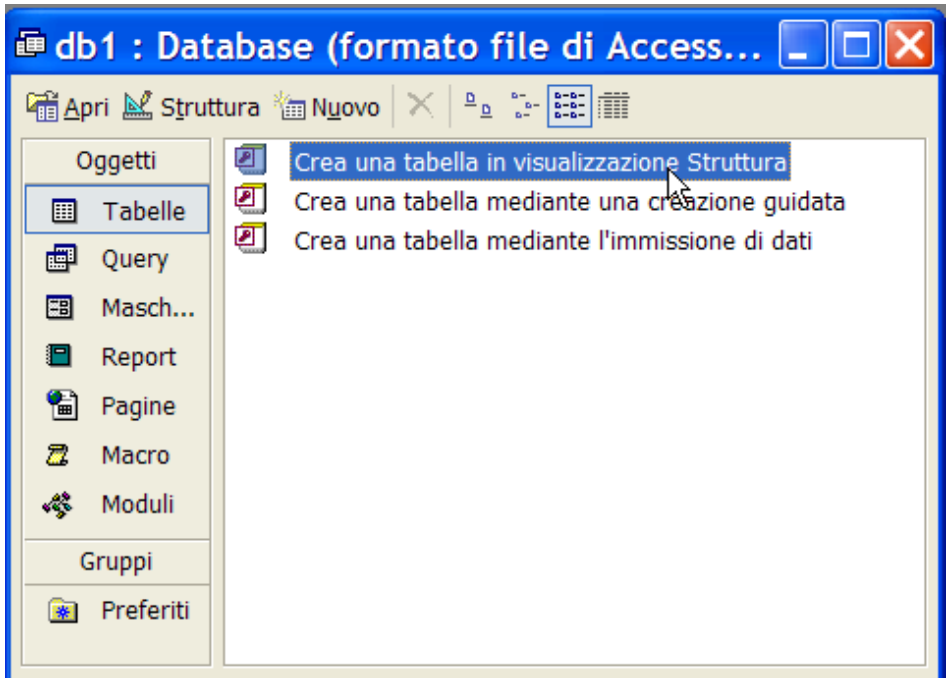
- Access ha un set di database pronti di esempio



- Dal modello Access crea tabelle, relazioni e query

- In alternativa si crea un database vuoto e si creano tabelle e relazioni manualmente o in modo guidato




db1 : Database (formato file di Access...
Apri
Struttura
Nuovo
Oggetti
Tabelle
Query
Masch...
Report
Pagine
Macro
Moduli
Gruppi
Preferiti
Crea una tabella in visualizzazione Struttura
Crea una tabella mediante una creazione guidata
Crea una tabella mediante l'immissione di dati

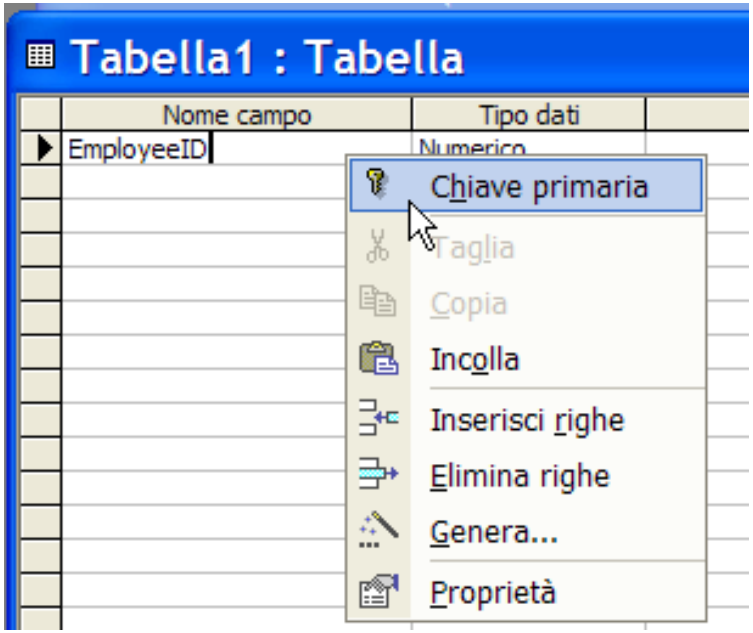
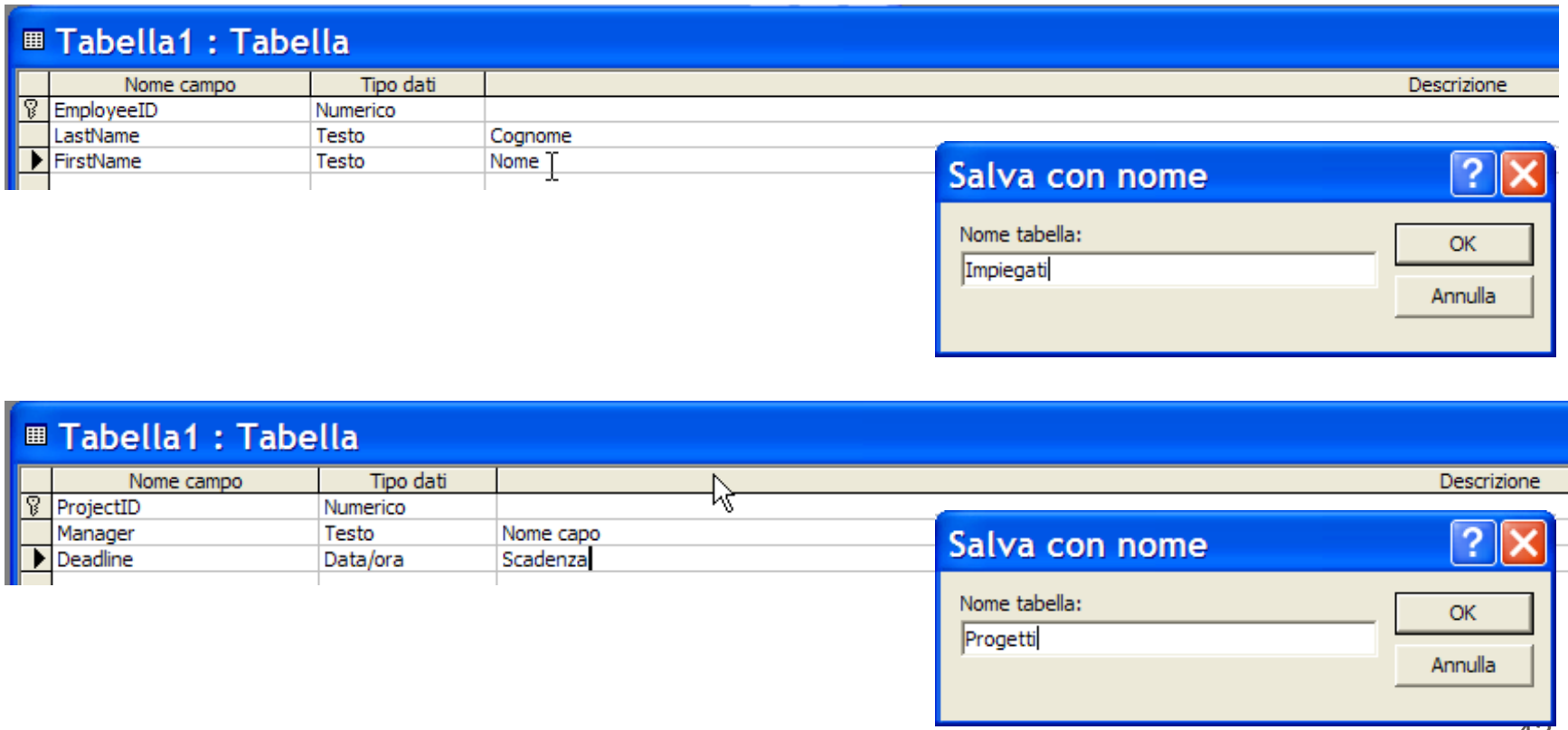
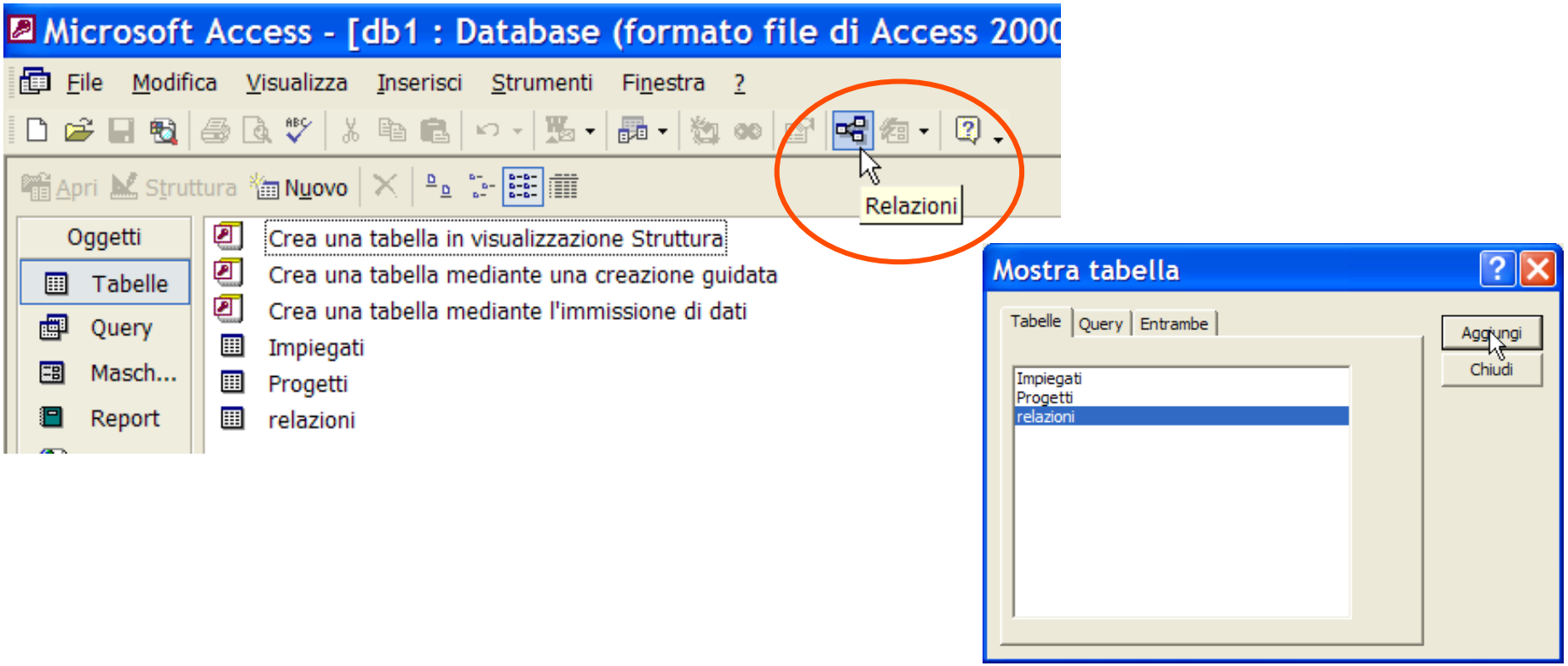


Tabella1 : Tabella
Nome campo
Tipo dati
EmployeeID
Numerico
Chiave primaria
Taglia
Copia
Incolla
Inserisci righe
Elimina righe
Genera...
Proprietà

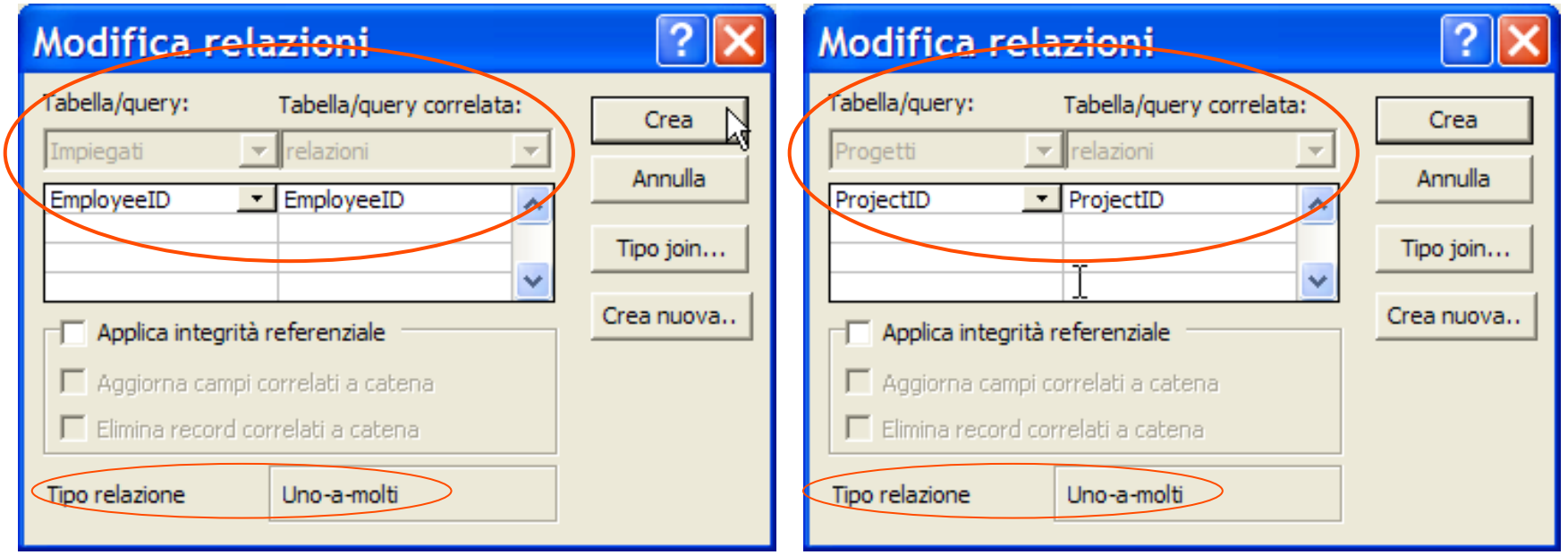
- Es.: creazione tabelle dell'esempio precedente: impiegati, progetti e tabella relazione tra impiegati e progetti

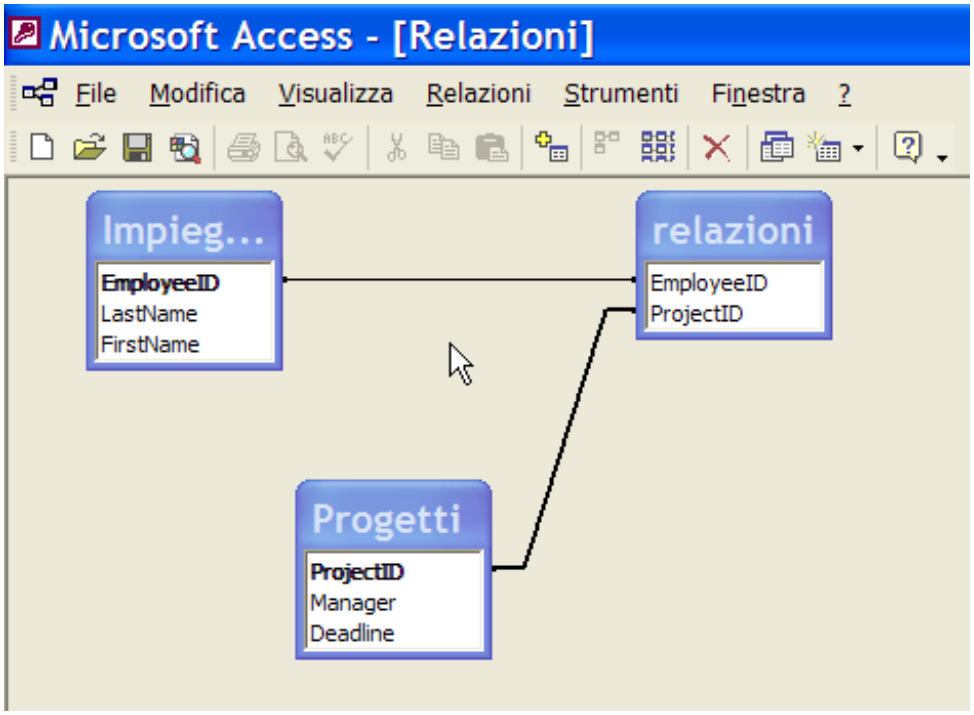


- Dopo aver creato la tabella delle relazioni si deve creare la relazione vera e propria che mette in collegamento i campi

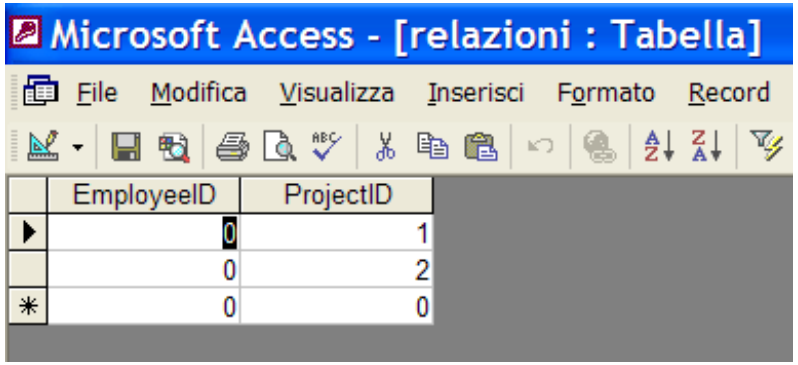


- Si trascinano i campi dalle tabelle degli impiegati e progetti ai rispettivi campi della tabella di relazione

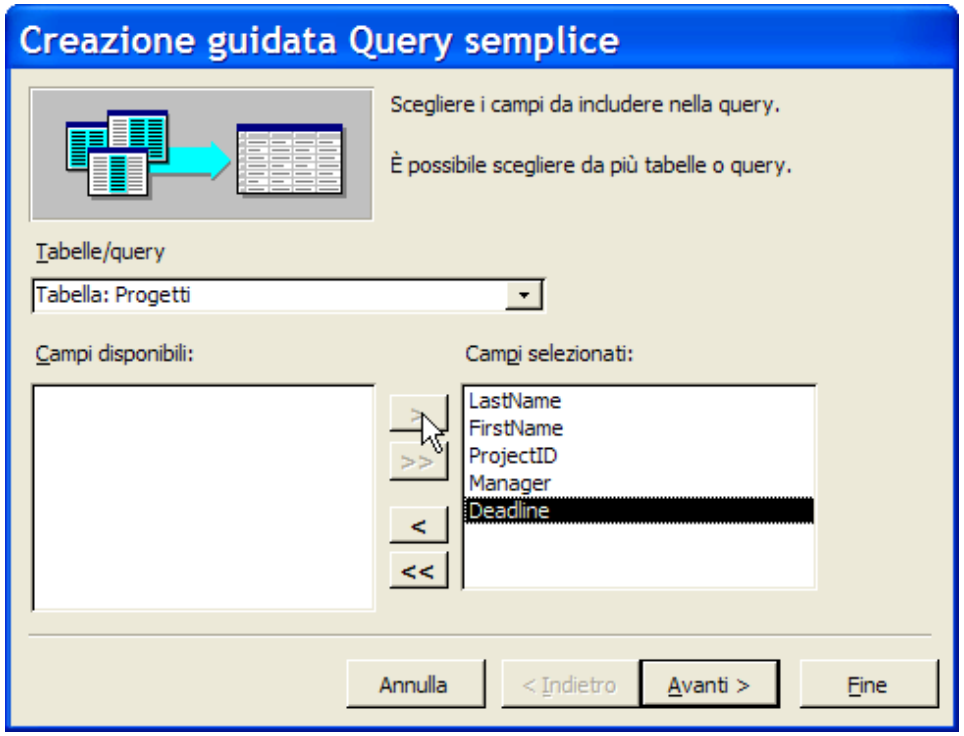


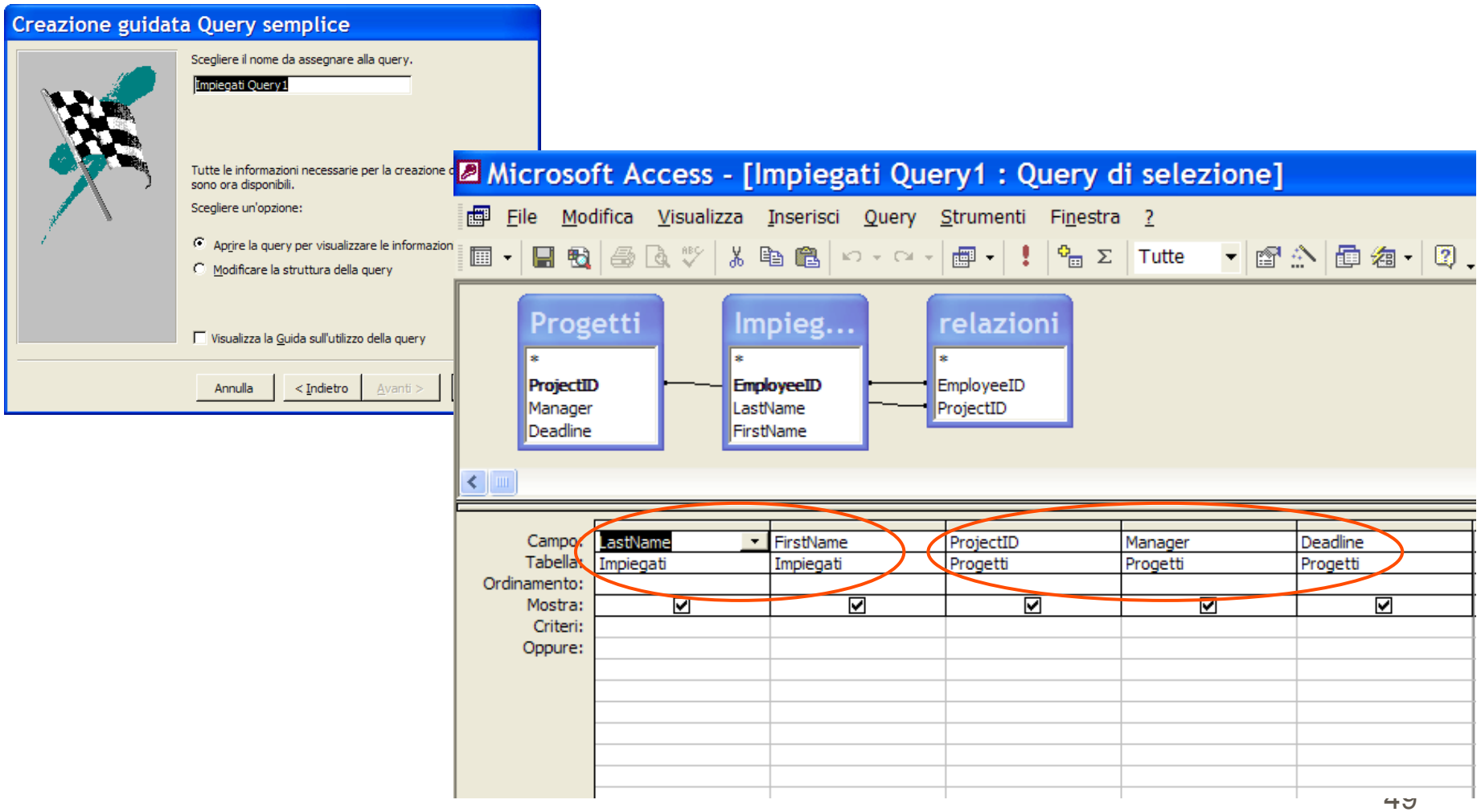

Microsoft Access - [Relazioni]
File Modifica Visualizza Relazioni Strumenti Finestra ?
Impieg...
EmployeeID
LastName
FirstName
relazioni
EmployeeID
ProjectID
Progetti
ProjectID
Manager
Deadline

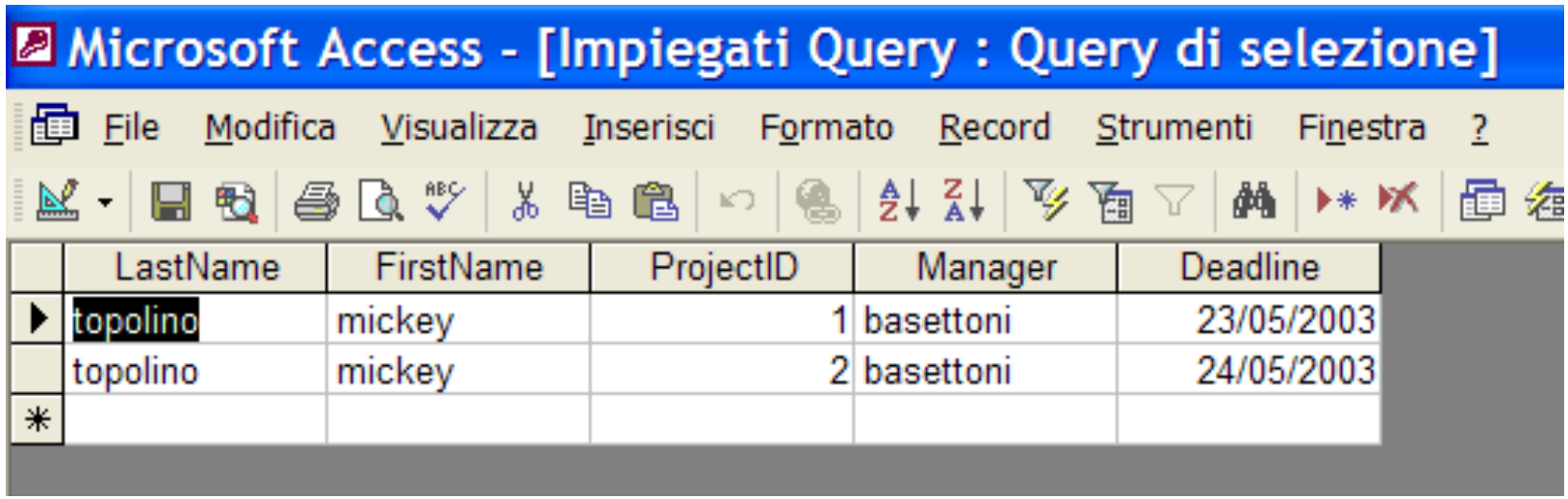
- Per fare una prova si immettono dei dati nelle tabelle di impiegati e progetti, e poi nella tabella relazioni



- Per vedere il risultato dobbiamo fare una query: selezioniamo tabelle e campi che vogliamo esaminare




Creazione guidata Query semplice
Scegliere il nome da assegnare alla query.
Impiegati Query1
Tutte le informazioni necessarie per la creazione d
sono ora disponibili.
Scegliere un'opzione:
Aprire la query per visualizzare le informazion
Modificare la struttura della query
Visualizza la Guida sull'utilizzo della query
Annulla
< Indietro
Avanti >
Microsoft Access - [Impiegati Query1 : Query di selezione]
File Modifica Visualizza Inserisci Query Strumenti Finestra ?
Tutte
Progetti
*
ProjectID
Manager
Deadline
Impieg...
*
EmployeeID
LastName
FirstName
relazioni
*
EmployeeID
ProjectID
Campo:
Tabella:
Ordinamento:
Mostra:
Criteri:
Oppure:
LastName
FirstName
ProjectID
Manager
Deadline
Impiegati
Impiegati
Progetti
Progetti
Progetti


Microsoft Access - [Impiegati Query : Query di selezione]
File Modifica Visualizza Inserisci Formato Record Strumenti Finestra ?
LastName FirstName ProjectID Manager Deadline
topolino mickey 1 basettoni 23/05/2003
topolino mickey 2 basettoni 24/05/2003

Criteri di progetto di un database

- I moderni software permettono di costruire in modo semplice un database, ma forniscono uno scarso supporto nel processo di progetto.
- Infatti, il processo di progetto dovrebbe essere una fase completamente indipendente dalla scelta del particolare software realizzativo.
- Spesso si è vincolati ad un particolare software; tuttavia delle buone scelte di progetto dovrebbero venire prima.

Criteri di progetto di un database

- I passi base del processo di progetto sono:
  - 1. definizione del problema e degli obiettivi
  - 2. valutazione della soluzione corrente (se esiste)
  - 3. progetto delle strutture dati
  - 4. costruzione delle relazioni
  - 5. implementazione di regole e vincoli
  - 6. creazione di viste
  - 7. implementazione del progetto.
- Notare che l'implementazione con un particolare software è il passo finale. I passi precedenti sono indipendenti dalla realizzazione.

Criteri di progetto: 1. definizione del problema

- Il primo passo nel progetto di un database è quello di delineare chiaramente la natura dei dati che è necessario memorizzare, senza considerare le interrogazioni che verranno usate sul database per trasformare i dati memorizzati in informazioni.

- Questo perché le domande possono cambiare nel tempo ed un database progettato solo per rispondere a domande specifiche diventerebbe inutile.

Criteri di progetto:
1. definizione del problema

- Il database deve essere pensato in modo da organizzare tutti i dati che sono utilizzati per risolvere un particolare problema o raggiungere un obiettivo,
  in modo che sia possibile rispondere ad ogni tipo di domanda su quei dati.

Criteri di progetto: 2. valutazione della soluzione corrente

- Nella maggior parte delle situazioni di progetto, qualche tipo di database esiste già, sia esso in forma cartacea o digitale.
- Spesso rappresenta un punto di partenza per conoscere i dati che sono attualmente di interesse.
- Nel caso si usi Access è bene esaminare i database di esempio

Criteri di progetto:
3. progetto delle strutture dati

- Un database è essenzialmente una collezione di tabelle che devono essere definite in base ai dati da rappresentare.

- Ogni tabella dovrebbe rappresentare un soggetto o un oggetto fisico distinto.

- Di solito, il modo migliore per determinare i dati che appartengono ad una stessa tabella è quello di fare una lista di tutti i campi di interesse per il problema in esame e determinare i raggruppamenti logici tra questi.

Criteri di progetto:
3. progetto delle strutture dati

Subjects
Buildings
Classrooms
Courses
Employees
Faculty Members
Part-time Employee
Staff

Fields
Address
Course Description
Course Name
Course Number
Courses Taught
Date Hired
Funding Source
Home Department
Hourly Rate
Meeting Times
Name
Number of Seats
Phone Number
Prerequisites
Salary
Social Security Number
Tenure Status
……

Criteri di progetto:
3. progetto delle strutture dati

- Es. di campi individuati per un sistema di pagamento per l'Università.
  - Nella lista dei campi alcuni fanno chiaramente riferimento agli impiegati, come Name, Address, Phone Number, Social Security Number, e Date Hired.
  - Altri campi come Courses Taught, Salary e Tenure Status sono parte di una struttura dati separata.
  - I campi Hourly Rate, Founding Source e Home Department sono parte di una struttura per i Part-time Employees.
- In questo modo è possibile raggruppare i campi secondo le entità cui appartengono.

Criteri di progetto:
3. progetto delle strutture dati

- Una volta che le tabelle sono state determinate e i campi sono loro assegnati, il passo successivo è la specifica dei campi.

- Un campo perfetto dovrebbe essere:
  - unico tra tutte le tabelle nel database, a meno che non sia usato come chiave;
  - dovrebbe contenere un singolo valore;
  - non dovrebbe essere possibile scomporlo in sottoparti componenti più piccole.

Criteri di progetto:
3. progetto delle strutture dati

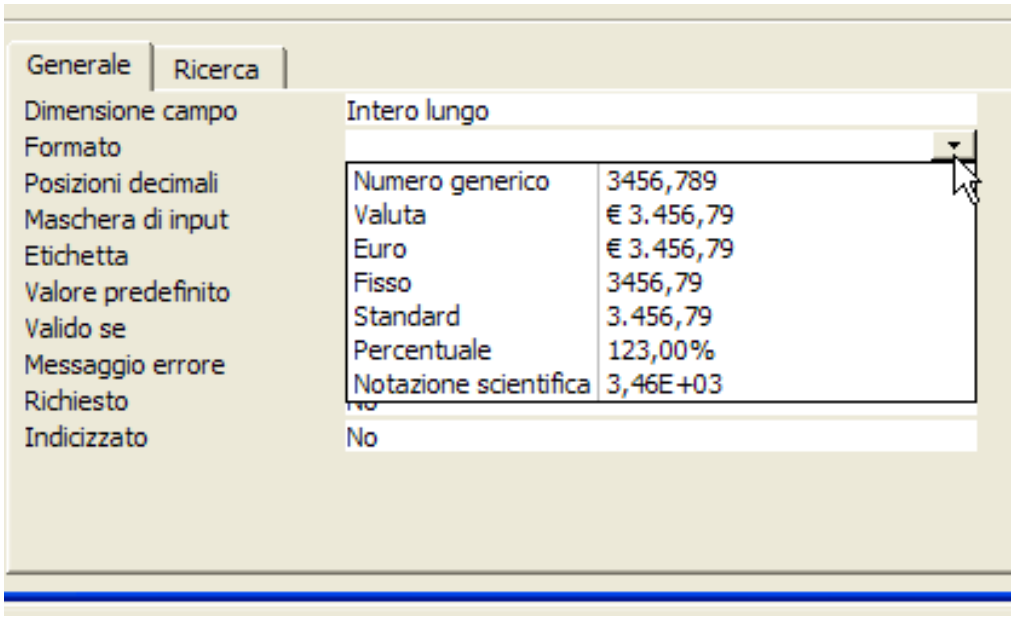
- Deve essere stabilito anche il tipo di dati che devono essere posti in ciascun campo.

- Esistono cinque tipi base per i campi (possono essere ulteriormente suddivisi per incrementare l'efficienza di memorizzazione e di elaborazione):
  - Numerico - è un qualunque valore numerico utilizzabile in operazioni di tipo matematico. È di solito possibile specificare campi numerici interi e reali (float).
  - Booleano - assume solo due valori si/no, vero/falso.

Criteri di progetto:
3. progetto delle strutture dati

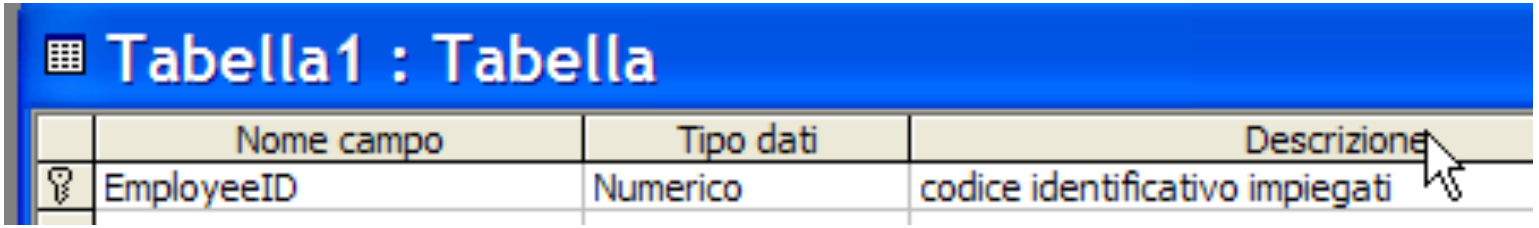
- Ora/Data - memorizza ora e/o data usando vari formati (europeo o americano, conversione a 12-24 ore, etc.).
- Testuale - rappresenta virtualmente qualunque altro tipo di dato. Di solito esistono campi a lunghezza fissa, variabile o campi speciali per grandi sezioni di testo.
- Binario - alcuni database possono memorizzare oggetti in forma binaria in un campo. Es. documento, file grafico, campione di suono, o altro. Sono campi essenzialmente contenitori di dati. Su questi campi non possono essere eseguite operazioni di ordinamento o ricerca.

- Es. in Access, tipi di dati e formato per dato numerico



Criteri di progetto:
3. progetto delle strutture dati

- Un ulteriore aspetto di interesse riguarda la nomenclatura da adottare nel nominare tabelle e campi. È di aiuto nella comunicazione e quando il database cresce in complessità e dimensione.



- Alcuni suggerimenti:
  - usare nomi descrittivi che riflettono il soggetto della tabella o i dati;
  - non usare nomi specifici per dati che potrebbero assumere una forma più generale;
  - evitare abbreviazioni e acronimi;
  - non usare punteggiatura o spazi nei nomi;
  - usare nomi delle tabelle plurali, nomi dei campi singolari;
  - tabelle di relazione hanno di solito come nome la combinazione dei nomi delle tabelle che collegano.

Criteri di progetto:
4. costruzione delle relazioni

- Un prerequisito alla costruzione delle relazioni è garantire che ogni tabella abbia una chiave univoca per identificare i singoli records della tabella.

- Qualunque campo esistente contenente un valore univoco è un candidato accettabile per essere utilizzato come chiave.

Criteri di progetto:
4. costruzione delle relazioni

- Una soluzione migliore è quella di aggiungere un campo arbitrario ad ogni tabella con il compito specifico di fungere da campo chiave.
- Tale campo deve contenere un valore univoco, anche senza un particolare significato per i dati contenuti nella tabella.
  - Il valore della chiave è tipicamente un intero assegnato ad ogni record inserito nella tabella e mai ripetuto.
- Le relazioni tra tabelle possono essere costruite usando le chiavi delle singole tabelle.

Criteri di progetto:
4. costruzione delle relazioni

- Per una relazione uno-a-uno, la chiave primaria della tabella principale è inserita nella sottotabella:
  - la sottotabella utilizza la chiave esterna come chiave principale e non necessita di una propria chiave primaria in quanto gli ingressi nelle due tabelle sono univocamente collegati gli uni agli altri.

EmployeeID	LastName	FirstName	Address	Phone
77620	Southworth	Jerry	496 Laguardia Place	555-2222
77621	Talbot	Gina	426 Lake St	555-8267
77622	Russell	David	318 W 39 Th St	555-4445
77623	Thompson	Linda	2427 Broadway	555-1568
77624	Albertson	Steve	224 Lawrence Ave	555-9872
77625	Davis	Russell	224 Lawrence Ave	555-1213

EmployeeID	Salary	401(k)	Health
77621	41,000	5	BC
77622	55,755	2	BC
77623	22,155	N/A	QS
77624	18,975	2	N/A

Criteri di progetto:
4. costruzione delle relazioni

- In una relazione uno-a-molti, la chiave primaria della tabella con la relazione “uno”, è usata per identificare i record collegati nella tabella “molti:
  - il collegamento è fatto inserendo la chiave primaria della tabella “uno” come nuovo campo nella tabella “molti” dove rappresenta una chiave esterna.

OrderID	CustID	Name	Shipping Address
23455	1521	Velio Cooper	152 W Waterloo St Austin, TX 78752
23457	1567	Francis Cioni	1445 E Delavan Ave Laredo, TX 78043
23458	1765	Jane Carroll	1551 Westwood Blvd Ocilla, GA 31774
23459	0021	Jody Hawes	2664 Woodhill Rd Bronx, NY
23460	1235	David Hill	742 Hearne Cleveland, OH 44104

OrderID	ItemCode	Quantity	Price
23455	ZD-552	1	8.95
23457	XT-211	2	16.50
23457	XT-212	2	22.00
23458	CC-48	15	225.15
23459	A-567	1	19.99
23460	XD-234	1	49.95

Criteri di progetto:
4. costruzione delle relazioni

- Una relazione molti-a-molti è la più complessa da costruire. Richiede una tabella aggiuntiva per rappresentare la relazione:
  - la tabella di relazione è creata riportando in essa le chiavi di una tabella con tutte le chiavi della seconda tabella che sono ad essa associate.

EmployeeID	LastName	FirstName
77620	Southworth	Jerry
77621	Talbot	Gina
77622	Russell	David
77623	Thompson	Linda
77624	Albertson	Steve
77625	Davis	Russell

ProjectID	Manager	Deadline
22356	Jones	4/9
27685	Lewis	12/22
33564	Albertson	8/17

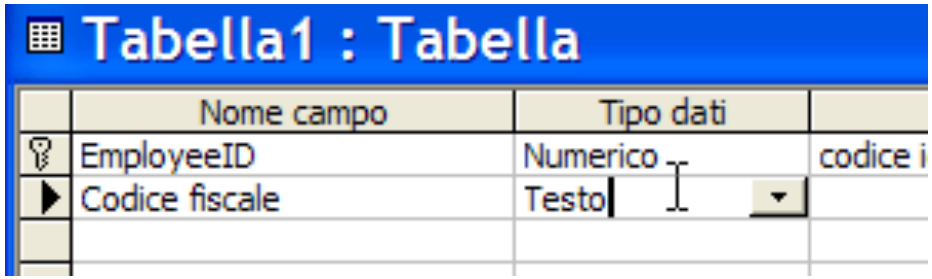
EmployeeID	ProjectID
77620	22356
77621	27685
77624	27685
77622	33564
77624	33564
77622	22356

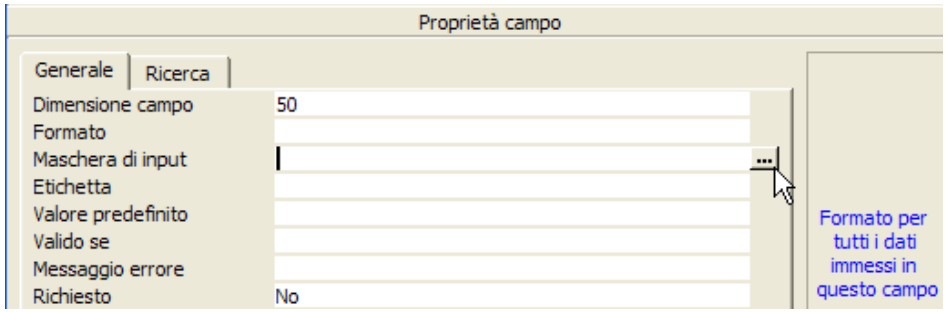
Criteri di progetto:
5. implementazione di regole e vincoli

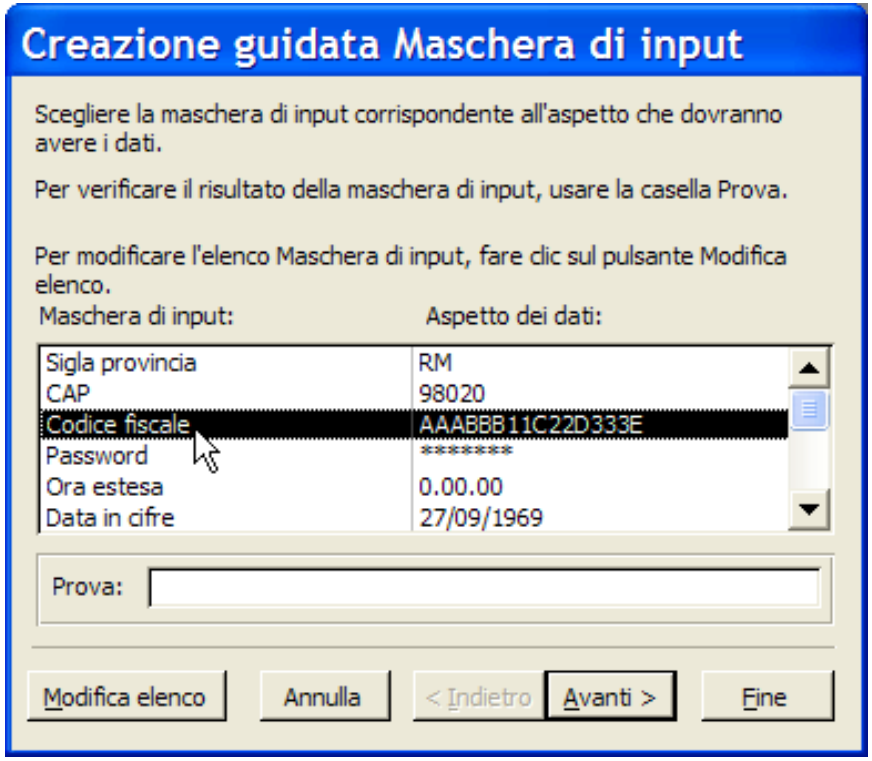
- Regole e vincoli conducono di norma ad una maggiore “pulizia” dei dati e a migliori informazioni nel loro uso.

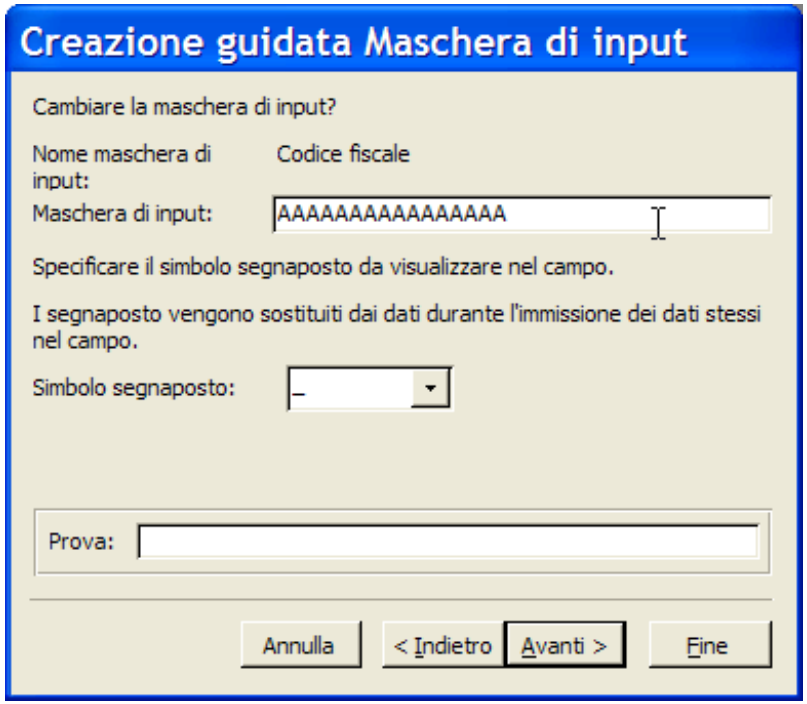
- Alcuni vincoli sono tipicamente imposti dalla natura stessa dei dati.
  - Es.: un campo per il codice fiscale può essere dimensionato per contenere sempre 16 caratteri. Assicura che il dato sia corretto ed accurato. Il sistema può impedire l’inserimento di un codice con un numero diverso di caratteri (in difetto o in eccesso).
  - Altri dati possono assumere valori solo in uno specifico intervallo. Il sistema può operare un controllo sulla validità dei dati in ingresso.

- In Access si specificano i vincoli nelle proprietà del campo









Definizione della maschera di input	Esempi di valori
(000) 000-0000	(206) 555-0248
(999) 999-9999!	(206) 555-0248 ( ) 555-0248
(000) AAA-AAAA	(206) 555-TELE

Criteri di progetto: 6. creazione di viste

- Consente di trasformare i dati del database in informazioni utili all'utente.

- Le viste sono semplicemente collezioni di dati resi accessibili in un certo modo. Una vista potrebbe essere un sottoinsieme dei dati delle tabelle.
  - Ad esempio, nel database degli impiegati il nome ed il numero di telefono potrebbero essere una vista per una interrogazione che ricerca il numero di telefono di un impiegato. Risponde alla domanda senza includere informazione inutile come altri dati personali o privati.
  - In altri casi una vista può raccogliere dati da più tabelle e condensarli insieme. Ad esempio i dati delle tabelle Corsi e Studenti potrebbero essere usati per fornire una lista aggiornata degli studenti in ciascun corso.

Criteri di progetto: 7. implementazione del progetto

- Il progetto è condotto fino a questo punto senza considerare il particolare software da utilizzare per realizzare il database.
- Il criterio di base che guida la scelta è la possibilità di realizzare con una data soluzione software il database di cui si ha bisogno.
  - Alcune tra le scelte possibili sono:
    - Microsoft Access, lo standard di fatto per Windows;
    - Microsoft SQL Server, database professionale;
    - MySQL, prodotto free molto diffuso per applicazioni sul Web;
    - Oracle, il leader in ambiente Unix. Richiede hardware e supporto costosi.

Normalizzazione

- In qualunque libro che affronti il progetto di database dal punto di vista teorico viene sempre riservata una certa enfasi al problema della normalizzazione o decomposizione del database.

- In sostanza, la normalizzazione è un modo di analizzare e migliorare la stabilità ed integrità di un insieme di dati relazionali.
  - Previene la possibilità di inconsistenze nei dati conseguenza di una cattiva progettazione.

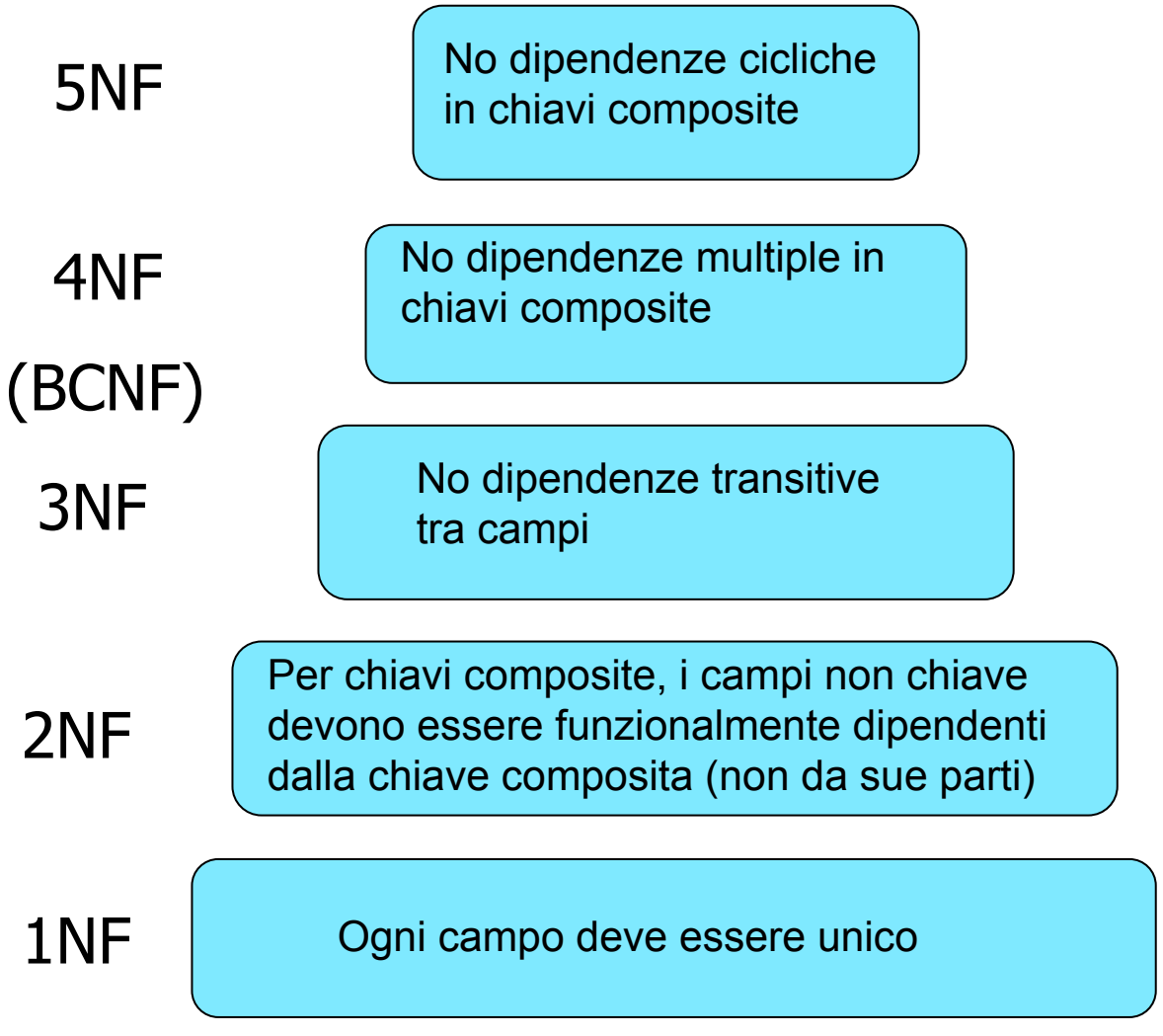
Normalizzazione

- Il processo di progettazione discusso è tale da verificare i requisiti di normalizzazione per il progetto di database tipici.

- Il processo di normalizzazione è composto da più passi, ognuno dei quali rappresenta una forma normale (NF).

- Esistono 5 forme normali che possono essere applicate ad un database, ma di solito il progettista può accontentarsi di realizzare le prime tre (quindi il processo può arrestarsi alla 3NF).

Processo di Normalizzazione



- Un database deve passare attraverso le normalizzazioni in sequenza:
  - un database in 3NF deve soddisfare anche i requisiti della 1NF e 2NF.

Prima forma normale (1NF)

- Un database che soddisfa la 1NF ha una chiave primaria e non contiene campi dati ripetuti o gruppi di campi (si intende che queste proprietà devono essere soddisfatte da tutte le tabelle del database).
  - Questo significa che tutti i record di una tabella nel database contengono lo stesso numero di campi dati distinti.

- Garantisce che ciascun record possa essere identificato in modo univoco, ed evita la duplicazione di campi base.
  - Tutti i database visti soddisfano questa condizione.

Prima forma normale (1NF)

- Le tabelle non soddisfano la 1NF.

Student	Album
StudentID (key)	AlbumID (Key)
Name	Artist
Year	Title
Period 1 class	Track 1 name
Period 1 teacher	Track 1 duration
Period 1 grade	Track 2 name
Period 2 class	Track 2 duration
Period 2 teacher	Track 3 name
Period 2 grade	Track 3 duration
Period 3 class	Track 4 name
Period 3 teacher	Track 4 duration
Period 3 grade	(etc.)
(etc.)	

Prima forma normale (1NF)

- Nell'esempio precedente la tabella Student ha una chiave univoca, ma è chiaramente composta da due entità distinte e dovrebbe pertanto essere divisa in due tabelle in relazione tra loro:
  - una tabella di informazioni relative agli studenti;
  - una tabella dei corsi.
- Una relazione molti a molti dovrebbe rappresentare i corsi seguiti dallo studente.
- Nella forma attuale si potrebbe avere un numero di campi per i corsi grande o piccolo rispetto a quelli effettivamente seguiti dallo studente. Lo stesso limite esiste per la tabella degli album.

Seconda e Terza forma normale (2NF, 3NF)

- La 2NF si applica solo a database con chiavi composite. Una chiave composita è costituita da informazioni di due o più entità logiche distinte (chiave costituita da più campi).
  Di conseguenza:
  - una tabella la cui chiave non è composita e che verifica la 1NF verifica anche la 2NF.
- Nella maggior parte dei casi progettare il database in modo da verificare la 2NF determina che anche la 3NF sia verificata.
  - È la forma di normalizzazione finale applicata a database relazionali nella pratica di impiego comune.

Seconda forma normale (2NF)

Item

Item number (component of key)

Vendor name (component of key)

Product name

Product description

Vendor contact name

Vendor address

Vendor phone number

Unit price

Unit cost

La tabella non soddisfa la 2NF

Seconda forma normale (2NF)

- La tabella dell'esempio precedente non verifica i requisiti della 2NF.
- La chiave composita è costituita da informazioni che riguardano due distinte entità logiche, l'item ed il vendor:
  - il problema di questa soluzione è che quando il Vendor contact name è cambiato, tutti i record che contengono quella informazione devono essere cambiati altrimenti il database conterrà informazione errata;
  - infatti, i record di questa tabella hanno alcuni campi (Vendor contact name, Vendor address, Vendor phone number) i cui valori dipendono dal valore di un componente della chiave composita (Vendor name).

Seconda forma normale (2NF)

- Il processo di progetto definito in precedenza evitava questi problemi attraverso due accorgimenti:
  - evitare l'uso di chiavi composite;
  - entità logiche distinte, siano esse oggetti od eventi, richiedono tabelle dei dati distinte.

Terza forma normale (3NF)

- Richiede che non esistano dipendenze transitive tra campi nelle tabelle dei dati:
  - una dipendenza transitiva si presenta quando un campo che non è la chiave primaria della tabella agisce come se fosse un'altra chiave primaria per parte o tutti i campi del record.
- La tabella dell'esempio visto per la 2NF contiene sia dipendenze transitive che funzionali, poiché i campi Vendor name e Item number determinano ciascuno i valori di diversi record nella tabella.
- Nella maggior parte dei casi riprogettare il database in modo da verificare la 2NF determina che anche la 3NF sia verificata.

Normalizzazione

- Riassumendo è possibile dire che i processi di base per verificare la 1NF, 2NF e 3NF, si possono tutti ricondurre ad un unico criterio:
  - garantire che entità logiche distinte siano memorizzate in tabelle differenti.

- Come si è visto è possibile arrivare ad una buona progettazione semplicemente seguendo alcune regole di base.

ESEMPI DI PROGETTO

Esercizio di progetto - I

- 1. Definizione ed obiettivi.
  - Personal Information Manager (PIM) - il principale obiettivo di questo database è di mantenere ed organizzare i dati per contattare le persone coinvolte nella vita personale e professionale.
  - Per il momento la definizione non include il tipo particolare di dati da immagazzinare nel database, né il tipo di interrogazioni eseguite sui dati.
- 2. Ricerca del database corrente.
  - Comprende la ricerca del modo in cui i dati sono attualmente raccolti e come quei dati sono trasformati in informazioni utili.

Esercizio di progetto - I

- 3. Progetto delle strutture dati.
  - Assumiamo che il database contenga una sola tabella (Contacts) sia per contatti che riguardano il lavoro che privati. Una soluzione alternativa potrebbe essere quella di avere due tabelle separate per questi due tipi di contatti.
  - Comunque il modo usuale di fare delle interrogazioni è quello di usare il cognome indipendentemente dal fatto che il soggetto cercato appartenga ai contatti di lavoro o privati.
- 4. Costruzione delle relazioni.
  - Esiste una sola tabella e quindi non ci sono relazioni.

Esercizio di progetto - I

Fieldname	Notes
ID	Campo numerico usato come chiave
LastName	Limite a 40 caratteri
FirstName	Limite a 20 caratteri
MailingAddress	Campo testo a lunghezza variabile
City	Limite a 30 caratteri
State	Limite a 2 caratteri
Zip	10 cifre
Country	Limite a 20 caratteri
Directions	Campo testo
HomePhone	Memorizzato come campo testo
WorkPhone	Memorizzato come campo testo
Fax	Memorizzato come campo testo
Email	Limitato a 40 caratteri
WebPage	Home page URL
Birthday	Data in formato mm/dd/yyyy

Esercizio di progetto - I

- 5. Regole e vincoli.
  - Sono usate per garantire che le relazioni tra tabelle siano valide e che i dati in ingresso alle tabelle siano privi di errori.
  - In questo caso non ci sono relazioni, ma la chiave primaria deve essere fissata. La sola regola necessaria è che ciascun record sia individuato da un numero intero univoco. Pertanto si assegna 1 al primo record e si incrementa il valore di uno per ogni record successivo.
  - Si potrebbero definire dei vincoli al formato dei numeri telefonici per garantire una maggiore accuratezza.

Esercizio di progetto - I

- 6. Creazione di viste e reports.
  - Una vista potrebbe essere quella per una query tipica che ricerca nome, indirizzo di posta elettronica e numero(i) di telefono. L'altra vista necessaria è quella che fornisce tutto il record del database.
  - Un report utile è quello che presenta la lista ordinata alfabeticamente di tutti i contatti con numeri di telefono ed indirizzi. Un altro potrebbe essere una lista mensile di tutti i compleanni, o solo per il mese corrente.
- 7. Implementazione del progetto.
  - Realizzazione del database usando Access

Esercizio di progetto - I

- Due possibili alternative alla soluzione vista.
  - Due tabelle distinte. Si potrebbero avere due tabelle separate, una per i contatti personali e una per quelli di lavoro.
    Un vantaggio di questa scelta è che i campi relativi ad un solo tipo di contatto devono essere posti solo nella tabella corrispondente. Ad esempio il campo compleanno potrebbe essere rilevante solo per i contatti personali. Le queries potrebbero essere più efficienti. Potrebbe però generare problemi se un contatto viene inserito in entrambe le tabelle

Esercizio di progetto - I

- Tabella primaria con due sottotabelle. Un progetto più relazionale potrebbe consistere di una tabella primaria con tutte le informazioni costanti sull'individuo (nome, data di nascita, etc.) e due tabelle addizionali che mantengano i contatti personali e di lavoro. Queste sottotabelle addizionali sono collegate a quella principale da una relazione uno-a-uno, ottenuta semplicemente attraverso la chiave primaria della tabella principale. Questa, dal punto di vista del progetto, è la soluzione più corretta.

Esercizio di progetto - II

- 1. Definizione ed obiettivi.
  - Media File Database - l'obiettivo di questo database è di consentire la gestione di diversi media files e di permettere il loro uso per costruire in modo dinamico il contenuto di un sito Web.
- 2. Ricerca del database corrente.
  - In questo caso il database corrente è quello costituito dalle informazioni che riteniamo utili. Inoltre è necessario capire come il database verrà utilizzato. Un modo ovvio è di cercare media files rapidamente e facilmente. Un altro possibile utilizzo è la pubblicazione dinamica del contenuto sul Web. Es. per creare un catalogo online per la vendita.

Esercizio di progetto - II

Fieldname	Notes
Filename	Nome del media file
Filesize	Può variare da pochi KB a molti MB
File type or format	Deve essere flessibile
Image dimensions	Dimensione in pixels
Color depth	In bit da 1 a 32
Compression level	Alcuni formati hanno diversi livelli di compressione
Scanned resolution	Immagini scannerizzate a diversi livelli
File manupulations	Storia di come il file è stato editato
Last modification	Data di ultima modifica
Sampling rate	Per audio files. Frequenza di campionamento
Video frame rate	Per i video sono possibili diversi frame rate
Codec used	Video compressi con formati proprietari
Original source	Book, CD, videotape o altre sorgenti
Descriptive text	Descrive il tipo di media presente
Keywords	Usate per una ricerca rapida (es. albero, sinfonia, etc.)
Length of sound clip	Lunghezza del file audio
Plug-in location	Sito Web da cui scaricare il plug-in se necessario
Video duration	Durata del file video

Esercizio di progetto - II

- 3. Progetto delle strutture dati.
  - In questo esempio è possibile individuare più oggetti: i files video, audio e di immagini. In aggiunta è possibile considerare un oggetto corrispondente a files di testo con le informazioni descrittive utili per realizzare una pubblicazione dinamica sul Web.
    Una tabella può essere impiegata per mantenere la descrizione testuale e il nucleo di informazioni comuni ai vari files (es. nome, dimensione e lista di keywords). Questa tabella è collegata a quella dei singoli media, che ne riportano le informazioni specifiche (una tabella separata per audio, video ed immagini).

Esercizio di progetto - II

Files Data Table

Fieldname	Notes
Filename	Nome del media file
Filesize	Può variare da pochi KB a molti MB
FileType	Deve essere flessibile
FileChanges	Storia di come il file è stato editato
LastModification	Data di ultima modifica
DescriptiveText	Descrive il tipo di media presente
FileKeywords	Usate per una ricerca rapida (es. albero, sinfonia, etc.)

Images Data Table

Fieldname	Notes
ImageHeight	In pixels
Image Width	In pixels
ImageColor depth	In bit da 1 a 32
ImageCompressionLevel	Alcuni formati hanno diversi livelli di compressione
ImageScannedResolution	Immagini scannerizzate a diversi livelli

Esercizio di progetto - II

Sounds Data Table

Fieldname	Notes
SoundSamplingRate	Audio files possono essere campionati a diversi rates
SoundClipLength	Lunghezza del file audio

Videos Data Table

Fieldname	Notes
Height	In pixels
Width	In pixels
VideoColorDepth	In bit da 1 a 32
VideoCompressionLevel	Alcuni formati hanno diversi livelli di compressione
VideoSamplingRate	Campionamento per il video
FrameRate	Per i video sono possibili diversi frame rate
Codec	Video compressi con formati proprietari
VideoLength	Lunghezza del file video

Esercizio di progetto - II

- Possiamo considerare altre due tabelle utili.
  Dato che una particolare sorgente può fornire un numero di media diversi, è meglio utilizzare una tabella separata per questa informazione. Le sorgenti possono essere compagnie o singoli individui; altre sorgenti sono libri, riviste o altri media tradizionali digitalizzati.
  L'ultima tabella contiene l'insieme di progetti in cui questi media files sono attualmente utilizzati.

Esercizio di progetto - II

Sources Data Table

Fieldname	Notes
OriginalSource	Book, CD, videotape o altre sorgenti
SourceNotes	Descrive il tipo di media presente
SourceKeywords	Usate per la sorgente

Projects Data Table

Fieldname	Notes
ProjectName	Nome del progetto
Contact	Persona da contattare per il progetto
ProjectsNotes	Note sul progetto

Esercizio di progetto - II

- 4. Costruzione delle relazioni.
  - Il primo problema è la definizione delle chiavi per ogni tabella.
    Le chiavi sono poi utilizzate per stabilire le relazioni.
    - Per la tabella dei Files il campo FileName può contenere duplicati. Come chiave primaria si assegna un intero in un campo FileID.
    - La tabella dei Files è legata alle tabelle Images, Videos e Sounds da relazioni uno-a-uno, in quanto ciascuna di queste rappresenta una sottotabella per la tabella dei Files. Per creare la relazione è sufficiente porre la chiave FileID della tabella Files nelle sottotabelle come chiave esterna. Le sottotabelle possono usare la chiave esterna come chiave primaria, in quanto consente di identificare in modo univoco ciascun record.

Esercizio di progetto - II

FileID	FileType	FileSize	Keywords	Description
101	JPG	427	Bullet,square	Homepage bullet
102	AVI	1592031	Plane	Flying plane
103	GIF	85	Line, red	Red rule
104	JPG	5627	Canyon	Copper canyon
105	WAV	13220	Ding	Doorbell
106	MOV	22843612	Cry, baby	Crying baby boy

Files Table

FileID	Height	Width
101	20	80
103	10	60
104	110	75

Images Table

FileID	Height	Width	Length
102	400	300	26.1
106	400	300	240.0

Videos Table

FileID	SoundSamplingRate	SoundClipLength
105	44.1	30

Sounds Table

Esercizio di progetto - II

- La relazione tra la tabella Files e la tabella Sources è invece una relazione uno-a-molti tra Sources (uno) e Files (molti). Infatti, ciascun record nella tabella Files è legato ad un solo record della tabella Sources (ogni file può avere una sola sorgente), mentre ogni record in Sources può essere legato ad uno o più record in Files (ogni sorgente può produrre più file diversi).
  È quindi necessario inserire la chiave primaria della tabella "uno" come chiave esterna nella tabella "molti".
  Alla tabella Sources è assegnata come chiave primaria un intero identificato con il campo SourceID, e questa chiave è aggiunta anche alla tabella Files dove rappresenta una chiave esterna.

Esercizio di progetto - II

FileID	SourceID	FileType	FileSize	Keywords	Description
101	001	JPG	427	Bullet,square	Homepage bullet
102	001	AVI	1592031	Plane	Flying plane
103	002	GIF	85	Line, red	Red rule
104	004	JPG	5627	Canyon	Copper canyon
105	001	WAV	13220	Ding	Doorbell
106	002	MOV	22843612	Cry, baby	Crying baby boy

Files Table

many

one

SourceID	Source	Copyright
001	Personal	Yes
002	BBC Library	No
003	Mega Studios	No
004	David Rides	Yes

Sources Table

Esercizio di progetto - II

- L'ultima relazione è tra la tabella Projects e la tabella Files.
  Un singolo progetto può comprendere un certo numero di files, ed un singolo file può essere parte di più progetti. La relazione tra le due tabelle è quindi molti-a-molti.
  Alla tabella progetti è assegnata una chiave primaria ProjectID costituita da un numero intero. La relazione è stabilita attraverso una nuova tabella in cui sono riportate le chiavi primarie dalle tabelle Files
  e Projects per i records che devono essere posti in relazione.
  La tabella di collegamento è identificata con Projects_Files.

- Nella sua versione finale il database comprende quindi 7 tabelle, 3 relazioni uno-a-uno, 1 relazione uno-a-molti e 1 relazione molti-a-molti.

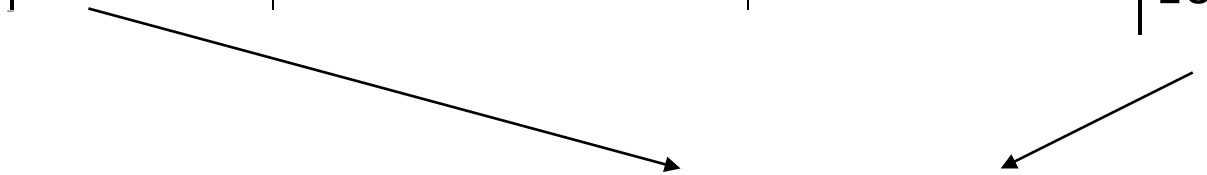
Esercizio di progetto - II

Projects Table

ProjectID	ProjectName	Cantact
2567	MegaSite	Dave Nelson
2568	Davis Presentation	Cindy Raye
2569	Monkey Movie	Nell Smith

FileID	SourceID	FileType
101	001	JPG
102	001	AVI
103	002	GIF
104	004	JPG
105	001	WAV
106	002	MOV

Files Table

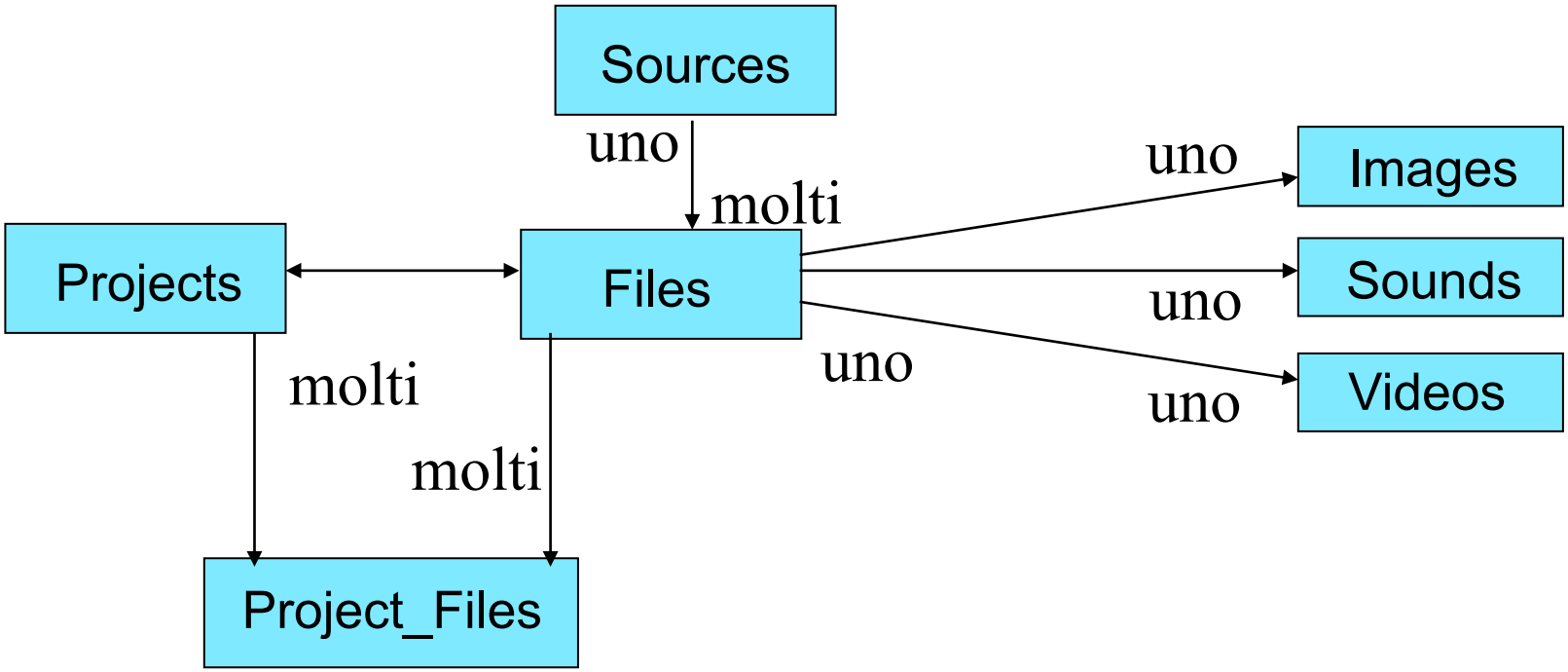


Projects_Files Table

ProjectID	FileID
2567	101
2569	102
2568	103
2567	104
2569	105
2569	106

Esercizio di progetto - II

- In uno schema entità-relazione il database può essere rappresentato nel modo seguente:



Esercizio di progetto - II

- Tenendo conto delle chiavi primarie e delle relazioni, le tabelle assumono la forma seguente:

Files Data Table

Fieldname	Notes
FileID	Chiave primaria
SourceID	Chiave esterna che collega alla Sources Data Table
FileName	Nome del media file
FileSize	Può variare da pochi KB a molti MB
FileType	Deve essere flessibile
FileChanges	Storia di come il file è stato editato
LastModified	Data di ultima modifica
Description	Descrive il tipo di media presente
FileKeywords	Usate per una ricerca rapida (es. albero, sinfonia, etc.)

Esercizio di progetto - II

Images Data Table

Fieldname	Notes
ImageID	Chiave esterna dalla Files Data Table
ImageHeight	In pixels
ImageWidth	In pixels
ImageColorDepth	In bit da 1 a 32
ImageCompressionLevel	Alcuni formati hanno diversi livelli di compressione
ImageScannedResolution	Immagini scannerizzate a diversi livelli

Sounds Data Table

Fieldname	Notes
SoundID	Chiave esterna dalla Files Data Table
SoundSamplingRate	Audio files possono essere campionati a diversi rates
SoundClipLength	Lunghezza del file audio

Esercizio di progetto - II

Videos Data Table

Fieldname	Notes
VideoID	Chiave esterna dalla Files Data Table
Height	In pixels
Width	In pixels
VideoColorDepth	In bit da 1 a 32
VideoCompressionLevel	Alcuni formati hanno diversi livelli di compressione
VideoSamplingRate	Campionamento per il video
FrameRate	Per i video sono possibili diversi frame rate
Codec	Video compressi con formati proprietari
VideoLength	Lunghezza del file video

Sources Data Table

Fieldname	Notes
SourceID	Chiave primaria
OriginalSource	Book, CD, videotape o altre sorgenti
SourceNotes	Descrive il tipo di media present
SourceKeywords	Usate per la sorgente

Esercizio di progetto - II

Projects Data Table

Fieldname	Notes
ProjectID	Chiave primaria
ProjectName	Nome del progetto
ProjectContact	Persona da contattare per il progetto
ProjectsNotes	Note sul progetto

Projects_Files Table

Fieldname	Notes
FileID	Chiave esterna dalla Files Data Table
ProjectID	Chiave esterna dalla Projects Data Table

Esercizio di progetto - II

- 5. Regole e vincoli.
  - Diversi campi per le sottotabelle Images, Sound e Videos, sono soggetti a vincoli. Ad esempio un file audio può essere campionato sia in modalità mono che stereo. La frequenza di campionamento è normalmente limitata ad un insieme di frequenze, che includono 11, 22, 32, 44.1 e 48KHz.
    Inoltre possono essere stabilite delle regole per i nomi dei files. Ad esempio tutti con lettere minuscole senza spaziature o altri caratteri di punteggiatura.

Esercizio di progetto - II

- 6. Creazione di viste e reports.
  - Le viste essenziali da creare sono quelle che combinano i dati dalla tabella dei Files con quelli dei singoli media. Le informazioni dai records collegati possono essere ottenute attraverso la chiave che definisce la relazione.
  - Un report di interesse potrebbe essere quello che produce la lista di files che verificano una qualche combinazione di keyword, size, type e/o filename. Ad esempio, tutte le immagini con dimensione maggiore di una soglia data. Oppure la lista dei files forniti da ciascuna sorgente. Infine, una lista dei file impiegati in ogni progetto.