**1. Nozioni fondamentali della progettazione di database**

http://office.microsoft.com/it-it/access/HA012242471040.aspx

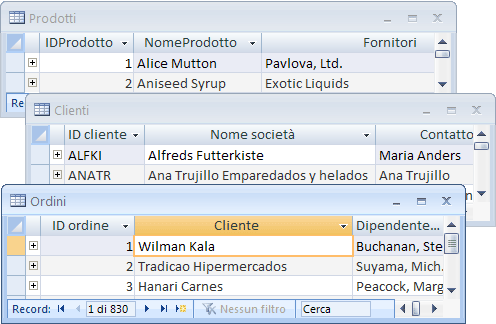
Un database progettato correttamente consente di accedere a informazioni aggiornate e accurate. Poiché una progettazione corretta è essenziale per raggiungere gli obiettivi desiderati utilizzando un database, è opportuno dedicare il tempo necessario all'apprendimento dei principi di una progettazione appropriata. Al termine, sarà molto più probabile disporre di un database in grado di soddisfare le esigenze e supportare facilmente eventuali modifiche.

In questo articolo vengono descritte le linee guida per la pianificazione di un database. Verrà illustrato come determinare le informazioni necessarie e come suddividere tali informazioni all'interno di tabelle e colonne appropriate. Verrà inoltre illustrato il modo in cui le tabelle sono correlate l'una all'altra. È consigliabile leggere questo articolo prima di creare il primo database.

Contenuto dell'articolo

1. Terminologia relativa ai database
2. Progettazione appropriata di un database
3. Processo di progettazione
4. Definizione dello scopo del database
5. Individuazione e organizzazione delle informazioni necessarie
6. Suddivisione delle informazioni in tabelle
7. Trasformazione degli elementi di informazioni in colonne
8. Specifica di chiavi primarie
9. Creazione di relazioni tra tabelle
10. Perfezionamento della struttura
11. Applicazione delle regole di normalizzazione
12. **Terminologia relativa ai database**

Le informazioni vengono organizzate da Microsoft Office Access 2007 in **tabelle**, ovvero elenchi di righe e colonne simili a un blocco per la contabilità o un foglio di lavoro di Microsoft Office Excel 2007. Un database semplice può includere una sola tabella. Nella maggior parte dei database sono necessarie più tabelle. Si può ad esempio disporre di una tabella contenente informazioni sui prodotti, un'altra tabella contenente informazioni sugli ordini e un'altra tabella con informazioni sui clienti.



Ogni riga viene denominata **record** e ogni colonna viene denominata **campo**. Un record costituisce un modo significativo e coerente per combinare le informazioni su un determinato argomento. Un campo rappresenta un singolo elemento di informazioni, ovvero un tipo di elemento incluso in ogni record. Nella tabella Prodotti, ad esempio, ogni riga o record conterrà informazioni su un prodotto, mentre ogni colonna o campo includerà un tipo di informazioni sul prodotto, ad esempio il nome o il prezzo.

## Progettazione appropriata di un database

Il processo di progettazione di database è basato su determinati principi. Il primo è costituito dal fatto che la presenza di informazioni duplicate, denominate anche dati ridondanti, è negativa poiché causa uno spreco di spazio e aumenta le probabilità di errori e incoerenze. Il secondo principio è rappresentato dall'importanza della correttezza e della completezza delle informazioni. Se il database contiene informazioni non corrette, qualsiasi report che estrae informazioni dal database conterrà a propria volta informazioni non corrette. Di conseguenza, qualsiasi decisione a partire da tali report sarà basata su informazioni errate.

Una progettazione di database appropriata presenta pertanto le caratteristiche seguenti:

1. Suddivide le informazioni in tabelle per argomento in modo da ridurre i dati ridondanti.
2. Offre ad Access le informazioni necessarie per unire le informazioni delle tabelle in base alle esigenze.
3. Supporta e garantisce l'accuratezza e l'integrità delle informazioni.
4. Soddisfa le esigenze di elaborazione dei dati e creazione di report.

## Processo di progettazione

Il processo di progettazione è costituito dai passaggi seguenti:

* **Identificare lo scopo del database**

È così possibile prepararsi per i restanti passaggi.

* **Individuare e organizzare le informazioni necessarie**

Raccogliere tutti i tipi di informazioni che si desidera registrare nel database, ad esempio il nome del prodotto e il numero d'ordine.

* **Suddividere le informazioni in tabelle**

Suddividere gli elementi di informazioni in argomenti o entità principali, ad esempio Ordini o Prodotti. A ogni argomento verrà quindi assegnata una tabella.

* **Trasformare gli elementi di informazioni in colonne**

Determinare le informazioni che si desidera memorizzare in ogni tabella. A ogni elemento viene assegnato un campo, visualizzato come una colonna della tabella. Una tabella Dipendenti può ad esempio includere campi come Cognome e Data assunzione.

* **Specificare chiavi primarie**

Scegliere la chiave primaria di ogni tabella. La chiave primaria è la colonna utilizzata per identificare in modo univoco ogni riga, ad esempio ID prodotto o ID ordine.

* **Impostare le relazioni tra le tabelle**

Esaminare ogni tabella e determinare la relazione tra i dati di una tabella e i dati di altre tabelle. Aggiungere campi alle tabelle o creare nuove tabelle per chiarire le relazioni, in base alle esigenze.

* **Perfezionare la progettazione**

Analizzare la progettazione per individuare eventuali errori. Creare le tabelle e aggiungere alcuni record di dati di esempio. Verificare se è possibile ottenere i risultati desiderati dalle tabelle. Apportare modifiche alla progettazione in base alle esigenze.

* **Applicare le regole di normalizzazione**

Applicare le regole di normalizzazione dei dati per verificare la corretta strutturazione delle tabelle. Apportare modifiche alle tabelle in base alle esigenze.

## Definizione dello scopo del database

È opportuno definire su carta lo scopo del database, ovvero lo scopo, le modalità di utilizzo previste e gli utenti. Per un database di piccole dimensioni per una società a conduzione familiare, ad esempio, è possibile scrivere semplicemente "Il database dei clienti contiene un elenco di informazioni relative ai clienti per la generazione di elenchi di indirizzi e report". In caso di database più complesso o utilizzato da numerosi utenti, così come in genere in un ambiente aziendale, lo scopo può essere facilmente articolato in uno o più paragrafi e dovrà includere i tempi e le modalità con cui il database verrà utilizzato da ogni utente, in modo da disporre di una dichiarazione degli obiettivi ben definita a cui fare riferimento nell'intero processo di progettazione. Una dichiarazione di questo tipo consente di concentrarsi sugli obiettivi al momento delle decisioni.

## Individuazione e organizzazione delle informazioni necessarie

Per individuare e organizzare le informazioni necessarie, partire dalle informazioni esistenti. È ad esempio possibile che gli ordini di acquisto vengano registrati in un libro mastro o le informazioni relative ai clienti vengano mantenute in moduli cartacei in uno schedario. Raccogliere tali documenti ed elencare ogni tipo di informazioni visualizzato, ad esempio ogni casella riempita in un modulo. Se non si dispone di moduli esistenti, immaginare invece di dover progettare un modulo per registrare le informazioni relative ai clienti, identificando ed elencando le informazioni da inserire nel modulo e le caselle da riempire che dovranno essere create. Se ad esempio l'elenco dei clienti viene attualmente mantenuto su schede, esaminando tali schede si potrebbe riscontrare che ognuna contiene il nome di un cliente, l'indirizzo, la città, la provincia, il codice postale e il numero telefonico. Ognuno di questi elementi rappresenta una potenziale colonna di una tabella.

Nella preparazione di questo elenco, non è necessario definire immediatamente un elenco perfetto. Elencare invece ogni elemento possibile. Chiedere inoltre suggerimenti agli altri eventuali utenti del database. L'elenco potrà essere ottimizzato in un secondo momento.

Considerare quindi i tipi di report o elenchi indirizzi che si desidera generare dal database. Può ad esempio essere opportuno creare un report delle vendite di prodotti in modo da visualizzare le vendite per regione oppure un report di riepilogo dell'inventario che indichi i livelli di scorte dei prodotti. Può inoltre essere utile generare lettere tipo da inviare ai clienti per annunciare un evento di vendita oppure offrire un premio. Progettare mentalmente il report e immaginarne l'aspetto, considerando le informazioni da inserire nel report ed elencando ogni elemento. Ripetere questa operazione per la lettera tipo o per qualsiasi altro report che si prevede di creare.



Considerando i report e gli elenchi di indirizzi che si desidera creare, è possibile identificare gli elementi necessari nel database. Si supponga ad esempio di offrire ai clienti l'opportunità di acconsentire esplicitamente all'invio di aggiornamenti periodici per posta elettronica (o rifiutare esplicitamente tali aggiornamenti) e di voler stampare un elenco dei clienti che hanno acconsentito. Per registrare tali informazioni, si aggiunge una colonna “Invio posta elettronica” alla tabella dei clienti. Per ogni cliente, è possibile impostare il campo su Sì o No.

Il requisito dell'invio di messaggi di posta elettronica ai clienti suggerisce un altro elemento da registrare. Dopo aver determinato che un cliente desidera ricevere messaggi di posta elettronica, sarà inoltre necessario conoscere l'indirizzo a cui devono essere inviati. Dovrà pertanto essere registrato un indirizzo di posta elettronica per ogni cliente.

È opportuno creare un prototipo di ogni report o elenco di output e considerare gli elementi necessari per generare il report. Esaminando una lettera tipo, ad esempio, si possono riscontrare alcuni aspetti. Se si desidera includere una formula di apertura appropriata, ad esempio la stringa "sig.", "sig.ra" o "sig.na" in un saluto, sarà necessario creare un elemento per la formula di apertura. Poiché in genere si inizia una lettera con “Gentile sig. Macagno” anziché con “Gentile sig. Maurizio Macagno”, si rivelerà solitamente opportuno memorizzare il cognome separatamente dal nome.

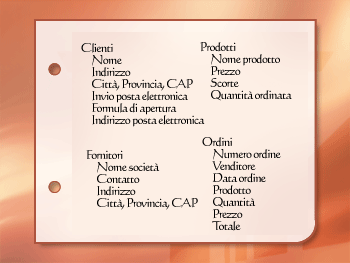
Un punto chiave da tenere presente è costituito dal fatto che è consigliabile suddividere al massimo ogni informazione nelle relative parti utili. Nel caso di un nome, per rendere rapidamente disponibile il cognome si suddividerà il nome in due parti, ovvero nome di battesimo e cognome. La memorizzazione separata del cognome si rivelerà utile, ad esempio, per ordinare un report in base al cognome. In generale, se si desidera eseguire l'ordinamento, la ricerca, il calcolo o la creazione di report in base a un elemento di informazioni, è consigliabile inserire tale elemento in un campo apposito.

Considerare le domande a cui si desidera trovare risposta mediante il database. Ad esempio, quante vendite del prodotto di rilievo sono state concluse nell'ultimo mese? Dove risiedono i migliori clienti? Qual è il fornitore del prodotto con il volume di vendite più elevato? Prevedendo queste domande è possibile identificare elementi aggiuntivi da registrare.

Dopo aver raccolto queste informazioni, è possibile procedere con il passaggio successivo.

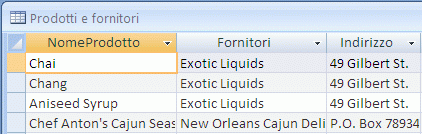
## Suddivisione delle informazioni in tabelle

Per suddividere le informazioni in tabelle, scegliere le entità principali, ovvero gli argomenti. Dopo l'individuazione e l'organizzazione delle informazioni per un database relativo alle vendite di prodotti, ad esempio, l'elenco preliminare può presentare l'aspetto illustrato di seguito.



In questo caso, le entità principali sono costituite da prodotti, fornitori, clienti e ordini. È pertanto opportuno iniziare con queste quattro tabelle: una per i dati relativi ai prodotti, una per i dati relativi ai fornitori, una per i dati relativi ai clienti e una per i dati relativi agli ordini. Nonostante l'elenco non sia completo, costituisce un punto di partenza appropriato. Si potrà continuare a perfezionare l'elenco fino a ottenere una progettazione efficiente.

Esaminando inizialmente l'elenco preliminare degli elementi, può apparire sufficiente inserire tutti gli elementi in un'unica tabella, anziché nelle quattro della figura precedente. Di seguito vengono illustrati i motivi per cui non è opportuno. Considerare la tabella della figura seguente.



In questo caso, ogni riga contiene informazioni sia sul prodotto che sul relativo fornitore. Poiché si può disporre di numerosi prodotti dello stesso fornitore, le informazioni relative al nome e all'indirizzo del fornitore devono essere ripetute molte volte. Ciò costituisce uno spreco di spazio su disco. La registrazione delle informazioni relative al fornitore una sola volta in una tabella Fornitori separata e il successivo collegamento a tale tabella dalla tabella Prodotti rappresentano una soluzione notevolmente più efficiente.

Un secondo problema associato a questa progettazione viene riscontrato quando si devono modificare le informazioni sul fornitore. Si supponga ad esempio di dover modificare l'indirizzo di un fornitore. Poiché tale indirizzo è presente in numerose posizioni, è accidentalmente possibile modificarlo in una posizione ma omettere di modificarlo in altre. Il problema può essere risolto registrando l'indirizzo del fornitore in un'unica posizione.

Quando si progetta il database, provare sempre a registrare ogni dato una sola volta. Se si rileva che le stesse informazioni devono essere ripetute in più posizioni, ad esempio l'indirizzo di un determinato fornitore, inserire tali informazioni in una tabella separata.

Si supponga infine di disporre di un unico prodotto fornito da Cantina ABC e di voler eliminare il prodotto mantenendo tuttavia le informazioni relative al nome e all'indirizzo del fornitore. Non è possibile eliminare il record del prodotto senza perdere le informazioni relative al fornitore. Poiché ogni record contiene sia i dati su un prodotto che i dati su un fornitore, gli uni non possono essere eliminati senza eliminare gli altri. Per mantenere separati questi dati, è necessario dividere questa unica tabella in due: una tabella per le informazioni relative ai prodotti e un'altra tabella per le informazioni relative ai fornitori. Eliminando il record di un prodotto dovranno essere eliminati soltanto i dati sul prodotto, non i dati sul fornitore.

Dopo che è stato scelto l'argomento rappresentato da una tabella, le colonne della tabella dovranno contenere soltanto dati relativi a tale argomento. Nella tabella dei prodotti, ad esempio, dovranno essere memorizzati soltanto dati sui prodotti. Poiché l'indirizzo del fornitore costituisce un dato relativo al fornitore e non un dato relativo al prodotto, tale indirizzo appartiene alla tabella dei fornitori.

## Trasformazione degli elementi di informazioni in colonne

Per determinare le colonne di una tabella, definire le informazioni di cui è necessario tenere traccia relativamente all'argomento registrato nella tabella. Per la tabella Clienti, ad esempio, un elenco iniziale appropriato di colonne è ad esempio costituito da Nome, Indirizzo, Città-Provincia-CAP, Invio posta elettronica, Formula di apertura e Indirizzo posta elettronica. Ogni record della tabella contiene lo stesso insieme di colonne, in modo da consentire la memorizzazione delle informazioni di Nome, Indirizzo, Città-Provincia-CAP, Invio posta elettronica, Formula di apertura e Indirizzo posta elettronica per ogni record. La colonna dell'indirizzo, ad esempio, contiene gli indirizzi dei clienti. Ogni record contiene i dati su un cliente e il campo dell'indirizzo contiene l'indirizzo di tale cliente.

Dopo aver determinato l'insieme iniziale di colonne per ogni tabella, è possibile perfezionare ulteriormente le colonne. È ad esempio opportuno memorizzare il nome del cliente in due colonne separate, corrispondenti a nome di battesimo e cognome, in modo da eseguire l'ordinamento, la ricerca e l'indicizzazione in base a tali colonne. Analogamente, l'indirizzo è effettivamente costituito da cinque componenti distinti, ovvero indirizzo, città, provincia, codice postale e paese, che è ugualmente opportuno memorizzare in colonne separate. Per eseguire un'operazione di ordinamento, ricerca o filtro in base alla provincia, ad esempio, è necessario che le informazioni relative alla provincia siano memorizzate in una colonna distinta.

È inoltre consigliabile valutare se il database conterrà informazioni di provenienza esclusivamente nazionale o anche internazionale. Ad esempio, se si prevede di memorizzare indirizzi internazionali è preferibile disporre di una colonna Zona anziché Provincia, in modo da poter inserire in tale colonna sia le province nazionali che le zone di altri paesi. In modo analogo, se si prevede di memorizzare indirizzi statunitensi, è più opportuno utilizzare Codice postale zip anziché CAP.

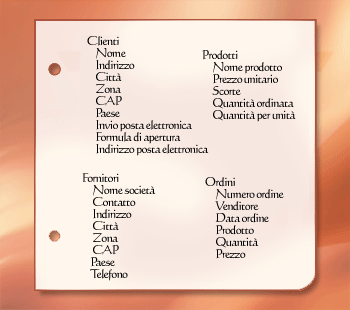
Di seguito sono elencati alcuni suggerimenti per la definizione delle colonne.

* **Non includere colonne calcolate**

Nella maggior parte dei casi, è consigliabile non memorizzare risultati di calcoli nelle tabelle. Il calcolo potrà invece essere eseguito in Access quando si desidera visualizzare il risultato. Si supponga ad esempio di disporre di un report dei prodotti ordinati in cui viene visualizzato il subtotale delle unità ordinate per ogni categoria di prodotto nel database. Non esiste tuttavia una colonna del subtotale Quantità ordinata in una tabella. Nella tabella Prodotti è invece inclusa una colonna Quantità ordinata in cui sono memorizzate le unità ordinate per ogni prodotto. Il subtotale verrà calcolato da Access utilizzando tali dati ogni volta che si stampa il report, ma non verrà memorizzato in una tabella.

* **Memorizzare le informazioni suddividendole al massimo nelle relative parti logiche**

Può apparire sufficiente disporre di un unico campo per i nomi completi o per nomi e descrizioni dei prodotti. Combinando più tipi di informazioni in un campo, tuttavia, risulterà in seguito difficoltoso recuperare singoli dati. Provare a suddividere le informazioni in parti logiche, ad esempio creare campi separati per nome di battesimo e cognome oppure per nome, categoria e descrizione del prodotto.



Dopo aver perfezionato le colonne di dati delle tabelle, è possibile scegliere la chiave primaria di ogni tabella.

## Specifica di chiavi primarie

Ogni tabella dovrà includere una colonna o un insieme di colonne che identifica in modo univoco ogni riga memorizzata nella tabella. Si tratta in genere di un numero di identificazione univoco, ad esempio un numero di ID dipendente o un numero di serie. Nella terminologia dei database, queste informazioni vengono definite la **chiave primaria** della tabella. I campi chiave primaria vengono utilizzati da Access per associare rapidamente i dati di più tabelle e combinare i dati per l'utente.

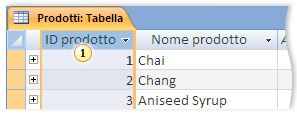
Se si dispone già di un identificatore univoco per una tabella, ad esempio un numero prodotto che identifica in modo univoco ogni prodotto del catalogo, è possibile utilizzare tale identificatore come chiave primaria della tabella, ma solo se i valori della colonna saranno sempre diversi per ogni record. Non sono consentiti valori duplicati in una chiave primaria. Ad esempio, non utilizzare nomi di persona come chiave primaria, poiché i nomi non sono univoci ed è facile disporre di due persone con lo stesso nome all'interno della stessa tabella.

Una chiave primaria deve sempre disporre di un valore. Se il valore di una colonna può talvolta risultare non assegnato o sconosciuto (valore mancante), tale valore non può essere utilizzato come componente di una chiave primaria.

È sempre consigliabile scegliere una chiave primaria il cui valore non verrà modificato. In un database che utilizza più tabelle, la chiave primaria di una tabella può essere utilizzata come riferimento in altre tabelle. Se la chiave primaria viene modificata, la modifica dovrà inoltre essere applicata in ogni posizione in cui viene fatto riferimento alla chiave. L'utilizzo di una chiave primaria non soggetta a modifiche riduce la possibilità che la chiave primaria risulti non sincronizzata con le altre tabelle che vi fanno riferimento.

Come chiave primaria viene in genere utilizzato un numero univoco arbitrario. È ad esempio possibile assegnare a ogni ordine un numero d'ordine univoco, con l'unico scopo di identificare un ordine. Dopo l'assegnazione, il numero d'ordine non verrà mai modificato.

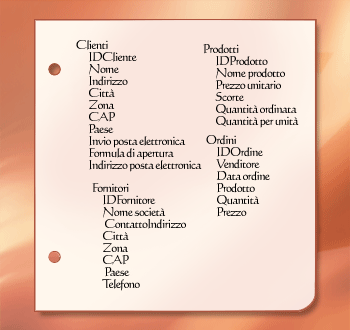
Se non si individua una colonna o un insieme di colonne che potrebbe costituire una chiave primaria appropriata, può essere opportuno utilizzare un colonna con tipo di dati Contatore. Quando si utilizza il tipo di dati Contatore, in Access viene assegnato automaticamente un valore. Un identificatore di questo tipo non contiene dati o informazioni effettive che descrivono la riga rappresentata. Tali identificatori costituiscono la chiave primaria ideale poiché non sono soggetti a modifiche. Esistono maggiori probabilità che venga modificata una chiave primaria contenente dati su una riga, ad esempio un numero telefonico o il nome di un cliente, poiché le informazioni effettive sono soggette a modifiche.



Callout 1 Una colonna impostata sul tipo di dati Contatore costituisce in genere una chiave primaria appropriata. Non esistono due ID di prodotto uguali.

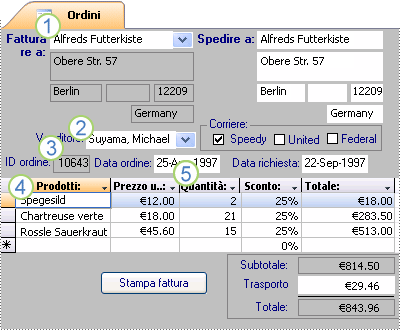
In alcuni casi, può essere opportuno utilizzare due o più campi che congiuntamente costituiscono la chiave primaria di una tabella. Nella chiave primaria di una tabella Dettagli ordini contenente le voci degli ordini, ad esempio, verranno utilizzate due colonne: ID ordine e ID prodotto. Una chiave primaria basata su più colonne viene denominata anche chiave composta.

Nel database relativo alle vendite di prodotti, è possibile creare come chiave primaria una colonna di tipo Contatore per ogni tabella: ID prodotto per la tabella Prodotti, ID ordine per la tabella Ordini, ID cliente per la tabella Clienti e ID fornitore per la tabella Fornitori.



## Creazione di relazioni tra tabelle

Dopo aver suddiviso le informazioni in tabelle, è necessario combinare nuovamente le informazioni in modi significativi. Nella maschera seguente sono ad esempio incluse informazioni di diverse tabelle.



Callout 1 Le informazioni nella maschera provengono dalla tabella Clienti...

Callout 2 ...dalla tabella Dipendenti...

Callout 3 ...dalla tabella Ordini...

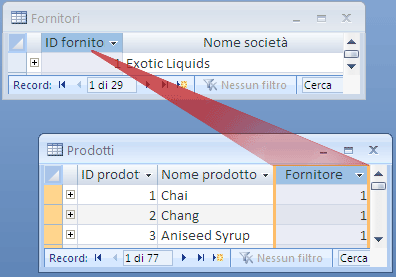
Callout 4 ...dalla tabella Prodotti...

Callout 5 ...e dalla tabella Dettagli ordini.

Access è un sistema di gestione di database relazionali. In un database relazionale, le informazioni vengono suddivise in tabelle separate per argomento. Si utilizzano quindi relazioni tra le tabelle per combinare le informazioni in base alle esigenze.

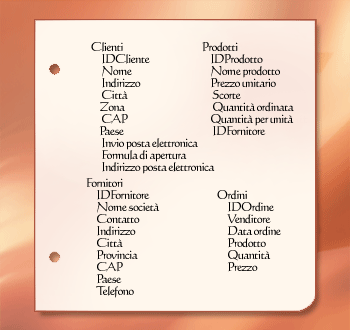
### Creazione di una relazione uno-a-molti

Si considerino a titolo di esempio le tabelle Fornitori e Prodotti del database relativo agli ordini di prodotti. Un fornitore può fornire qualsiasi numero di prodotti. Di conseguenza, per qualsiasi fornitore rappresentato nella tabella Fornitori possono essere rappresentati numerosi prodotti nella tabella Prodotti. Tra la tabella Fornitori e la tabella Prodotti esiste pertanto una relazione uno-a-molti.



Per rappresentare una relazione uno-a-molti nella progettazione del database, aggiungere la chiave primaria del lato "uno" della relazione come ulteriore colonna o insieme di colonne nella tabella sul lato "molti" della relazione. In questo caso, ad esempio, si aggiunge la colonna ID fornitore della tabella Fornitori alla tabella Prodotti. L'ID del fornitore può quindi essere utilizzato da Access nella tabella Prodotti per individuare il fornitore corretto per ogni prodotto.

La colonna ID fornitore della tabella Prodotti viene denominata chiave esterna. Una chiave esterna è costituita dalla chiave primaria di un'altra tabella. La colonna ID fornitore della colonna Prodotti è una chiave esterna poiché costituisce anche la chiave primaria della tabella Fornitori.



Il join tra tabelle correlate è basato sulla definizione di coppie di chiavi primarie e chiavi esterne. In caso di dubbi sulle tabelle che dovranno condividere una colonna comune, l'identificazione di una relazione uno-a-molti garantisce che le due tabelle considerate necessitino effettivamente di una colonna condivisa.

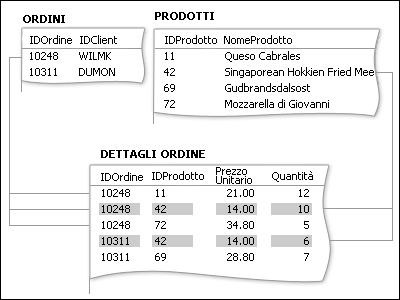
### Creazione di una relazione molti-a-molti

Si consideri la relazione tra la tabella Prodotti e la tabella Ordini.

Un singolo ordine può includere più prodotti, mentre un singolo prodotto può essere incluso in molti ordini. A ogni record della tabella Ordini possono pertanto corrispondere numerosi record della tabella Prodotti e a ogni record della tabella Prodotti possono corrispondere numerosi record della tabella Ordini. Questo tipo di relazione viene denominata relazione molti-a-molti, poiché a qualsiasi prodotto possono corrispondere molti ordini e a qualsiasi ordine possono corrispondere molti prodotti. Per rilevare relazioni molti-a-molti tra le tabelle, è importante considerare entrambi i lati della relazione.

L'esistenza di una relazione molti-a-molti tra gli argomenti delle due tabelle, ordini e prodotti, presenta un problema. Se si tenta di creare la relazione tra le due tabelle aggiungendo il campo ID prodotto alla tabella Ordini, per includere più prodotti in un ordine è necessario disporre di più record per ordine nella tabella Ordini, ripetendo le informazioni relative all'ordine in ogni riga correlata a un singolo ordine e ottenendo così una progettazione inefficiente che può determinare dati non accurati. Lo stesso problema si riscontra inserendo il campo ID ordine nella tabella Prodotti, poiché si disporrà di più record per ogni prodotto nella tabella Prodotti. Questo problema deve essere risolto.

A tale scopo, si crea una terza tabella, in genere denominata tabella di collegamento, che consente di suddividere la relazione molti-a-molti in due relazioni uno-a-molti. Nella terza tabella viene inserita la chiave primaria di ognuna delle due tabelle, registrando così ogni occorrenza o istanza della relazione.



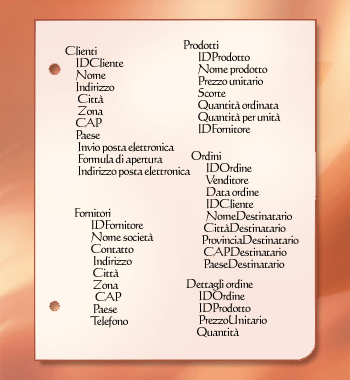
Ogni record della tabella Dettagli ordini rappresenta una voce di un ordine. La chiave primaria della tabella Dettagli ordini è costituita da due campi, corrispondenti alle chiavi esterne delle tabelle Ordini e Prodotti. Non è sufficiente utilizzare il solo campo ID ordine come chiave primaria per la tabella, poiché un ordine può contenere numerose voci. Poiché l'ID di ordine viene ripetuto per ogni voce di un ordine, il campo non contiene valori univoci. L'utilizzo del solo campo ID prodotto non è ugualmente sufficiente, poiché un prodotto può essere incluso in molti ordini diversi. Congiuntamente, tuttavia, i due campi generano sempre un valore univoco per ogni record.

Nel database relativo alle vendite di prodotti, la tabella Ordini e la tabella Prodotti non sono direttamente correlate. Sono invece correlate in modo indiretto tramite la tabella Dettagli ordini. La relazione molti-a-molti tra ordini e prodotti è rappresentata nel database utilizzando due relazioni uno-a-molti:

* Tra la tabella Ordini e la tabella Dettagli ordini esiste una relazione uno-a-molti. Ogni ordine può contenere più voci, ma ogni voce è connessa a un solo ordine.
* Tra la tabella Prodotti e la tabella Dettagli ordini esiste una relazione uno-a-molti. A ogni prodotto possono essere associate più voci, ma ogni voce fa riferimento a un solo prodotto.

Dalla tabella Dettagli ordini è possibile determinare tutti i prodotti in uno specifico ordine, nonché tutti gli ordini per uno specifico prodotto.

Dopo l'incorporamento della tabella Dettagli ordini, l'elenco di tabelle e campi presenterà un aspetto simile al seguente:



### Creazione di una relazione uno-a-uno

Un altro tipo di relazione è rappresentato dalla relazione uno-a-uno. Si supponga ad esempio di dover registrare alcune speciali informazioni supplementari sui prodotti che sono raramente necessarie o sono applicabili soltanto ad alcuni prodotti. Poiché non sono in genere necessarie e la relativa memorizzazione nella tabella Prodotti determinerebbe spazio vuoto per ogni prodotto a cui non sono applicabili, le informazioni vengono inserite in una tabella separata. Come per la tabella Prodotti, come chiave primaria si utilizza ID prodotto. Tra questa tabella supplementare e la tabella Prodotti esiste una relazione uno-a-uno. Per ogni record della tabella Prodotti esiste un unico record corrispondente nella tabella supplementare. Quando si identifica una relazione di questo tipo, entrambe le tabelle devono condividere un campo comune.

Quando si rileva l'esigenza di una relazione uno-a-uno nel database, considerare se le informazioni delle due tabelle possono essere unite in una tabella. Se non si desidera tale unione per qualsiasi motivo, ad esempio poiché determinerebbe una notevole quantità di spazio vuoto, la relazione verrà rappresentata nella progettazione come segue:

* Se le due tabelle condividono lo stesso argomento, è probabilmente possibile impostare la relazione utilizzando la stessa chiave primaria in entrambe le tabelle.
* Se le due tabelle riguardano argomenti diversi con chiavi primarie diverse, scegliere una delle tabelle e inserirne la chiave primaria nell'altra tabella come chiave esterna.

L'identificazione delle relazioni tra le tabelle consente di verificare la presenza delle tabelle e delle colonne appropriate. Se esiste una relazione uno-a-uno o uno-a-molti, le tabelle interessate devono condividere una o più colonne comuni. Se esiste una relazione molti-a-molti, è necessaria una terza tabella per rappresentare la relazione.

## Perfezionamento della struttura

Quando si dispone delle tabelle, delle relazioni e dei campi necessari, è consigliabile creare e popolare le tabelle con dati di esempio e provare a utilizzare le informazioni creando query, aggiungendo nuovi record e così via. È così possibile evidenziare potenziali problemi. Può ad esempio essere necessario aggiungere una colonna che non è stata inserita durante la fase di progettazione oppure suddividere una tabella in due per rimuovere dati duplicati.

Controllare se è possibile utilizzare il database per ottenere le risposte desiderate. Creare bozze delle maschere e dei report e verificare se contengono i dati previsti. Cercare eventuali dati duplicati non necessari e, se presenti, modificare la progettazione in modo da eliminarli.

Quando si eseguono prove con il database iniziale, sarà possibile individuare margini di miglioramento. Di seguito sono elencati alcuni aspetti da verificare:

* Sono state omesse eventuali colonne? In caso affermativo, le informazioni appartengono alle tabelle esistenti? Se le informazioni riguardano un altro argomento, può essere necessario creare un'altra tabella. Creare una colonna per ogni informazione di cui si deve tenere traccia. Se le informazioni non possono essere calcolate da altre colonne, è probabilmente necessario aggiungere una nuova colonna.
* Sono presenti colonne non necessarie poiché calcolabili dai campi esistenti? Se un'informazione può essere calcolata da altre colonne esistenti, come nel caso di un prezzo scontato calcolato dal prezzo di listino, è in genere preferibile evitare di creare una nuova colonna.
* Si immettono ripetutamente informazioni duplicate in una delle tabelle? In questo caso, è probabilmente necessario suddividere la tabella in due tabelle con una relazione uno-a-molti.
* Sono presenti tabelle con molti campi, un numero limitato di record e numerosi campi vuoti nei singoli record? In questo caso, può essere opportuno riprogettare la tabella in modo da ridurre il numero di campi e aumentare il numero di record.
* Ogni informazione è stata suddivisa nelle sue unità significative più piccole? Se è necessario eseguire l'ordinamento, la ricerca, il calcolo o la creazione di report in base a un'informazione, inserire tale informazione in un'apposita colonna.
* Ogni colonna contiene un dato relativo all'argomento della tabella? Una colonna che non contiene informazioni sull'argomento della tabella appartiene a una tabella diversa.
* Sono rappresentate tutte le relazioni tra le tabelle, mediante campi comuni o una terza tabella? Le relazioni uno-a-uno e uno-a-molti necessitano di colonne comuni, mentre per le relazioni molti-a-molti è necessaria una terza tabella.

### Perfezionamento della tabella Prodotti

Si supponga che ogni prodotto nel database relativo alla vendite di prodotti appartenga a una categoria generale, ad esempio bevande, salse o pesce. La tabella Prodotti potrebbe includere un campo contenente la categoria di ogni prodotto.

Dopo aver esaminato e perfezionato la progettazione del database, si supponga di decidere di memorizzare una descrizione della categoria insieme al relativo nome. L'aggiunta di un campo Descrizione categoria alla tabella Prodotti non rappresenta la soluzione appropriata, poiché determina l'esigenza di ripetere ogni descrizione di categoria per ogni prodotto appartenente alla categoria.

È preferibile considerare le categorie come un nuovo argomento di cui deve essere tenuta traccia nel database, con una tabella e una chiave primaria specifiche. È quindi possibile aggiungere la chiave primaria della tabella Categorie alla tabella Prodotti come chiave esterna.

Tra le tabelle Categorie e Prodotti esiste una relazione uno-a-molti. Una categoria può includere più prodotti, ma un prodotto può appartenere a una sola categoria.

Quando si esaminano le strutture delle tabelle, prestare attenzione ai gruppi ripetuti. Considerare ad esempio una tabella contenente le colonne seguenti:

* ID prodotto
* Nome
* ID prodotto1
* Nome1
* ID prodotto2
* Nome2
* ID prodotto3
* Nome3

In questo caso, ogni prodotto costituisce un gruppo ripetuto di colonne che differisce dagli altri solo per l'aggiunta di un numero alla fine del nome della colonna. Quando si rilevano colonne numerate in questo modo, è consigliabile rivedere la progettazione.

Una progettazione di questo tipo presenta diversi difetti. Impone innanzitutto l'applicazione di un limite massimo al numero di prodotti. Quando viene superato tale limite, è necessario eseguire l'aggiunta di un nuovo gruppo di colonne alla struttura della tabella, che costituisce un'importante attività amministrativa.

Un altro problema è rappresentato dal fatto che i fornitori con un numero di prodotti inferiore al numero massimo determineranno uno spreco di spazio, poiché le colonne aggiuntive resteranno vuote. Il difetto più grave associato a una progettazione di questo tipo è costituito dalla difficoltà di esecuzione di numerose attività, come l'ordinamento o l'indicizzazione della tabella in base al nome o all'ID del prodotto.

Ogni volta che si rilevano gruppi ripetuti, rivedere attentamente la progettazione prendendo in considerazione la suddivisione della tabella in due. Nell'esempio precedente è preferibile utilizzare due tabelle, una per i fornitori e una per i prodotti, collegate mediante l'ID del fornitore.

## Applicazione delle regole di normalizzazione

Il passaggio successivo nella progettazione può essere costituito dall'applicazione delle regole di normalizzazione dei dati, talvolta denominate semplicemente regole di normalizzazione. Queste regole consentono di verificare la corretta strutturazione delle tabelle. Il processo di applicazione delle regole alla progettazione del database viene denominato normalizzazione del database o semplicemente normalizzazione.

Il termine normalizzazione indica il processo di organizzazione dei dati in un database. Tale processo comprende la creazione di tabelle e la definizione di relazioni tra di esse sulla base di regole progettate in modo da proteggere i dati e rendere il database più flessibile mediante l'eliminazione della ridondanza e delle dipendenze incoerenti.   
  
La presenza di dati ridondanti comporta uno spreco di spazio su disco nonché problemi di manutenzione. Se è necessario modificare dati presenti in più posizioni, la modifica deve essere effettuata ovunque seguendo le stesse modalità. Ad esempio, è più agevole implementare la modifica relativa all'indirizzo di un cliente se i dati relativi sono memorizzati solo nella tabella Clienti.   
  
Che cos'è una "dipendenza incoerente"? Mentre è intuitivo per un utente cercare nella tabella Clienti l'indirizzo di un cliente specifico, può non avere senso cercare in tale tabella le informazioni relative allo stipendio del dipendente che si occupa di quel cliente. Le informazioni sullo stipendio sono correlate al dipendente, ovvero sono dipendenti da quest'ultimo, pertanto devono essere spostate nella tabella Dipendenti. Le dipendenze incoerenti possono rendere difficile l'accesso ai dati, in quanto il percorso per la ricerca dei dati può risultare mancante o danneggiato.   
  
Per la normalizzazione dei database è necessario seguire alcune regole, ciascuna delle quali viene definita "forma normale". Se si osserva la prima regola, il database viene considerato nella "prima forma normale". Se si osservano le prime tre regole, il database viene considerato nella "terza forma normale". Le regole vengono applicate in successione, verificando in ogni passaggio che la struttura raggiunga la cosiddetta "forma normale". Sono in genere accettate cinque forme normali, dalla prima forma normale fino alla quinta forma normale. In questo articolo vengono illustrate le prime tre, poiché la terza forma normale è considerata il livello massimo necessario per la maggior parte delle applicazioni.   
  
Come per molte regole e specifiche formali, gli scenari reali non consentono sempre la conformità totale. In generale poiché la normalizzazione richiede l'uso di tabelle aggiuntive, viene considerata troppo dispendiosa da alcuni clienti. Se si decide di violare una delle prime tre regole di normalizzazione, assicurarsi che l'applicazione sia in grado di prevenire i problemi che possono derivarne, quali la ridondanza dei dati e le dipendenze incoerenti.

### Vedi dispensa relativa