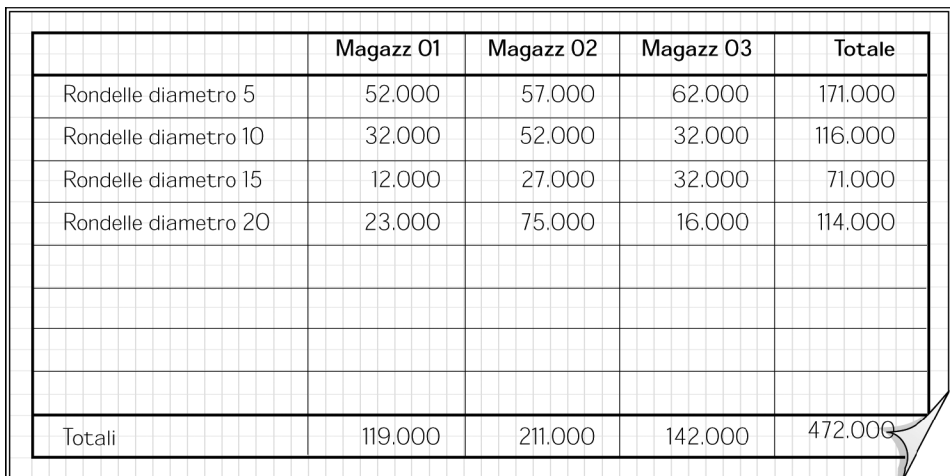


# Introduzione

## Presentazione

Lo strumento di lavoro più diffuso negli uffici di tutto il mondo è un foglio rettangolare di carta a quadretti, disposto in orizzontale, cioè con il lato lungo in basso. Su questo foglio si scrivono numeri che rappresentano di solito realtà economiche: quantità vendute o prodotte, numero di pezzi in magazzino, costi e prezzi unitari, associando i numeri a una descrizione, come vediamo qui di seguito.



	Magazz 01	Magazz 02	Magazz 03	Totale
Rondelle diametro 5	52.000	57.000	62.000	171.000
Rondelle diametro 10	32.000	52.000	32.000	116.000
Rondelle diametro 15	12.000	27.000	32.000	71.000
Rondelle diametro 20	23.000	75.000	16.000	114.000
Totali	119.000	211.000	142.000	472.000

Gli americani, che hanno un nome per tutte le cose, chiamano questo foglio *spreadsheet*, che vuol dire letteralmente “foglio dispiegato” e se ne servono egregiamente, con la convinta determinazione che hanno di solito quando lavorano.

Qui da noi non ha un nome ufficiale o comunemente accettato, possiamo chiamarlo *foglio di lavoro*, tanto per intenderci. Pur non avendo un nome, è comunque usato parecchio anche in Italia, in tutte le innumerevoli situazioni di lavoro in cui occorre presentare sinteticamente un insieme di informazioni quantitative.

Le linee verticali del foglio tracciano *colonne* e quelle orizzontali *righe*. Nel punto in cui si intersecano righe e colonne si formano *celle*, nelle quali si scrivono numeri o descrizioni, come nell'esempio che stiamo vedendo.

Il foglio di lavoro o spreadsheet che dir si voglia è indubbiamente un ottimo strumento per raccogliere dati, ma crea non pochi fastidi quando si sbaglia a inserire un dato in una cella o quando occorre modificare qualcosa.

Prendiamo l'esempio che abbiamo davanti. Se nel Magazz01 la giacenza di rondelle di diametro 5 passa da 52.000 pezzi a 50.000, il totale in fondo alla riga di queste rondelle non è più 171.000, ma 169.000. E il totale generale per la colonna del Magazz01 scende a 117.000, mentre si riduce – sempre di 2.000 unità – la colonna dei totali generali (da 472.000 passa a 470.000).

Per decenni contabili, analisti di gestione, magazzinieri e altri ancora hanno accettato questo inconveniente in cambio dei grandi vantaggi di chiarezza che offre il foglio di lavoro. L'unico accorgimento possibile per non soffrire troppo consisteva nell'usare sempre fogli di carta robusta, scrivere i numeri soltanto con la matita e tenere a portata di mano una gomma per cancellare. Non è un caso che negli Stati Uniti, dove si fa un uso smodato degli spreadsheet, non si trovino in commercio matite di legno senza il gommino fissato a un'estremità.

## Tra cronaca e storia

Nel 1978 due giovanotti iscritti al secondo anno del corso per il Master of Business Administration della Harvard Business School si chiesero se non sarebbe stato possibile trovare una soluzione non cartacea e non manuale al problema della compilazione dei fogli di lavoro. Studiare per conseguire l'MBA alla Harvard Business School vuol dire passare intere giornate (e spesso nottate) a riempire decine di fogli di lavoro, trascrivendovi stati patrimoniali, conti economici, schemi di budget, rendiconti di costi e ricavi, per analizzare e capire gli aspetti quantitativi della gestione delle imprese.

Da poco più di un anno era sul mercato un traliccio formato da un basamento di lamiera che conteneva un po' di circuiti integrati e nel quale era ricavato l'alloggiamento per una tastiera; sul basamento si appoggiava un monitor monocromatico (di quelli che si usavano per gli oscilloscopi) e il tutto veniva presentato col nome di Apple II: era uno dei primissimi personal computer (ancora non avevano questo nome) nati dalla fertile inventiva di alcuni giovanotti californiani.

I nostri amici, che si chiamavano Dan Bricklin e Bob Frankston, stufo marci di mettere giù i numeri a matita, di fare i calcoli a mano e di cancellare e ricancellare i dati parziali per farli quadrare con i totali (o viceversa), decisero di investire la cospicua somma di 1000 dollari e di provare a fare su Apple II un programma che servisse a compilare i fogli di lavoro.

Dopo pochi mesi, nella primavera del 1979, il risultato tanto atteso prese forma, gli venne dato il nome di VisiCalc e fu un trionfo. La voce passò fulmineamente di bocca in bocca (l'Harvard Business School è una immensa cassa di risonanza), tutti coloro che avevano a che fare con gli spreadsheet (e negli USA sono legioni) vollero provare se con VisiCalc si potevano davvero ottenere i risultati sperati (numeri giusti subito, calcoli e risultati aggiornati automaticamente) e la società Apple Computer Inc., che aveva fino ad allora venduto poche decine di esemplari di Apple II, fu subissata di richieste e divenne ben presto leader di un fiorente mercato in sviluppo.

VisiCalc dominò il mercato per qualche anno, poi comparve una società di software molto aggressiva, di nome Lotus Development Corp., che sviluppò un prodotto concorrente, predisposto per lavorare su macchine più potenti, quali il Personal Computer IBM che nacque nell'agosto del 1981.

Il programma creato dalla Lotus per gli spreadsheet si chiamava 1-2-3, per due ragioni: veniva propagandato con lo slogan «facile da usare, come fare 1-2-3...» e perché i suoi creatori erano appassionati di musica (altri prodotti della stessa società si chiamarono poi Symphony e Jazz).

Il valzer di Lotus 1-2-3 durò a lungo: il prodotto era molto ben fatto e appagava completamente tutte le esigenze dei milioni di utenti che usavano un personal computer soltanto per creare e compilare fogli di lavoro. Il software lavorava nell'ambiente a caratteri tipico delle macchine basate sul sistema operativo MS-DOS; pochi fronzoli, molta sostanza, quello che contava era poter costruire in fretta e bene i fogli di calcolo ed essere sicuri che i totali per riga e colonna fossero sempre corretti e aggiornati.

## **Il ruolo di Microsoft**

Che cosa faceva, nel frattempo, la Microsoft Corporation? Tramava nell'ombra, verrebbe voglia di dire. Non esattamente. Agli inizi degli Ottanta del secolo scorso, Microsoft era una piccola società che aveva vinto una lotteria. Per un complesso concorso di circostanze era proprietaria del sistema operativo MS-DOS che veniva distribuito insieme con i Personal Computer IBM. Considerato l'enorme successo di vendita di queste macchine e dei milioni di esemplari costruiti dai concorrenti in modo da dare le stesse identiche prestazioni, la richiesta di copie del sistema operativo MS-DOS e l'impegno per aggiornarlo e potenziarlo assorbivano totalmente le energie della piccola e grintosa società di Seattle.

Per un lungo periodo, quindi, Microsoft si tenne ai margini del mercato del software applicativo, per concentrarsi sul software di sistema, potenziando MS-DOS e preparando un ambiente di lavoro più gradevole e più facile da usare, che sarebbe poi stato Windows. La domanda di programmi per gli spreadsheet, però, era talmente vivace che Microsoft pensò di fare comunque qualcosa e mise a punto nel 1985 un suo prodotto, che chiamò orgogliosamente Excel, destinato a lavorare sulla nuova generazione di macchine create dalla Apple Computer Inc. Queste macchine, che si chiamavano Macintosh (dal nome di una varietà di mele della California), erano strutturalmente diverse dai PC: invece di lavorare con i caratteri presentavano tutto per mezzo di immagini (le lettere e i numeri che si vedevano sul monitor erano in realtà micro disegni), usavano il mouse al posto della tastiera per l'immissione dei comandi più frequenti ed erano in genere molto più semplici e gradevoli da usare degli austeri personal computer IBM e derivati.

Excel nacque, quindi, nel contesto grafico dei Macintosh e venne proposto per i PC soltanto quando su queste macchine si rese disponibile un software, Windows, che trasformava l'ambiente di lavoro originario dei PC in un ambiente grafico simile a quello dei computer della Apple.

L'accoppiata Windows ed Excel mise fuori dal mercato degli spreadsheet il poderoso concorrente della Lotus (la società sopravvisse occupandosi di altro) e oggi Excel è lo strumento software più diffuso per lavorare su fogli di calcolo con un personal computer dotato del sistema operativo Windows.

## I grandi computer e i dati

Il tumultuoso sviluppo dei personal computer negli ultimi cinque anni in tutte le loro varianti, fino agli attuali smartphone (che sono computer a tutti gli effetti nei quali è integrato un telefono), tende a far dimenticare che i computer non sono nati come oggetti d'uso personale, ma come macchine industriali di grandi dimensioni, dette genericamente “mainframe”, utilizzate esclusivamente come strumenti per gestire enti e istituzioni.

La prima di queste macchine si chiamava UNIVAC ed esordì sul mercato agli inizi degli anni Cinquanta del secolo scorso, seguita poco tempo dopo da altre macchine chiamate 709 e poi 7090 nel decennio successivo e poi ancora negli anni Settanta, Ottanta e Novanta si diffusero nuovi modelli di mainframe fisicamente più grandi e più potenti, prodotti da numerose società che operavano nel mercato informatico di allora, fra le quali primeggiava la International Business Machines meglio nota con la sigla Ibm.

Per ottenere qualcosa di utile da un computer – si tratti di un mainframe grande come una sala da ballo sul quale sono attestate reti di migliaia di computer satelliti e di terminali o di un semplice tablet tascabile – è necessario fargli eseguire un *programma*, vale a dire un insieme ben organizzato di istruzioni che devono essere scritte a mano da una persona che conosca quello che viene definito un po' pomposamente il “linguaggio” di quel computer, cioè l'insieme delle istruzioni che la macchina è in grado di eseguire. Questo vincolo è stato per tutta la storia dell'informatica il più faticoso e costoso ostacolo da superare per arrivare a un uso ottimale dei computer: il progresso vertiginoso nell'elettronica raddoppiava ogni 24 mesi la velocità dei circuiti, la parte tangibile, l'*hardware*, dei computer, mentre lo sviluppo nella realizzazione dei programmi, della parte non tangibile, il *software* dei sistemi, restava lontanissimo da progressi analoghi.

Naturalmente anche nel campo della programmazione si sono fatti notevoli passi avanti rispetto agli inizi. Agli albori dell'informatica si davano istruzioni ai computer utilizzando direttamente il cosiddetto “linguaggio macchina”, ovverosia il codice delle istruzioni elementari predisposte dai progettisti del nucleo logico del computer, la Central Processing Unit o CPU.

Per rendere meno arduo il lavoro si ricorreva a un linguaggio elementare chiamato genericamente *Assembler*, col quale si potevano scrivere le istruzioni usando una codifica in lettere e numeri. Un programma scritto in *Assembler* veniva dato in input a un programma speciale, che traduceva le istruzioni nel codice binario che era l'unico “capito” dalle CPU.

Per migliorare la produttività del lavoro di programmazione vennero sviluppati, a partire dalla fine degli anni Cinquanta del secolo scorso, numerosi linguaggi concepiti per semplificare ulteriormente la stesura dei programmi. Diversamente dall'*Assembler*, i nuovi linguaggi di programmazione non contenevano istruzioni che corrispondevano direttamente con quelle del linguaggio macchina, ma mettevano a disposizione comandi formulati con parole chiave e operatori aritmetici e logici, molto più semplici da capire e da ricordare delle astruse istruzioni dell'*Assembler*.

Le sequenze di caratteri in cui si concretizzavano i programmi scritti usando questi linguaggi venivano acquisite in input da programmi specializzati (scritti in *Assembler*) che “traducevano” quel flusso di caratteri in istruzioni in linguaggio macchina, che venivano poi passate per l'esecuzione ai computer mainframe.

Due linguaggi di programmazione si affermarono col passare del tempo fra i tanti che furono sperimentati, chiamati rispettivamente FORTRAN (da *FOR*mula *TRAN*slation)

particolarmente orientato allo sviluppo di programmi per il calcolo scientifico e COBOL (da *COmmon Business Oriented Language*), concepito per creare programmi per la gestione aziendale. La diffusione e il consolidarsi nell'uso di questi linguaggi ha determinato il rigoglioso sviluppo dell'informatica gestionale, che costituisce tuttora la spina dorsale dei sistemi di gestione delle grandi strutture organizzative (imprese e istituzioni).

COBOL e FORTRAN sono linguaggi "procedurali", vale a dire permettono di definire puntualmente procedure da seguire per elaborare i dati, per esempio preparare fatture partendo da bolle di consegna, listini dei prezzi e indirizzi dei clienti o bollettini di stipendi elaborando dati sugli orari di presenza al lavoro del personale e sulle retribuzioni associate a quegli orari.

La disponibilità di questi e altri linguaggi procedurali favorì enormemente lo sviluppo dei sistemi informatici per la gestione di strutture organizzative grandi e piccole, ma il lavoro di sviluppo continuava a restare lento e faticoso, perché comportava comunque un massiccio impiego di risorse umane.

Per rendere più spedito e più efficiente il lavoro intrinsecamente lento della scrittura dei programmi dei computer si definirono standard di scrittura e si cercò di individuare elementi ripetitivi nel lavoro di programmazione che potessero essere ricondotti a blocchi predefiniti di istruzioni, da utilizzare come "moduli" già pronti, da inserire nei nuovi sistemi gestionali – detti genericamente "applicazioni" – che si sviluppavano. Tipicamente, uno dei moduli che fu individuato per primo e si prestò meglio di tutti alla standardizzazione fu l'ordinamento. In quasi tutte le applicazioni era necessario ordinare in successione alfabetica o numerica stringhe di caratteri che rappresentavano matricole di personale o codici di materiali, quindi si crearono e si diffusero con successo programmi di servizio che ricevevano in input una descrizione schematica dei dati da ordinare e in base a quella descrizione acquisivano flussi di dati disordinati che restituivano come sequenze ordinate, pronte per successive elaborazioni di calcolo o di selezione.

Il successo anche commerciale di questi primi moduli di software utilizzabili con poche modifiche da chiunque dovesse sviluppare applicazioni che prevedevano di ordinare sequenze di dati condusse gradualmente a individuare altre caratteristiche delle applicazioni che si prestassero ad analoghe standardizzazioni.

Alla fine degli anni Sessanta, si individuò con chiarezza una struttura portante comune a tutti i sistemi informatici per la gestione aziendale, che venne chiamata genericamente "base dei dati" o *data base* in inglese. Col senno di poi sembra una banalità, ma si trattò di una intuizione luminosa che contribuì non poco a rendere più efficiente (e quindi meno costoso e più veloce) lo sviluppo delle applicazioni gestionali.

L'idea centrale è che qualunque applicazione gestionale fa riferimento a un insieme di dati, la sua base, che è opportuno gestire separatamente, con uno strumento generico che sia invariante rispetto al contenuto e alla tipologia dei dati, quindi utilizzabile sia da una banca per i suoi conti correnti sia da una catena di supermercati per i suoi magazzini. L'idea era quella di separare la gestione dei dati – acquisizione e manutenzione – dal loro utilizzo, consentendo a chi sviluppava le applicazioni di occuparsi soltanto dell'utilizzo dei dati e non della loro gestione/amministrazione: un altro importante passo avanti nella riduzione dei tempi (e quindi dei costi) dello sviluppo delle applicazioni.

Furono messi a punto svariati sistemi per la gestione delle basi di dati, chiamati genericamente *Data Base Management Systems*, in sigla DBMS, che furono promossi commercialmente con vigore non soltanto dalle società di sviluppo software che li avevano creati ma anche e soprattutto dalle imprese che producevano hardware, che vedevano giustamente

nei DBMS un fattore che avrebbe favorito l'adozione dei computer anche in imprese poco propense a farsi carico di notevoli costi per lo sviluppo del software applicativo. Nel 1970 un inglese naturalizzato americano studioso di logica e matematica di nome Edgar F. Codd pubblicò nella rivista istituzionale degli informatici USA (*Communications of the Association for Computing Machinery*), un articolo intitolato "A Relational Model of Data for Large Shared Data Banks" ("Un modello relazionale di dati per grandi banche dati condivise") che segnò la nascita formale di una nuova e più feconda concezione delle basi di dati, che prese il nome di *Relational Data Base Management System* ovvero RDBMS. Il termine "relational" sta a indicare l'elemento centrale della nuova concezione delle banche dati, concepite come insiemi di relazioni logiche fra dati elementari univoci che fisicamente si presentano come tabelle.

Nei dieci anni successivi alla pubblicazione del fondamentale contributo concettuale di Codd alcune società che realizzavano software lavorarono allo sviluppo di un linguaggio di programmazione che consentisse di creare RDBMS e di accedere agevolmente ai dati che contenevano: il risultato di questo lavoro di ricerca prende il nome di *Structured Query Language*, in sigla SQL, vale a dire "linguaggio per interrogazioni strutturate".

Con SQL si creano e si interrogano ("query" in inglese vuol dire domanda, interrogazione) RDBMS. Ciò su cui agisce lo SQL sono tabelle – idealmente identiche ai fogli di lavoro di Excel – composte da righe e colonne, secondo regole rigorose che vedremo in modo approfondito nei prossimi capitoli.

I linguaggi che si sono affermati nel tempo, il FORTRAN e il COBOL, di cui abbiamo già parlato, e molti altri ancora, sono accomunati da queste due caratteristiche: sono linguaggi generalizzati e procedurali.

- Un *linguaggio generalizzato* permette di far fare al computer qualsiasi cosa: eseguire calcoli matematici, acquisire dati da apparecchiature periferiche (tastiere, lettori di dischi ottici, schede di memoria) e salvarli su disco in forma di file, tracciare segni grafici su un monitor, emettere suoni e altro ancora.
- Un *linguaggio procedurale* è dotato di istruzioni e comandi con i quali si indica al computer come deve operare per arrivare al risultato che interessa al programmatore. Un linguaggio procedurale serve per scrivere *procedure*, che il computer esegue con sovrana imperturbabilità.

Lo Structured Query Language, invece, è un linguaggio specializzato e non procedurale.

- *Specializzato*: non si può usare SQL per fare grafica o per acquisire dati da trasferire su file, ma lo si usa soltanto per lavorare su tabelle.
- *Non procedurale*: con SQL non si creano procedure, non si spiega al computer come deve fare, ma gli si danno comandi che descrivono quello che si vuole ottenere. L'enfasi passa dal *come fare* al *che cosa ottenere*.

Naturalmente da qualche parte nel computer ci deve pur essere qualcosa che trasformi gli imperiosi comandi SQL in miriadi di istruzioni in linguaggio macchina affinché il computer riesca a dare i risultati che gli si chiedono, ma questo qualcosa è un *interprete* di SQL, scritto una sola volta per tutte da una squadra di sviluppatori di software di sistema. Chi vuole usare un database relazionale non deve preoccuparsi di come questo è fatto fisicamente, delle posizioni e dei collegamenti fisici dei record e delle tabelle. Il modello dei dati e della loro organizzazione è invariante rispetto alla loro realtà fisica: con SQL si accede ai dati e li si estrae sempre nello stesso modo, quale che sia la macchina e il database nel quale i dati risiedono.

## Lavorare con SQL ed Excel

Nato nel mondo dei grandi mainframe – dove tuttora domina come fondamentale strumento di lavoro – lo Structured Query Language è disponibile anche sui personal computer, nel cui mondo è nato Excel, che non è un linguaggio di programmazione ma un ambiente strutturato nel quale collocare dati per memorizzarli ed esaminarne le caratteristiche e le possibili affinità.

Nei fogli di lavoro Excel i dati si inseriscono per lo più a mano ma si possono anche acquisire insieme di dati già esistenti come file e inserirli in fogli di lavoro sui quali eseguire poi calcoli di varia natura – confronti, selezioni, ordinamenti – e ricavare illuminanti rappresentazioni grafiche dalle serie di dati contenuti in uno o più fogli di lavoro.

In Excel 2016 un foglio di lavoro può contenere oltre un milione di righe (per l'esattezza 1.048.576) e svilupparsi per 16.384 colonne. Sembrerebbe una capienza più che adeguata per acquisire e analizzare grandi insiemi di dati quali quelli che si generano nei supermercati o nelle banche, ma in realtà non sono sufficienti, perché quel genere di flussi di dati si sviluppa per decine di milioni di record nel giro di poco tempo, creando depositi sterminati di dati elementari chiamati “Big Data” che bisogna affrontare con gli strumenti di gestione tipici dei sistemi per database.

Si può concepire una comoda divisione del lavoro, quando è necessario lavorare sui Big Data: acquisirli e organizzarli in tabelle con l'aiuto di SQL e riversarli selettivamente (ricorrendo alle query di SQL) su un foglio di lavoro Excel nel quale lavorare di fino usando gli strumenti grafici di Excel per rappresentarli e metterli a confronto fra loro e ricorrendo agli strumenti di calcolo aritmetico e statistico di Excel per cercare di penetrare il significato di quelle masse di dati.

In questo libro presentiamo una rassegna completa dello Structured Query Language, fornendo anche indicazioni su come procurarsene una versione da usare su un personal computer.

Dopo questa rassegna, una serie di esempi concreti aiuta a capire come acquisire grandi flussi di dati con SQL e riversarli su fogli di lavoro Excel e quali strumenti intrinseci di Excel conviene usare caso per caso per analizzare quei dati e ricavarne opportune rappresentazioni grafiche.

### File degli esempi

All'indirizzo <http://www.apogeeonline.com/libri/9788850333745/scheda> i lettori potranno scaricare liberamente i file dei dati utilizzati per gli esempi mostrati nel corso del testo. Questi file possono essere utilizzati per fare pratica riproducendo le procedure di lavoro con SQL ed Excel.

*Marco Ferrero*  
ferrero\_m@libero.it  
Giugno 2016

