# Storia dell’SQL

"Va bene, in questo video daremo un'occhiata alla storia di SQL. Ora voglio illustrarti una serie di cose solo per mostrarti alcune delle origini di SQL e le cose con cui lavorerai che sono in circolazione ormai da moltissimo tempo.

Ho già menzionato nel 1970 il dott. Cod che pubblicò un articolo sul modello di dati relazionali. No, potrebbe essere finito lì, ma Donald Chamberlin e Raymond Voice iniziarono a lavorare su SQL, una prima versione di esso all'inizio degli anni settanta. Nel 1973, Chamberlin e i suoi colleghi iniziarono a lavorare su quello che chiamarono "sequel", e in questo momento era un acronimo che significava "linguaggio di query inglese strutturato". Ora hanno dovuto cambiarlo in SQL a causa di un conflitto di marchio con una compagnia aerea del Regno Unito.

Potresti chiederti se dico "sequel" o "SQL". Va bene, molte persone dicono entrambi. Alcuni dovrebbero dire che "SQL" è il modo corretto, altri dicono "sequel". Sono nel settore da sempre, e le persone onestamente usano entrambi. Non penso che ci sia un modo ufficiale giusto per dirlo. E se presti attenzione mentre seguo queste lezioni, mi sentirai andare avanti e indietro su "SQL" e "sequel". Penso di usare praticamente entrambe le pronunce, quindi non preoccuparti di quale sia quella giusta: SQL o sequel, va bene.

Ora, questo lavoro che stavano facendo su "sequel" ha iniziato a creare scalpore nel settore. L'azienda vide la promessa del modello di database relazionale alla fine degli anni '70. Questa azienda si chiamava Relational Software Inc e iniziò a lavorare su un sistema di gestione di database relazionale basato su "sequel" e lo vendette a diverse agenzie governative. Ora, nel 1979, la Relational Software pubblicò il suo primo software relazionale."

"il sistema di gestione di database relazionale disponibile in questa azienda in seguito ha cambiato nome in Oracle, e potresti aver sentito parlare di Oracle se sei nel campo della tecnologia. Scommetto che potresti averlo, sono assolutamente enormi.

Subito dopo quella versione del 1979, IBM ha rilasciato un sistema chiamato SQL D, il nostro sistema di dati seguito che ebbe vita molto breve. Onestamente, non ricordo di averne mai sentito parlare molto. Stavo facendo ricerche per lo slide deck e questa era una novità per me.

Nel 1985, IBM pubblicò db2, e questo è ancora il marchio di IBM, il loro sistema di gestione di database relazionali. Ora è abbastanza interessante che nel 1986 DARPA ha iniziato a lavorare sul progetto Berkeley Postgres. Se sei un nerd della tecnologia, iniziano a lavorare su quel progetto. Vedrai questo evolversi più tardi in quello che sappiamo come PostgreSQL oggi.

Nel 1987, Microsoft ha rilasciato la sua prima versione del server sequel. Quindi, sorpreso dalla cronologia, non mi ero reso conto che un server sequel risaliva a quel periodo. Nel 1993, le tecnologie informatiche Olestra commercializzarono il progetto Berkeley Postgres. Quindi presero il corpo di lavoro che veniva svolto sotto la DARPA e lo commercializzarono, il che significa che hanno creato un prodotto commerciale da esso, un'offerta. Ed è una specie di fork del codice, e quel codice è stato ora fuso in Informix. E se Informix è un molto popolare database, ma, per quanto ne so, sono vicini e commerciali, quindi hai bisogno di una licenza per lavorarci e di una sorta di nicchia. Informix, negli anni '90 e all'inizio degli anni 2000, era un contendente che andava contro Oracle, ma ora ha una quota di mercato molto piccola.

Nel 1994, Postgres di Valve è diventato un progetto open source, e l'hanno chiamato Postgres 95. Aveva molte funzionalità ed è stato un allontanamento significativo da dove si trovava. Stavano cercando di ottenere un seguito più ampio e più o meno nello stesso periodo, una società svedese chiamata MySQL rilasciava MySQL per uso interno era destinato a se stessi e alcuni clienti interni, e altri sviluppatori originali inclusi un ragazzo di nome Michael Monty. Penso che tu dica Monty, ma potrei sbagliarmi nel pronunciare il suo nome. David Axmark e Allan Larson sono alcuni degli sviluppatori originali. Fai attenzione che sentirai ancora il mio nome, Monty, un po'. Non sono sicuro se ho detto il mio esercito, ora.

Nel 1996, Postgres 95 è stato rinominato PostgreSQL, ed è ancora ampiamente diffuso e accettato. Io uso principalmente solo Postgres. Vedrai nella loro documentazione ufficiale che si riferiscono al prodotto come PostgreSQL, e questo è ancora il nome ufficiale nelle citazioni aeree. Ma io e la maggior parte delle persone nel settore ci riferiamo semplicemente a Postgres, è molto più semplice di PostgreSQL e risulta un po' più semplice. Sai, poco dopo che MySQL diventa un database open source sotto GPL, che è una licenza molto popolare e ti dà molta libertà. È davvero qualcosa che ha fatto decollare la mia scala.

Nel 2001, MySQL aveva oltre 2 milioni di installazioni attive, e poco dopo, nel 2005, Oracle ha acquisito InnoDB, che è la società dietro il motore di archiviazione del backend di MySQL. Quindi, ho fatto conoscere molte persone nella comunitàErano nervosi per ciò che Oracle potrebbe fare con questo, ma si è rivelato molto preoccupato per nient'altro. MySQL ha un motore di archiviazione collegabile, quindi puoi configurarlo per utilizzare diversi backend di archiviazione. Un approccio interessante che hanno lì.

È stato comunque un'opzione molto popolare per loro. Tra il 2001 e il 2006, è cresciuto da 2 milioni di installazioni attive a oltre 8 milioni, e l'azienda aveva 320 dipendenti in 25 paesi. Stavano crescendo rapidamente e ottenendo molte quote di mercato e buzz nel mercato dei database, sia open source che, non dimenticare, anche se è open source e come sviluppatori potremmo considerare che è un database gratuito, quando guardi le grandi aziende, loro cercano un supporto commerciale dietro. Quindi, è così come aziende come MySQL guadagnano soldi.

Sono ancora open source per un prodotto e dicono, 'Ehi, puoi scaricarlo, funziona con quello che vuoi'. Ma se hai bisogno di supporto e qualcuno che assicuri che funzioni, scrivilo. E se qualcosa va storto di notte, vogliamo qualcuno che chiamato è lì che inizia il supporto commerciale. Nel 2008 stavano diventando molto popolari e Sun Microsystems acquista MySQL. È un'acquisizione importante e causa alcuni cambiamenti a valle perché poco dopo Oracle acquisisce Sun Microsystems e con esso ottengono MySQL.

Questo ha causato molte polemiche nella comunità, perché Oracle sta semplicemente dominando il mercato dei database commerciali, e la gente era preoccupata che Oracle lo stesse facendo all'open source di MySQL. Uno dei fondatori originali, Monty, ricorda che ti avevo detto di ricordare quel nome. In realtà, ha lasciato Sun Microsystems perché è diventato un dipendente con l'acquisizione e ha sviluppato un fork del codice di MySQL chiamato MariaDB. Se ricordo bene, Maria è una delle altre figlie di Michael, quindi sembra che ci sia uno schema nel nominare i database in base a quello delle sue figlie.

Come ho detto, c'è una preoccupazione per il futuro di MySQL, e MariaDB API rimane compatibile con MySQL, quindi ottieni un alto grado di compatibilità lì. Quindi, come sviluppatore o qualcuno che ci lavora, non dovresti vedere molta differenza funzionale mentre lavori con quel database, e MariaDB sta diventando molto popolare. Quindi, se hai bisogno di usare MySQL o MariaDB, una volta che ti sei alzato e hai iniziato a correre, non dovresti vedere molta differenza.

Ora, la prossima cosa di cui dobbiamo parlare è ANSI, che sta per American National Standards Institute. L'ANSI ha pubblicato una serie di standard per il linguaggio SQL, e questi standard si sono evoluti. Questa è una specie di area in cui troverai molta frustrazione come qualcuno che lavora con diversi fornitori di database, perché il rapporto del database relazionale supporta l'ANSI e ancora, le citazioni aeree e il supporto in una certa misura sono migliorati molto. Ma ci sono ancora delle differenze.

Ci sono differenze nell'interpretazione degli standard ANSI tra i fornitori, quindi ci sono alcune funzionalità su quel fornitore di database che sono semplicemente diverse e non sono conformi allo standard. Quindi, è uno standard ampiamente adottato in teoria, dovresti essere in grado di scrivere ANSI SQL ed eseguirlo su qualsiasi database, ma ci sono stranezze e trucchi che vedrai da Oracle a Postgres a MySQL. Quindi, vedrai delle stranezze tra i database, ma per dare credito ai fornitori, è molto meglio oggi rispetto a dieci anni fa. Questo è certo, ma sono ancora presenti piccole stranezze perché tutti i fornitori di database hanno il loro stile SQL e poi hanno funzionalità conformi con ANSI SQL. Quindi è importante ricordarlo mentre ti muovi tra i fornitori di database.

Ora diamo una rapida occhiata alla cronologia delle versioni. Nel 1986, il primo standard SQL fu formalizzato da ANSI. Questo visse brevemente, subito dopo nel 1989 uscì SQL 89, e questo aggiunse vincoli di integrità e in un certo senso ripulì alcune carenze. In un certo senso, fece davvero maturare quella versione iniziale più per il settore.

Ora esce SQL 92 nel 1992. Questa è un'importante revisione di molte funzionalità che mancavano, e ciò che hanno inserito era una data di conformità molto più rigorosa. O almeno così ricordo. Stavo dicendo che le interpretazioni dello standard variavano ampiamente tra diversi fornitori di database o diverse piattaforme di database. Questo non è stato d'aiuto per rafforzarlo.

Poi, nel 1999 e da qualche parte tra il 1990 e il 1992, è stato quando ho iniziato personalmente a lavorare con SQL. Quindi ho avuto la possibilità di vedere un bel po' questo evolvre nel tempo. Nel '99, sono state aggiunte le espressioni regolari, creando molta confusione. Quando ottieni il primo approccio che usavi per valutare l'espressione della ricerca di cose e i trigger del database, che è una funzionalità interessante. Vedremo sicuramente quelli in arrivo, ed è lì che puoi dire se sta succedendo qualcosa nel database, come un inserimento per fare qualcosa. Puoi eseguire un piccolo evento di programmazione su questo.

Molte più funzionalità sono state aggiunte in SQL '99, che è stata una serie piuttosto ampia del popolarissimo sequel. Nel 2003, esce il supporto XML e colonne con valori generati automaticamente. Questo sembra una bella funzionalità. E dando un esempio, MySQL ce l'ha da molto tempo. Oracle, fino a poco tempo fa, non lo aveva. Quindi, il supporto dello standard ANSI è qualcosa che si evolverà, se ricordo bene. Molte funzionalità di generazione non sono in Oracle 11, ma sono state aggiunte in Oracle 12, cose piuttosto recenti.

La timeline tecnologica continua. Nel 2006, SQL 2006 esce con più supporto XML e funzionalità specifiche per l'utilizzo di SQL con XML. Quindi, molta attenzione su come utilizzare XML in questo lasso di tempo. Nel 2008, esce SQL 2008, aggiungendo invece dei trigger, come qualcosa che vedremo sicuramente nell'istruzione trunca. Ora, questa è un'affermazione molto pericolosa quando tronchi una tabella. In pratica, dice di svuotare quella tabella, strappando tutti i dati dalla tabella in modo così definitivo. Usa questo con attenzione, è potenzialmente pericoloso.

Dovrei dire che SQL 2011 esce senza molte funzionalità per i dati temporali, quindi questa è una specie di area interessante su cui lavorare. Infine, nel 2016, esce SQL 2016, e in questo annuncio c'è la corrispondenza del modello di riga. Puoi usare le espressioni regolari per abbinare le righe, che è una caratteristica interessante. Le funzionalità di JSON iniziano ad essere aggiunte allo standard, e ciò che è interessante è che l'industria IT nel suo complesso ha iniziato a migrare da XML a JSON. Quindi, diversi anni fa XML era molto popolare, ora JSON è molto popolare. Puoi vedere che lo standard si è evoluto con esso. Come ho detto, lo standard è uscito ed è maturato, e poi ha iniziato ad evolversi con il settore e le tendenze del settore, come XML e JSON. Quindi puoi vedere come è successo.

Ma tieni presente che ANSI è uno standard. Esamineremo la sintassi SQL ANSI durante il corso, ma tieni anche presente che ogni piattaforma di database avrà livelli diversi di supporto nella loro interpretazione di quello standard e della loro lingua. Quindi, le loro caratteristiche e quant'altro potrebbero non essere compatibili con quello standard. Vedrai differenze tra le piattaforme di database, anche se esiste