

Confronto tra un DBMS relazionale e un File System*

Oreste Signore, Vito Cusumano

CNUCE, Istituto del CNR - via S. Maria, 36 - 56126 Pisa

Sommario

Un DBMS e un File System sono stati posti a confronto sia sulla base della documentazione, che per quanto riguarda le prestazioni. A tale scopo, è stata progettata e realizzata una base di dati di prova, sono state sviluppate applicazioni mediante gli strumenti forniti dai due prodotti, sono stati popolati i due archivi con dati generati in modo casuale. I risultati ottenuti indicano come il rispetto del modello relazionale e degli standard internazionali, e soprattutto la garanzia dell' integrità dei dati, comportino un appesantimento delle prestazioni e un incremento delle risorse richieste, per cui la scelta del DBMS relazionale potrebbe, in qualche caso particolare, venir messa in discussione.

1 - Introduzione

Il confronto tra il DBMS Oracle e il file system Information è stato effettuato utilizzando un computer Prime 9955, da 5 MIPS, con architettura a 32 bit. Le versioni utilizzate sono il release 5.1.22.3 di Oracle, e il release 7.8 di Information.

I due sistemi sono stati confrontati per tre aspetti:

- l' architettura generale;
- la facilità di sviluppo di applicazioni mediante i tool forniti dai due sistemi;

* Questo lavoro è stato finanziato da una borsa di studio offerta dalla Prime Computer.

- i tempi di risposta a fronte di alcune query, eseguite sulla base di dati popolata con dati generati in modo casuale.

2 - I sistemi considerati

2.1 - Le caratteristiche generali

Oracle è composto da un nucleo base (kernel) corredato da una serie di interfacce e programmi di utilità, tra i quali i più importanti sono SQL*PLUS, SQL*FORMS, SQL*REPORT, PRO*C, PRO*COBOL, PRO*FORTRAN, ODL, IMP, EXP.

Information è un sistema della IV generazione, implementazione della Prime del sistema Pick, commercializzato come relazionale. E' basato su file discreti, di cui l'utente può avere visione, e può accedere ad archivi esterni. Per applicazioni non troppo complesse è utilizzabile un processo di esecuzione, denominato SIMPLE, che permette di definire applicazioni interattive che fanno uso di maschere video. Tali maschere sono anche richiamabili da programmi scritti in INFO/BASIC. Il sistema dispone anche di una interfaccia che funge da interprete dei comandi, denominata PERFORM. Un'altra interfaccia, CONNECTION, permette all'utente di utilizzare i tool Financial (spreadsheet, funzioni matematiche, statistiche e finanziarie), Graph (creazione di grafici a partire dai dati contenuti nella base di dati), Word (preparazione di report). Il linguaggio di interrogazione è INFORM, ma si prevede che nelle prossime versioni esso venga affiancato dal linguaggio SQL.

2.2 - Il DDL

Oracle aderisce sostanzialmente ai vincoli imposti dal rispetto del modello relazionale, con le limitazioni caratteristiche di quasi tutte le implementazioni di tipo commerciale, e quindi:

- tutte le operazioni del DDL vengono gestite completamente mediante il linguaggio SQL;
- la sintassi dei comandi è quella dell' SQL standard;
- i tipi di dato definibili sono quelli previsti dallo standard ISO 9075, arricchiti con il tipo data;
- non è ovviamente possibile definire attributi multivalore;

- non è disponibile il vincolo di unicità del valore di un certo attributo, a meno che non si costituisca un indice su tale attributo;
- non è possibile definire il vincolo di referential integrity, e solo mediante opportuni artifici (creazione di un indice UNIQUE) è possibile definire il vincolo di entity integrity;
- le operazioni permesse su una vista non sono univocamente determinate, ma dipendono dalle definizioni degli attributi della tavola base.

Per quanto concerne Information:

- alcune funzioni sono supportate dall' interfaccia PERFORM;
- gli attributi possono essere multivalore;
- il vincolo di unicità della chiave viene garantito automaticamente dal sistema;
- il vincolo di valore non nullo è gestito automaticamente sulla chiave primaria, ma non è specificabile per nessun altro attributo;
- è possibile controllare i vincoli di entity integrity e di referential integrity;
- non è agevole definire una vista;
- il meccanismo delle autorizzazioni è basato sul sistema operativo.

2.3 - Il DML

2.3.1 - Modifica dei dati

Per quanto riguarda le potenzialità e funzionalità, i due sistemi sono sostanzialmente equivalenti per aggiornamenti, inserimenti e cancellazioni. A vantaggio di Oracle giocano anche in questo caso il rispetto del linguaggio SQL e il controllo dell' accesso ai dati da parte del DBMS stesso.

Information presenta come punti di forza la flessibilità nel trattare i file di dati da caricare, la semplicità nell' effettuare inserimenti e aggiornamenti, il controllo di unicità della chiave, mentre presenta delle limitazioni per quanto riguarda il controllo degli accessi, che è totalmente a carico del sistema operativo.

2.3.2 - Il recupero dei dati

In Oracle il recupero dei dati avviene mediante la formulazione di query nel linguaggio SQL standard, sia in interattivo che nei programmi applicativi, mediante l' utilizzo della clausola: SELECT ... FROM ... WHERE ...

Vanno segnalati la possibilità utilizzare fino a sedici livelli di sottoquery, la possibilità di effettuare il join di una relazione con se stessa, il supporto dell' outer join, la gestione

delle operazioni relazionali di unione, intersezione e differenza, e l'estensione dello standard SQL per il supporto di interrogazioni di relazioni contenenti informazioni gerarchiche (esempio tipico la distinta base). Va tuttavia tenuto presente che l'ordine di specifica degli attributi nei predicati può influire sulla velocità di esecuzione della query, e che la scelta di particolari funzioni può determinare il mancato utilizzo degli indici. Infine, ma questo è un limite del linguaggio SQL, è compito del programmatore scegliere, tra le query semanticamente equivalenti, la formulazione sintattica più adatta.

La modalità di interrogazione adottata da Information non permette di effettuare join, e una query non può contenere sottoquery. Un (equi-)join può di fatto essere realizzato utilizzando un meccanismo di attivazione di liste. E' però possibile specificare condizioni sui campi multivalore, e rendere visibili o tutte le istanze, o solo quelle che soddisfano la condizione.

2.4 - Gli indici

Gli indici di Oracle, strutturati a B*-tree, possono essere definiti su attributi singoli o su più attributi concatenati, ed è anche possibile definirli con l'opzione UNIQUE, che garantisce l'unicità dei valori assunti, o con l'opzione COMPRESS, che riduce l'occupazione di spazio, anche se comporta un appesantimento in fase di ricerca. Gli indici vengono mantenuti costantemente aggiornati e consistenti con il contenuto della base di dati.

Gli indici di Information, anch'essi strutturati a B*-tree, a parte quello sulla chiave primaria, che è un indice di tipo hash, possono venir definiti sia sui campi primitivi che su campi derivati. In questo caso, tuttavia, eventuali aggiornamenti dei campi che fanno parte dell'espressione che definisce il tipo derivato, non vengono riflessi sull'indice, per cui si possono verificare casi di puntamenti inconsistenti. L'aggiornamento degli indici su campi primitivi è invece assicurato in modo coerente, ma è possibile disabilitarlo temporaneamente, in caso di aggiornamenti di massa, per poi ricostruire l'indice completo. Quando l'indice è disabilitato, si possono verificare altri casi anomali, legati a inserimento-cancellazione di record o a modifiche del valore della chiave, in cui il contenuto dell'indice non è coerente con quello del file.

2.5 - Le tecniche di ottimizzazione

Le caratteristiche del linguaggio SQL rendono problematica la realizzazione di un buon ottimizzatore, cioè di quella parte del DBMS che si incarica di determinare, a fronte di

una query, qual è il metodo di esecuzione della query che comporta il minor numero di accessi e quindi il peso minimo in termini di risorse richieste. L' ottimizzatore di Oracle utilizza o meno gli indici a seconda del predicato specificato, e realizza il join in modi diversi (tabella guida o sort-merge) a seconda della presenza o meno di indici sugli attributi di join. Il cammino scelto è comunque influenzato dall' ordine in cui vengono specificate le tabelle nella clausola SELECT ... FROM.

Nel caso di Information, il processo di ottimizzazione si limita esclusivamente a controllare se esiste un indice su un attributo del predicato, e decidere se utilizzarlo o meno. E' possibile che clausole equivalenti, ma con predicati scritti in ordine diverso, richiedano tempi di esecuzione differenti.

2.6 - Integrità e sicurezza dei dati

In Oracle, come in tutti i DBMS, l' integrità logica e fisica dei dati viene assicurata mediante i meccanismi di COMMIT e ROLLBACK, e mediante l' esistenza di procedure di Recovery automatico. Questi metodi comportano la presenza di file Before Image e After Image, su cui registrare i record prima e dopo gli aggiornamenti, e di un file di Log, su cui vengono registrate le marche di checkpoint. Grazie a questi meccanismi, l' integrità logica e fisica della base di dati è sempre garantita.

La protezione da accessi non autorizzati è gestita con i meccanismi di GRANT e REVOKE.

La correttezza dei dati, intesa come rispetto dei vincoli, non è gestita in modo automatico, in quanto questi non sono definibili nello schema, ma, in ambiente SQL*FORMS, è disponibile il meccanismo dei trigger, che possono essere definiti a livello di singolo campo o di mappa, e fanno scattare la verifica delle condizioni imposte dai vincoli ogni volta che il campo viene manipolato. Un aspetto negativo è costituito dal fatto che i trigger vengono associati alla particolare mappa video, e non al campo in quanto tale, per cui vanno specificati più volte, con possibili rischi di inconsistenza e disallineamento.

Infine, nel caso di accesso simultaneo da parte di più applicazioni, è possibile ad un processo bloccare la tabella, impedendone la modifica da parte di altri utenti. Se il sistema individua una situazione di stallo, fa abortire la transazione che ha effettuato la minor quantità di lavoro (ROLLBACK), permettendo a tutti gli altri processi di continuare il loro lavoro.

La gestione dell' integrità fisica e logica da parte di Information è decisamente carente, in quanto l' unica possibilità offerta all' utente è quella di attivare un disco di copia, per

cui tutte le modifiche apportate vengono registrate su due dischi fisicamente diversi. Tale soluzione garantisce l' integrità fisica, ma non quella logica. Gli accessi concorrenti vengono gestiti utilizzando le primitive messe a disposizione dal sistema operativo, con conseguente notevole aggravio di lavoro per l' utente.

2.7 - La struttura fisica

Tutti i dati gestiti da Oracle risiedono in una o più entità di spazio logico, dette partizioni. Ogni partizione è fisicamente residente su uno o più file. Ogni file è composto da una o più estensioni, e queste ultime sono composte da uno o più blocchi. Ogni partizione può contenere una o più tabelle, ma nessuna tabella si può estendere su più di una partizione. L' aggiunta di estensioni alle partizioni, necessaria nel caso di esaurimento dello spazio disponibile, fa sì che lo spazio occupato da una partizione non corrisponda più ad una porzione di spazio disco fisicamente contiguo. I record vengono memorizzati a lunghezza variabile, ma lo spazio reso disponibile a seguito di cancellazioni o di aggiornamenti di campi carattere a lunghezza variabile non viene riutilizzato se non a seguito di una operazione di riorganizzazione dei dati. Per utilizzare in modo ottimale lo spazio disco, è necessaria una valutazione accurata delle dimensioni e del numero di tuple che costituiranno il database.

E' da segnalare la possibilità di definire dei cluster, ossia delle associazioni che legano fisicamente più tabelle, in modo da mantenerle contigue, garantendo così una maggiore efficienza del join. Questo accorgimento, che impatta unicamente sugli aspetti fisici, ma è completamente trasparente all' utente, risulta particolarmente utile quando sia possibile individuare un cammino preferenziale di accesso ai dati.

Information immagazzina i dati in file discreti, e quindi ogni entità, con tutti i suoi attributi multivalore, può essere rappresentata in un singolo file. Ogni record è memorizzato come una stringa la cui lunghezza massima è di 32k; ogni campo è a lunghezza variabile, ed è separato dal successivo da speciali marche: lo spazio occupato è quello strettamente necessario. Anche in questo caso, comunque, lo spazio reso disponibile a seguito di cancellazioni o aggiornamenti è riutilizzabile solo dopo una operazione di compattamento.

3 - Il database di test

3.1 - Descrizione dell' applicazione

L' applicazione sviluppata per confrontare i due sistemi fa riferimento alla gestione delle informazioni di una concessionaria di auto, sia nuove che usate, di modelli e costruttori differenti.

Le operazioni previste nell' attività commerciale della concessionaria sono la vendita di auto nuove o usate, con eventuale ritiro e valutazione di un' auto usata.

Per ogni auto trattata, vengono gestite informazioni quali il numero di telaio, la data di immatricolazione, la targa, i numeri delle pratiche che l' hanno gestita, l' attuale posizione (venduta o disponibile), gli optional disponibili su quel modello, con relativo prezzo e se sono montati di serie, e, per le auto usate, le condizioni e i chilometri percorsi.

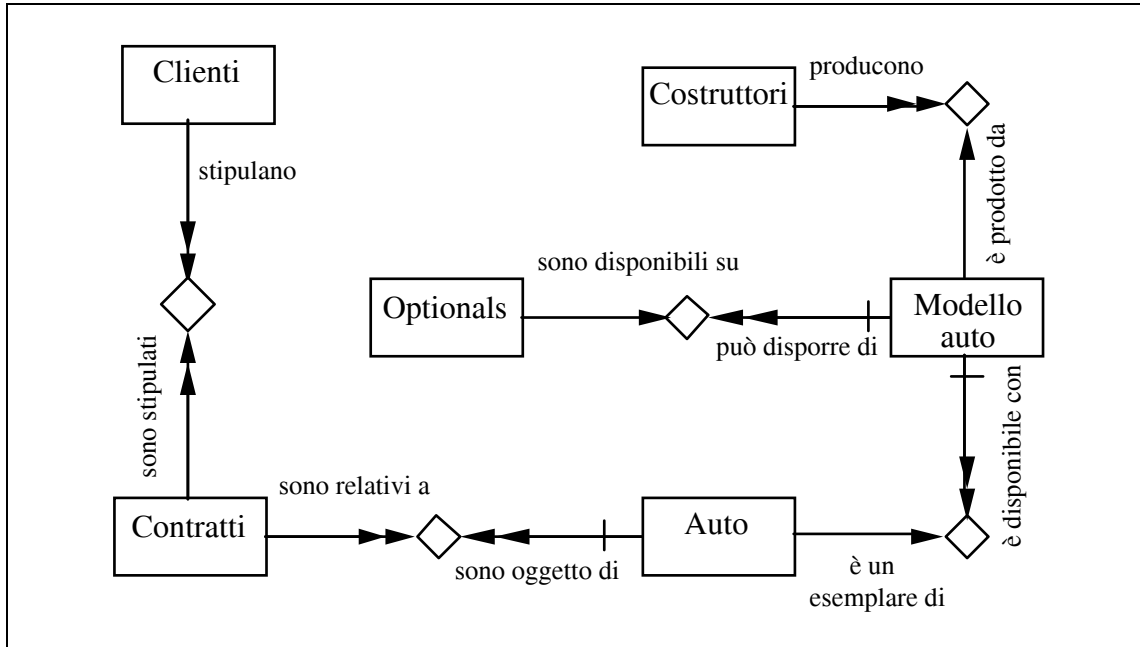
Per ogni modello, le informazioni sono quelle relative alle caratteristiche proprie del modello (dimensioni, consumo, velocità massima), il prezzo come da listino base, il costruttore.

Infine, le informazioni relative ai contratti sono le generalità del cliente, il tipo di contratto (vendita o permuta), data e importo, numero di telaio dell' auto venduta, e, in caso di contratto di permuta, il numero di telaio dell' auto ceduta e la relativa valutazione.

Le applicazioni devono essere realizzate in modo da garantire i vincoli, in particolare quelli relativi alla chiave primaria, alle chiavi esterne, ai possibili valori dei singoli attributi.

3.2 - Lo schema concettuale della base di dati

Lo schema concettuale della base di dati è rappresentato in figura, mediante un formalismo Entità-Relazioni:



3.3 - Lo schema logico

3.3.1 - Ambiente Oracle

Lo schema logico relazionale, ottenuto mediante l' applicazione delle consuete regole di traduzione utilizzate per trasformare le relazioni n:m e gli attributi multivalore, è quello costituito dalle seguenti relazioni, in cui gli attributi costituenti la chiave primaria sono sottolineati, e le chiavi esterne vengono evidenziate mediante il corsivo:

CONTRATTI (Numero-Pratica, *CF-Cliente*, Vendita, Permuta, Data-Contratto, Importo, Num-Tel-Vend, Num-Tel-Acq, Val-Usato)

CLIENTI (CF-Cliente, Nome, Indirizzo)

TEL_CLIENTI (*CF-Cliente*, Telefono-Cliente)

COSTRUTTORI (CF-Costruttore, Nome, Indirizzo)

TEL_COSTR (*CF-Costruttore*, Telefono-Costruttore)

AUTO (Num-Telaio, Data-Imm, Posizione, *Modello*, Usata/Nuova, Targa, Condizioni, Km-Percorsi))

PRAT_AUTO (Num-Telaio, Num-Pratica)

MODELLI (Nome-Modello, Lunghezza, Larghezza, Max-Vel, Consumo, Prezzo-Base, CF-Costruttore)

OPTIONAL (Optional, *Modello*, Prezzo, Di-serie?)

Per ragioni di efficienza, sono state poste in cluster le tabelle:

- CLIENTI e TEL_CLIENTI;
- COSTRUTTORI e TEL_COSTRUTTORI
- AUTO e PRAT_AUTO

Per annullare gli effetti della normalizzazione, che impone l'eliminazione degli attributi multivalore e la creazione di tabelle contenenti i valori assunti dagli attributi, sono state definite le viste I_CLIENTI e I_COSTRUTTORI e NS_AUTO (che associa ad ogni auto le pratiche che l'anno tratta), mentre per pura comodità implementativa sono state definite le viste COSTO_COMP e STORIA_AUTO, di ovvio significato.

3.3.2 - Ambiente Information

In Information, la possibilità di gestire attributi multivalore ha permesso di definire un file per ogni entità presente nello schema concettuale: CONTRATTI, CLIENTI, COSTRUTTORI, AUTO, MODELLI, OPTIONAL. I vincoli sui valori possibili degli attributi (compreso il pattern del Codice Fiscale), sono stati definiti a livello di dizionario.

Anche in questo caso sono state definite le viste COSTO_COMP e STORIA_AUTO

4 - Le prestazioni

Al fine di valutare le caratteristiche degli strumenti messi a disposizione dai due sistemi per la realizzazione di applicazioni, è stata sviluppata la medesima procedura sia in ambiente SQL*FORMS che in ambiente SIMPLE.

Non sono state riscontrate differenze sostanziali per quanto riguarda i tempi di risposta e l'aspetto esteriore delle applicazioni realizzate, anche se in SQL*FORMS è apparso un po' farraginoso il meccanismo dei trigger, che, come si è già detto, non possono essere definiti a livello globale, ma sono associati al particolare campo della particolare mappa.

Da questo punto di vista, la definizione dei vincoli a livello di schema, possibile in Information, costituisce un innegabile vantaggio.

L'aspetto più delicato dal punto di vista delle prestazioni era evidentemente costituito dai tempi necessari a rispondere a query non banali.

Per poter rilevare i tempi di CPU a fronte di varie query, sono stati utilizzati degli algoritmi che hanno permesso di generare dati casuali, che tuttavia rispettavano i vincoli, da inserire nelle tabelle di Oracle e nei file di Information. I dati sono stati generati secondo una legge di distribuzione di tipo poissoniano.

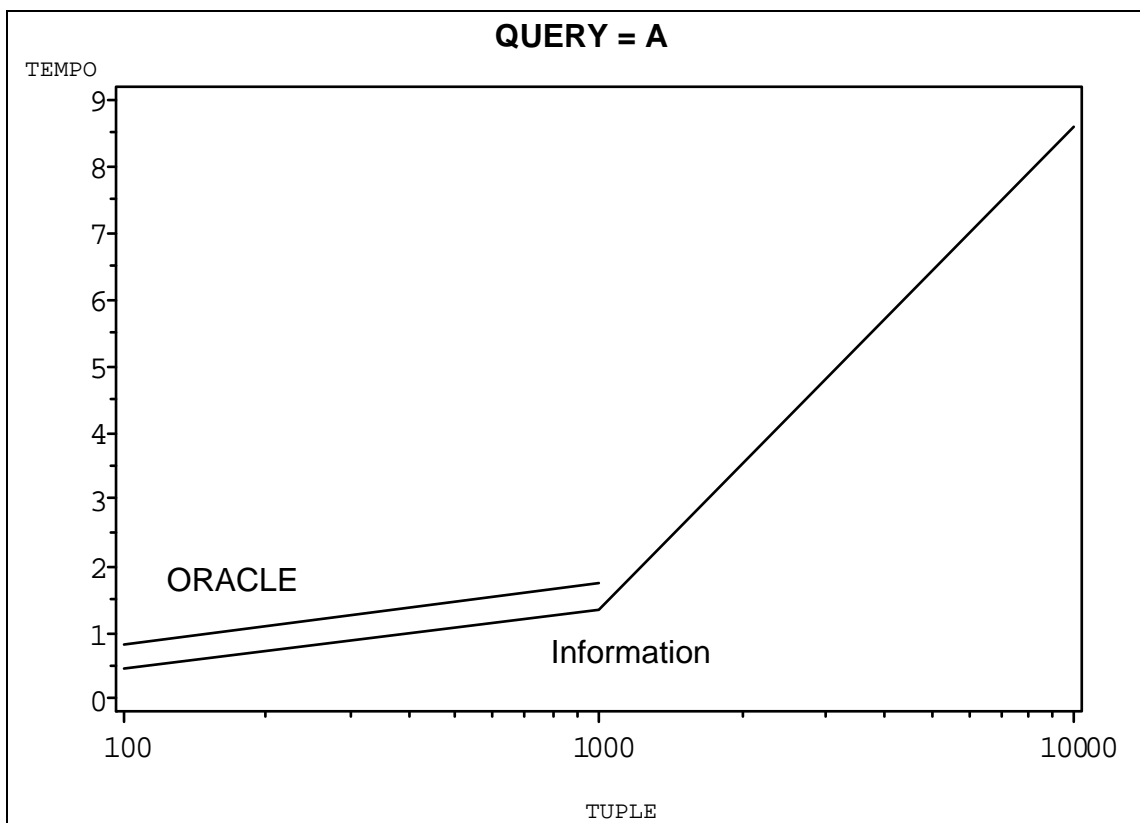
Le operazioni su cui sono stati valutati i tempi, sono:

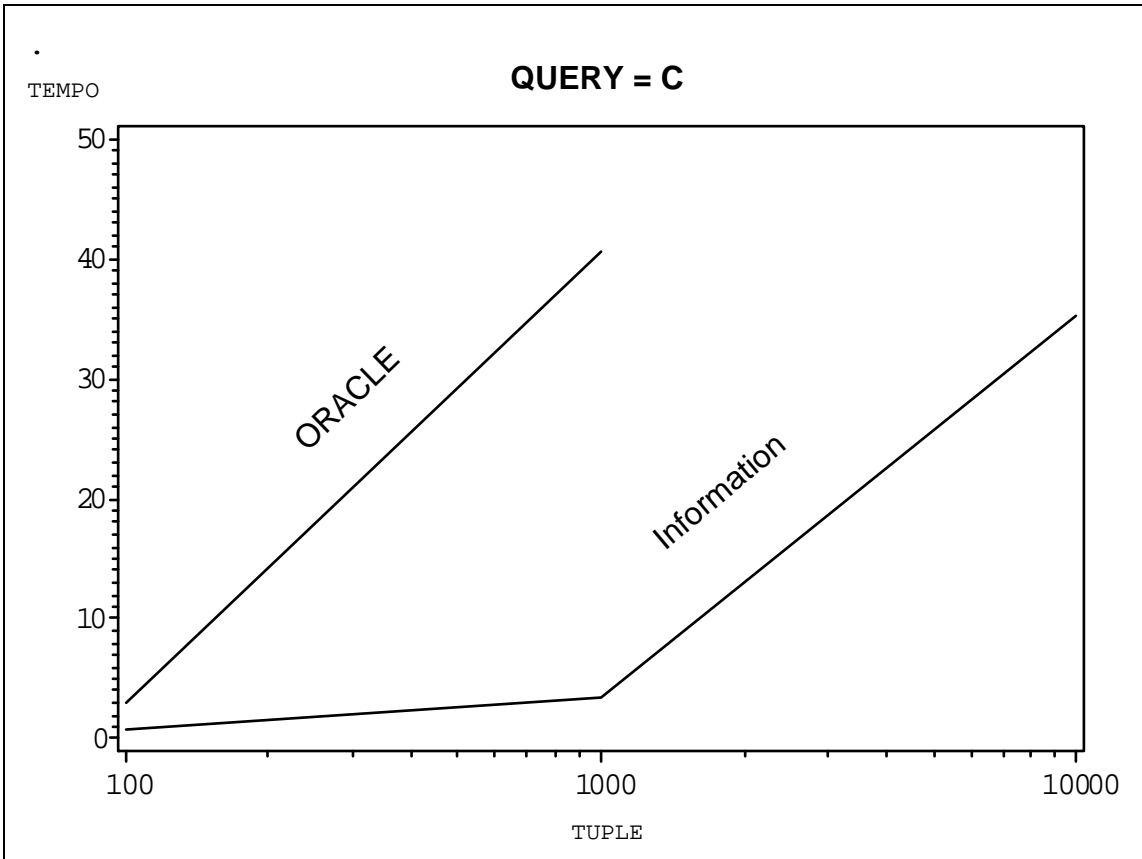
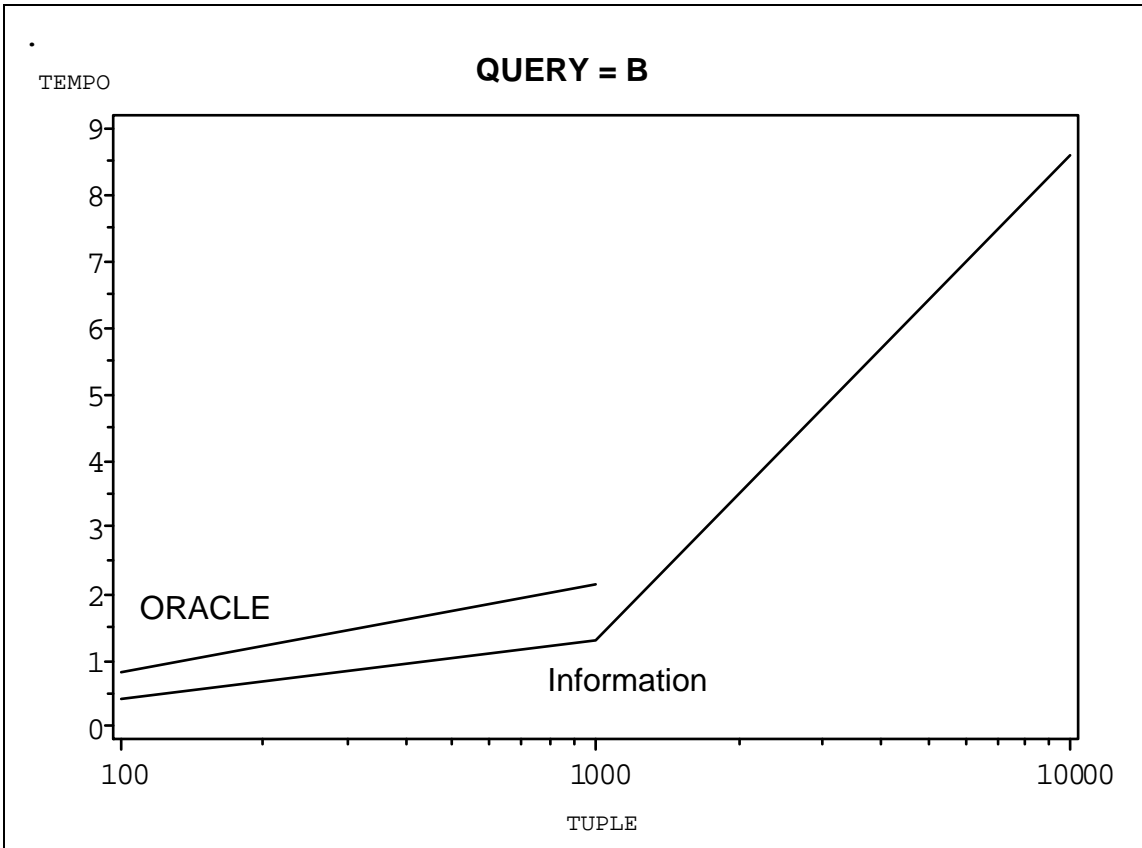
- a) ricostruzione della storia di un' auto trattata;
- b) ricostruzione della storia di un' auto trattata, con report ordinato per data contratto;
- c) generalità dell' attuale proprietario di un' auto;
- d) generalità del costruttore dell' auto acquistata da un certo cliente;
- e) prezzo complessivo di un' auto, su cui sono montati un certo numero di optional;
- f) inserimento di un nuovo cliente;
- g) creazione di un indice sul campo Nome della classe CLIENTI.

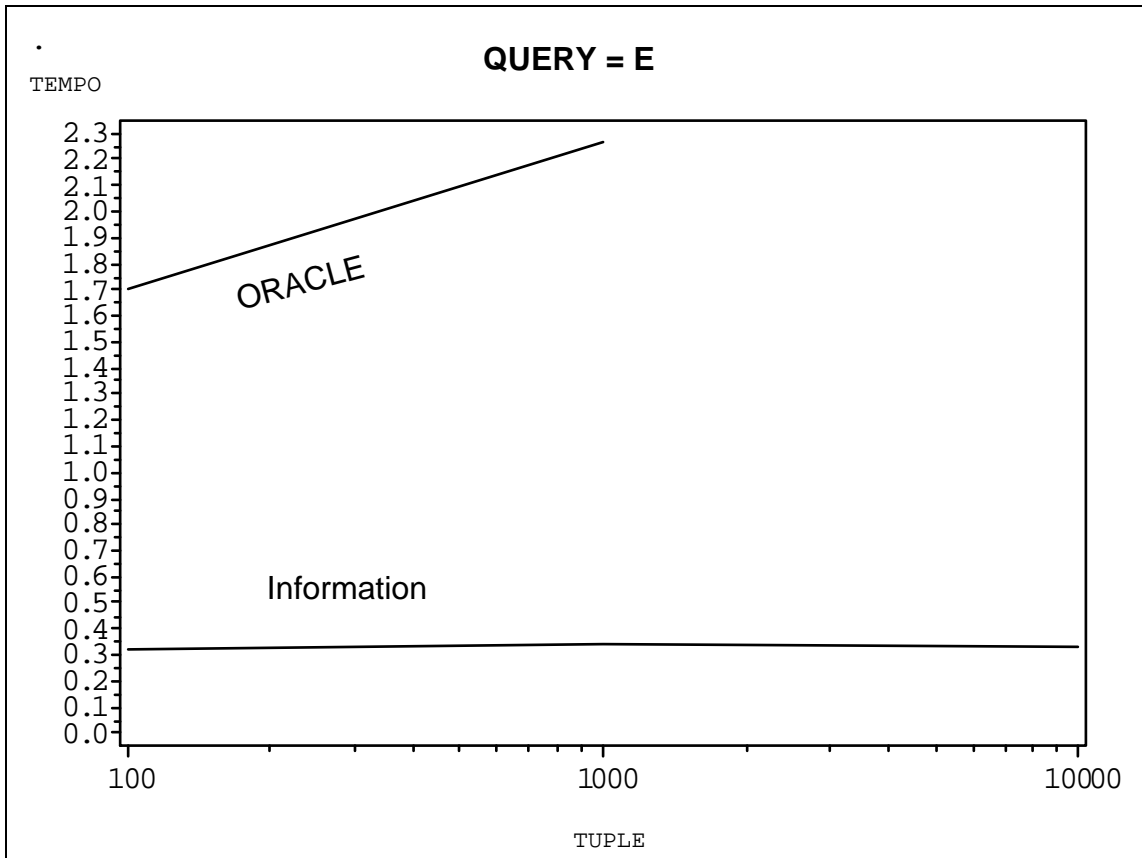
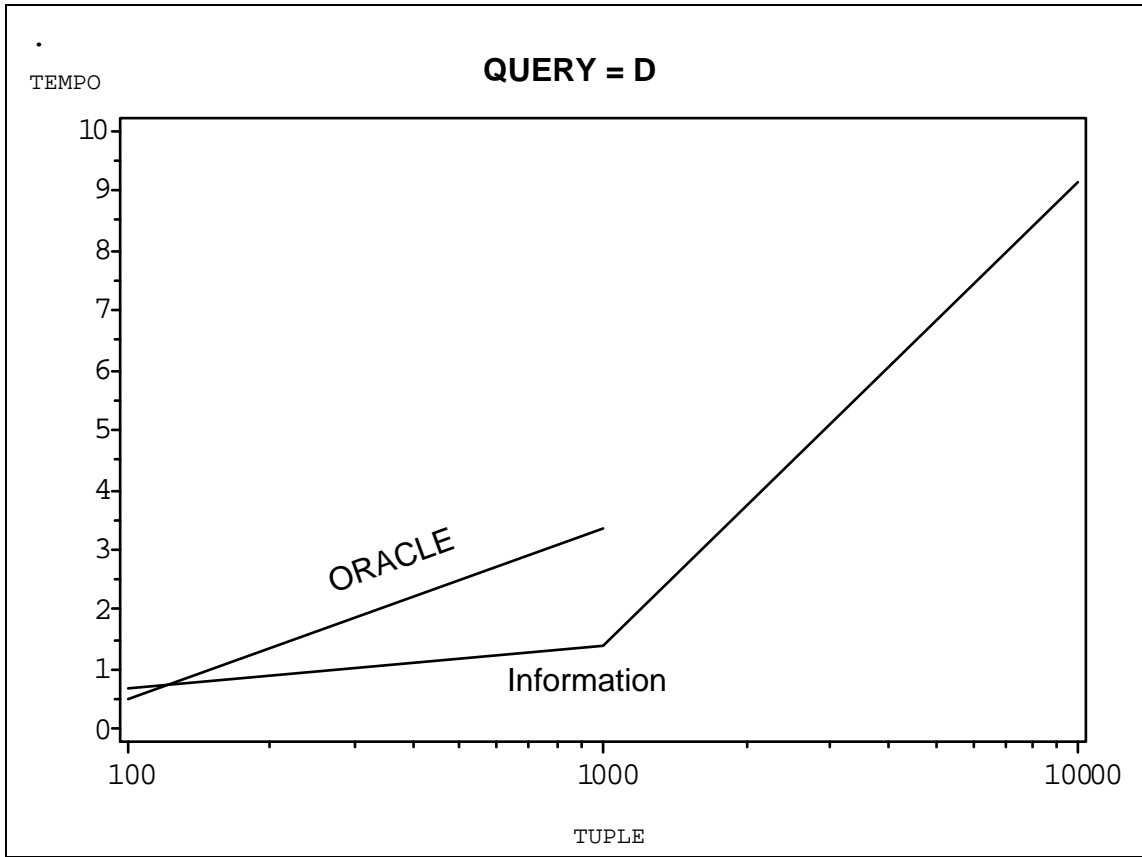
I tempi di CPU sono stati rilevati varie volte, variando di un ordine di grandezza il numero delle ennuple caricate nelle tabelle. Problemi sistemistici (esaurimento dello spazio libero nella partizione) hanno impedito il caricamento di 10.000 ennuple nel database Oracle, per cui i dati sono stati rilevati due volte (100 e 1000 tuple) per Oracle, e tre volte (100, 1.000 e 10.000 tuple) per Information.

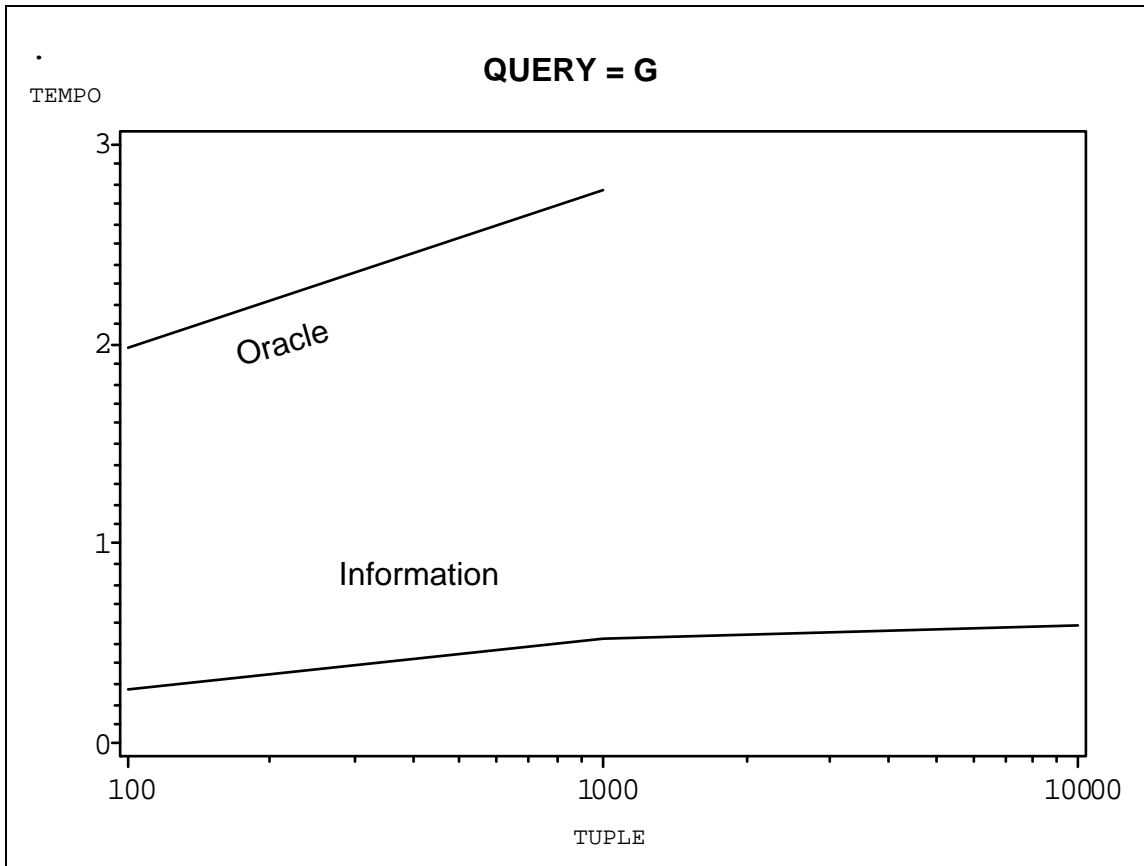
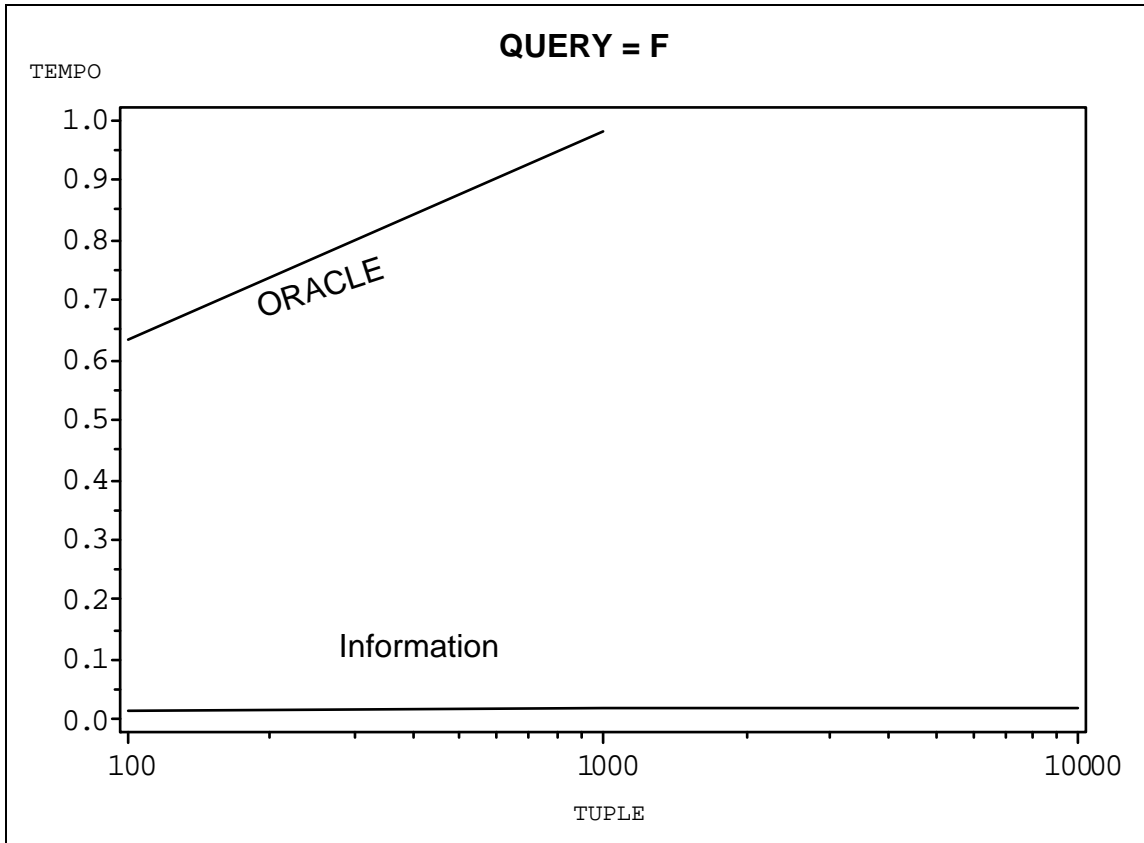
I risultati ottenuti sono riassunti nelle tabelle e nei grafici seguenti, in cui vengono riportati in ascisse il numero di tuple, e in ordinate il tempo di CPU in secondi.

tuple	100			1.000		
	Oracle	Information	Oracle/ Information	Oracle	Information	Oracle/ Information
a	0.833	0.4515	1.8450	1.760	1.3242	1.3291
b	0.842	0.4393	1.9167	2.121	1.2939	1.6392
c	3.003	0.8000	3.7538	40.491	3.4363	11.7832
d	0.518	0.6909	0.7497	3.339	1.4212	2.3494
e	1.709	0.3151	5.4237	2.272	0.3363	6.7559
f	0.636	0.0151	42.1192	0.981	0.0181	54.1990
g	1.984	0.2818	7.0405	2.763	0.5242	5.2709









5 - Conclusioni

Entrambi i sistemi dispongono di strumenti validi per la gestione di basi di dati.

Oracle dispone del linguaggio SQL, più efficiente ed espressivo di quello di Information, e aderente agli standard internazionali.

Information non dispone degli operatori dell' algebra relazionale, ma consente di verificare un gran numero di vincoli direttamente nella fase di digitazione. Inoltre, permette di gestire campi multivalore, il che permette di ridurre il numero di tabelle da creare, in quanto vengono parzialmente annullati gli effetti della normalizzazione.

Oracle dispone di adeguati meccanismi di rollback e recovery, e quindi è in grado di garantire l' integrità dei dati.

Information utilizza file nativi, il che gli permette di accedere direttamente a file creati da altre applicazioni, e, d' altro canto, consente ad applicazioni esterne di accedere ai suoi archivi.

In Information, l' accesso ai record per chiave primaria avviene mediante indici hash, e risulta quindi più veloce di quello effettuato da Oracle, mediante B*-trees. In Oracle è tuttavia possibile definire dei join fisici (cluster) che permettono di rendere più veloci i join su più tabelle, e di mitigare gli effetti prodotti dalla proliferazione di tabelle a fronte di attributi multivalore.

Per quanto concerne gli strumenti per lo sviluppo di applicazioni, SQL*FORMS è sicuramente molto più sofisticato e completo di SIMPLE, ma è estremamente macchinoso, e richiede molta pratica prima che l' utente sia in grado di utilizzarlo efficientemente.

Oracle non è in grado di recuperare lo spazio fisico inutilizzato a seguito di cancellazioni di record, e richiede un intervento di riorganizzazione da parte dell' amministratore della base di dati. Da questo punto di vista, Information richiede meno interventi di manutenzione.

Dall' esame dei tempi di risposta, appare evidente che Information risulta normalmente più veloce di Oracle, e che in alcuni casi tale differenza è notevole.

In definitiva, Oracle è da ritenersi senza alcun dubbio un sistema molto sofisticato, e offre alcuni vantaggi molto rilevanti per quanto concerne la portabilità, il rispetto del modello relazionale e l' integrità fisica e logica dei dati, ma ha prestazioni non eccezionali, richiede la presenza di un amministratore della base di dati, e dispone di una documentazione vasta ma dispersiva.

Information presenta i grossi limiti della mancanza di meccanismi di salvaguardia dell' integrità dei dati e della dipendenza dall' ambiente hardware e software, ma risulta uno strumento piuttosto semplice e di buone prestazioni.

Ringraziamenti

Un caloroso ringraziamento va al Dott. Jean Louis Renaud della Prime Computer, per le delucidazioni fornite, e a Edoardo Bracci del CNUCE, per l' aiuto fornito nella elaborazione statistica dei dati.