

Confronto tra un DBMS relazionale e un File System

Oreste Signore - Vito Cusumano
CNUCE - CNR via S. Maria, 36 56126 PISA

A.I.C.A. & CMG-Italia
*12^o giornata di studio sulla
Valutazione delle prestazioni dei sistemi informatici*

Milano
21 giugno 1991

Contenuto

- **I sistemi considerati:**
 - *Caratteristiche generali*
 - *Il DDL*
 - *Il DML*
 - *Gli indici*
 - *Le tecniche di ottimizzazione*
 - *Integrità e sicurezza dei dati*
 - *La struttura fisica*
- **Il database di test:**
 - *Descrizione dell' applicazione*
 - *Lo schema concettuale dei dati*
 - *Lo schema logico*
- **Le prestazioni**
- **Conclusioni**

Introduzione

- Oracle release 5.1.22.3

- Information release 7.8

- Confronto in base a:

Documentazione:

- fedeltà al modello relazionale
- garanzia dei vincoli di integrità
- rispetto dell' architettura a tre livelli
- architettura generale;
- portabilità

Prestazioni:

- facilità di sviluppo di applicazioni mediante i tool forniti dai due sistemi;
 - tempi di risposta a fronte di alcune query, eseguite sulla base di dati popolata con dati generati in modo casuale.
- Computer Prime 9955
 - 5 MIPS
 - architettura a 32 bit.

Caratteristiche generali

- **Oracle**
 - nucleo base (kernel)
 - SQL*PLUS
 - SQL*FORMS
 - SQL*REPORT
 - PRO*C
 - PRO*COBOL
 - PRO*FORTRAN
 - ODL
 - IMP
 - EXP

- **Information**
 - sistema di IV generazione
 - implementazione della Prime del sistema Pick
 - commercializzato come relazionale
 - basato su file discreti
 - SIMPLE per applicazioni interattive
 - maschere richiamabili da INFO/BASIC
 - interfaccia PERFORM interprete dei comandi
 - interfaccia CONNECTION per utilizzare i tool:
 - Financial
 - Graph
 - Word
 - linguaggio di interrogazione INFORM
 - prevista la disponibilità di SQL.

II DDL

Oracle

- **Modello relazionale**
- **Accesso tramite SQL (standard ANSI)**
- **Table e view**
- **Vincoli di integrità (entity e referential integrity)**
- **Operazioni sulle viste**

Information

- **File dati e file dizionario**
- **Viste** (*difficili da definire*)
- **Lunghezza degli attributi indefinita**
- **Gestione automatica di *Entity integrity* e *referential integrity***
- **Attributi multivalore**
- **meccanismo delle autorizzazioni basato sul sistema operativo.**

II DML

- **Sostanzialmente equivalenti per aggiornamenti, inserimenti e cancellazioni.**

Oracle

- **clausola: SELECT ... FROM ... WHERE ...**
- **fino a sedici livelli di sottoquery**
- **self-join e outer join**
- **unione, intersezione e differenza**
- **estensione dello standard SQL per il supporto di interrogazioni di relazioni contenenti informazioni gerarchiche (esempio tipico la distinta base)**
- **l'ordine di specifica degli attributi nei predicati può influire sulla velocità di esecuzione della query**

Information

- **non è possibile effettuare join**
- **una query non può contenere sottoquery**
- **Un (equi-)join può essere realizzato utilizzando un meccanismo di attivazione di liste**
- **è possibile specificare condizioni sui campi multivalore, e rendere visibili o tutte le istanze, o solo quelle che soddisfano la condizione.**

Indici e ottimizzazione

Oracle

- compressi
- UNIQUE
- su più colonne

Information

- indici su attributi multivalore
- indici su attributi derivati
- disabilitazione dell' indice

Oracle

- **utilizza o meno gli indici a seconda del predicato specificato**
- **realizza il join in modi diversi (tabella guida o sort-merge)**
- **il cammino scelto è influenzato dall' ordine in cui vengono specificate le tabelle nella clausola:
SELECT ... FROM. WHERE**

Information

- **si limita a controllare se esiste un indice su un attributo del predicato, e decidere se utilizzarlo o meno**
- **l' ordine di specifica dei predicati può influire sul tempo di esecuzione**

Organizzazione fisica, integrità, sicurezza

Oracle

- **spazio fisico gestito in modo indipendente**
(un unico spazio logico)
- **possibilità di definire cluster**
- **COMMIT e ROLLBACK**
- **Recovery automatico**
- **Before Image, After Image, Log, checkpoint.**
- **GRANT e REVOKE.**
- **Vincoli non definibili nello schema**
- **in ambiente SQL*FORMS, meccanismo dei trigger**
- **possibilità di bloccare l' accesso a una tabella**

Information

- **unica possibilità l' attivazione di un disco di copia**
(garanzia dell' integrità fisica, ma non di quella logica)
- **accessi concorrenti gestiti utilizzando le primitive messe a disposizione dal sistema operativo**
(aggravio di lavoro per l' utente)
- **file nativi di sistema**

Il database di test: specifiche informali

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.

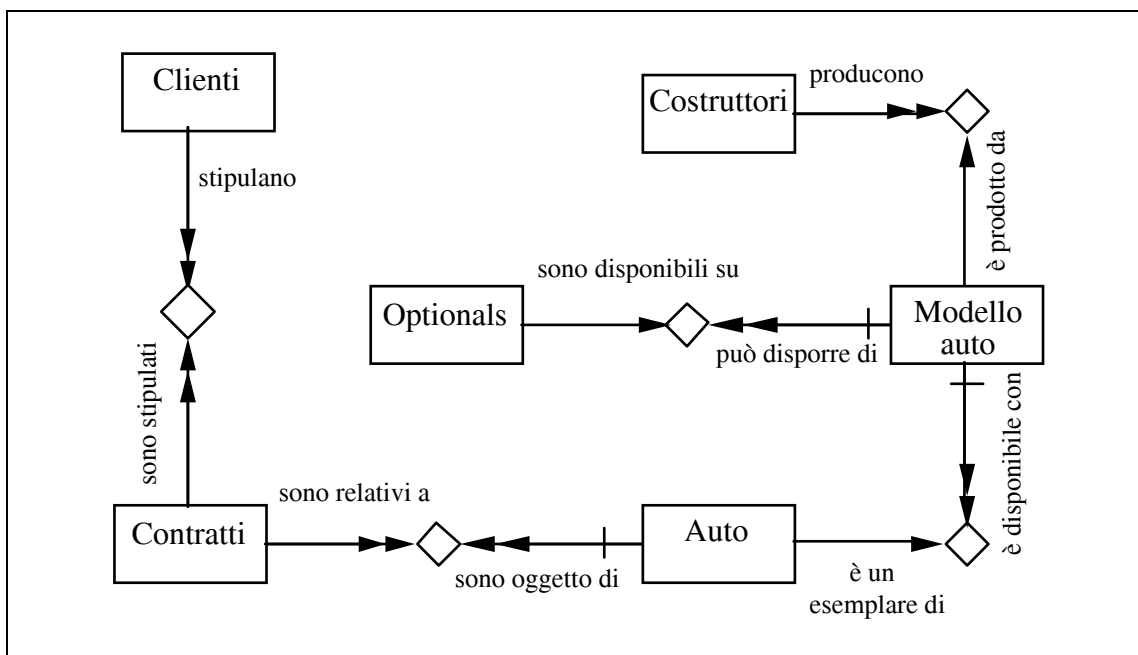
Per ogni auto trattata, vengono gestite informazioni quali il numero di telaio, la data di immatricolazione, la targa, i numeri delle pratiche che l'hanno gestita, l'attuale posizione (venduta o disponibile), gli optional disponibili su quel modello, con relativo prezzo e se sono montati di serie, e,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

Il database di test: lo schema concettuale



Lo schema logico

Oracle

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, Lungh, Largh, Max-Vel, Consumo, Prezzo-Base, *CF-Costruttore*)

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

Cluster:

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

Viste:

- I_CLIENTI
- I_COSTRUTTORI
- NS_AUTO (associa ad ogni auto le pratiche che l' hanno trattata)
- COSTO_COMP
- TOTALE
- STORIA_AUTO

Information

- Un file per ogni entità: **CONTRATTI, CLIENTI, COSTRUTTORI, AUTO, MODELLI, OPTIONAL**
- **Vincoli sui valori possibili (e pattern del Codice Fiscale), definiti a livello di dizionario.**
- **Viste COSTO_COMP e STORIA_AUTO**

Le prestazioni

Sviluppo di applicazioni:

- Medesima procedura sia in ambiente SQL*FORMS che in ambiente SIMPLE.
- Tempi di risposta e aspetto esterno equivalenti.
- Definizione dei vincoli a livello di schema preferibile al meccanismo dei trigger.

Query

- Tempi di risposta a query non banali.
- Dati di prova generati in modo casuale, secondo una distribuzione poissoniana, ma rispettando i vincoli.
- Tempi di CPU rilevati variando di un ordine di grandezza il numero delle ennuple caricate nelle tabelle.
- Oracle: 100 e 1.000 tuple
(errore di sistema su 10.000 tuple)
- Information: 100, 1.000, 10.000 tuple

Le prestazioni (cont.)

- **Le query:**
 - 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.

- **Tabella riassuntiva:**
(tempo di CPU in secondi)

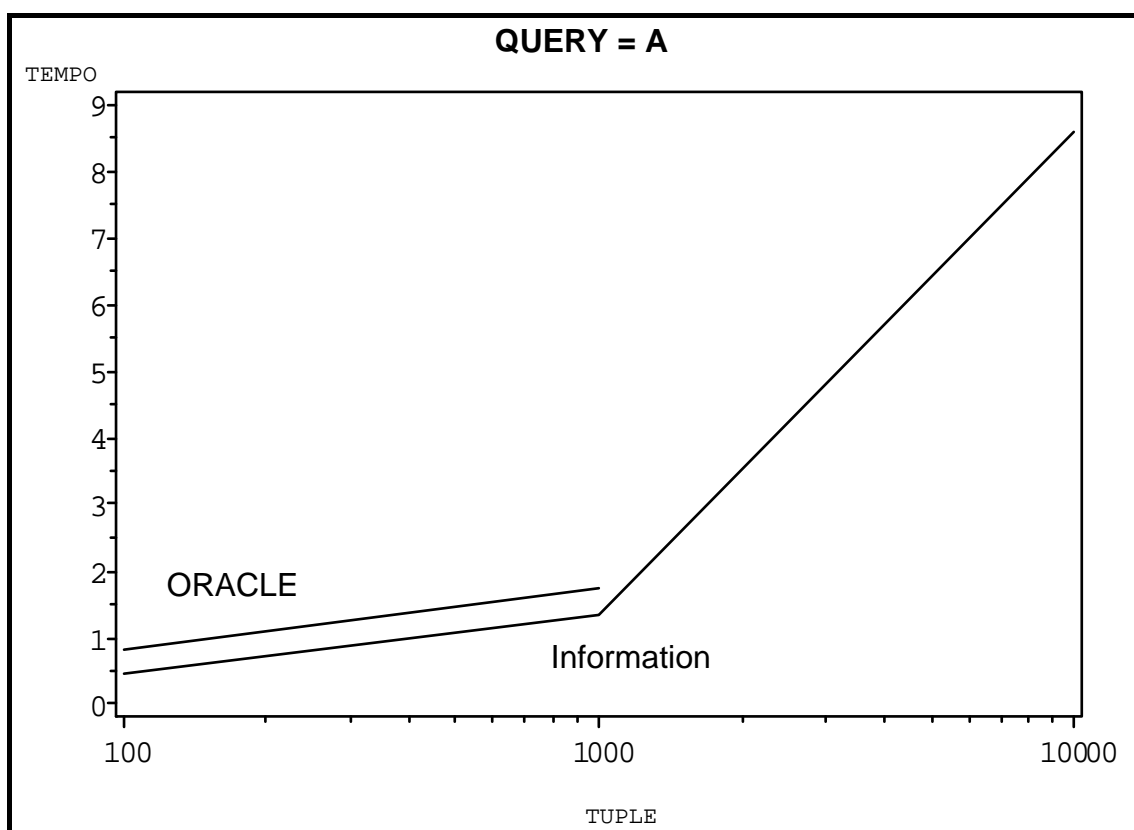
tuple	100			1.000		
	Oracle	Informatio <i>n</i>	Oracle/ Informatio <i>n</i>	Oracle	Informatio <i>n</i>	Oracle/ Informatio <i>n</i>
a	0.833	0.4515	1.8450	1.760	1.3242	1.3291
b	0.842	0.4393	1.9167	2.121	1.2939	1.6392
c	3.003	0.8000	3.7538	40.491	3.4363	11.7832
d	0.518	0.6909	0.7497	3.339	1.4212	2.3494
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

Le query (a)

Ricostruzione della storia di un' auto trattata

```
SELECT *  
FROM   STORIA_AUTO  
WHERE  NUM_TELAIO = 'XXX'
```

```
LIST STORIA_AUTO  
WITH NUM_TELAIO = 'XXX'
```

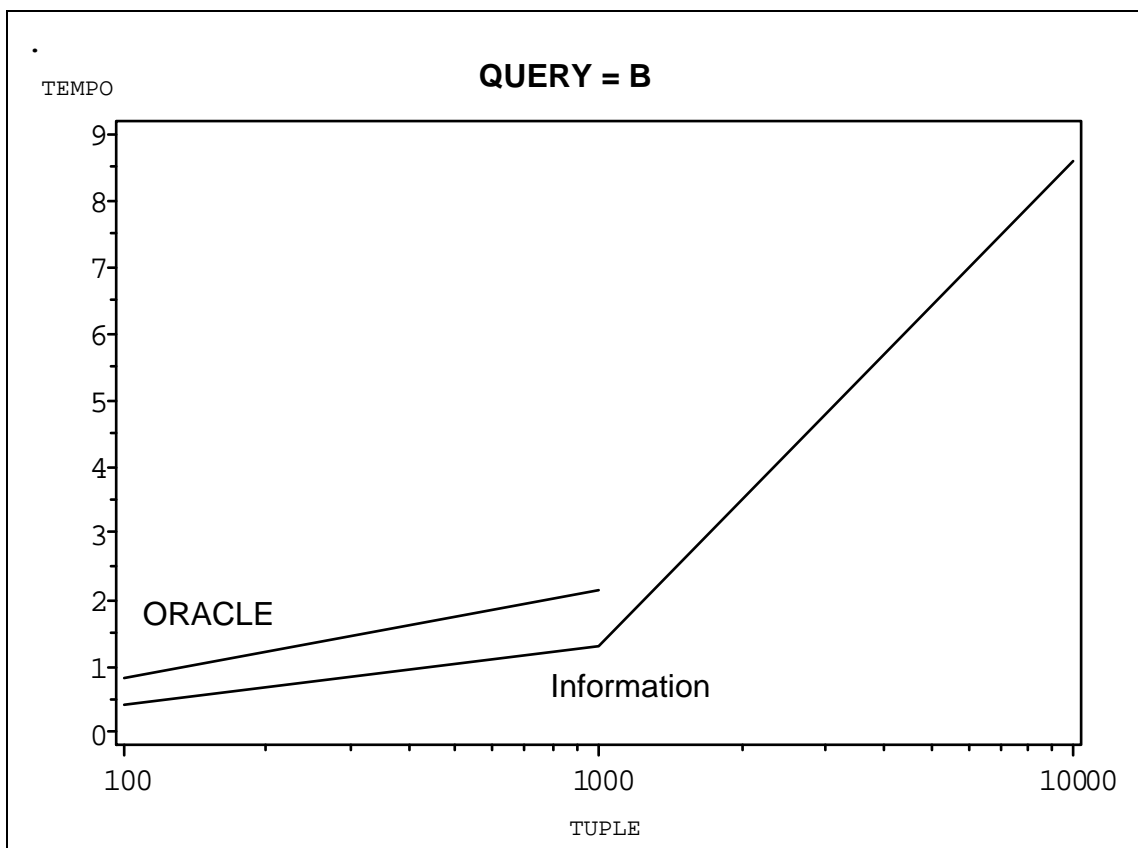


Le query (b)

Ricostruzione della storia di un' auto trattata, con report ordinato per data contratto

```
SELECT *  
FROM STORIA_AUTO  
WHERE NUM_TELAIO = 'XXX'  
ORDER BY DATA_CONTRATTO
```

```
LIST STORIA_AUTO  
WITH NUM_TELAIO = 'XXX'  
ORDER BY.DSND DATA_CONTRATTO
```

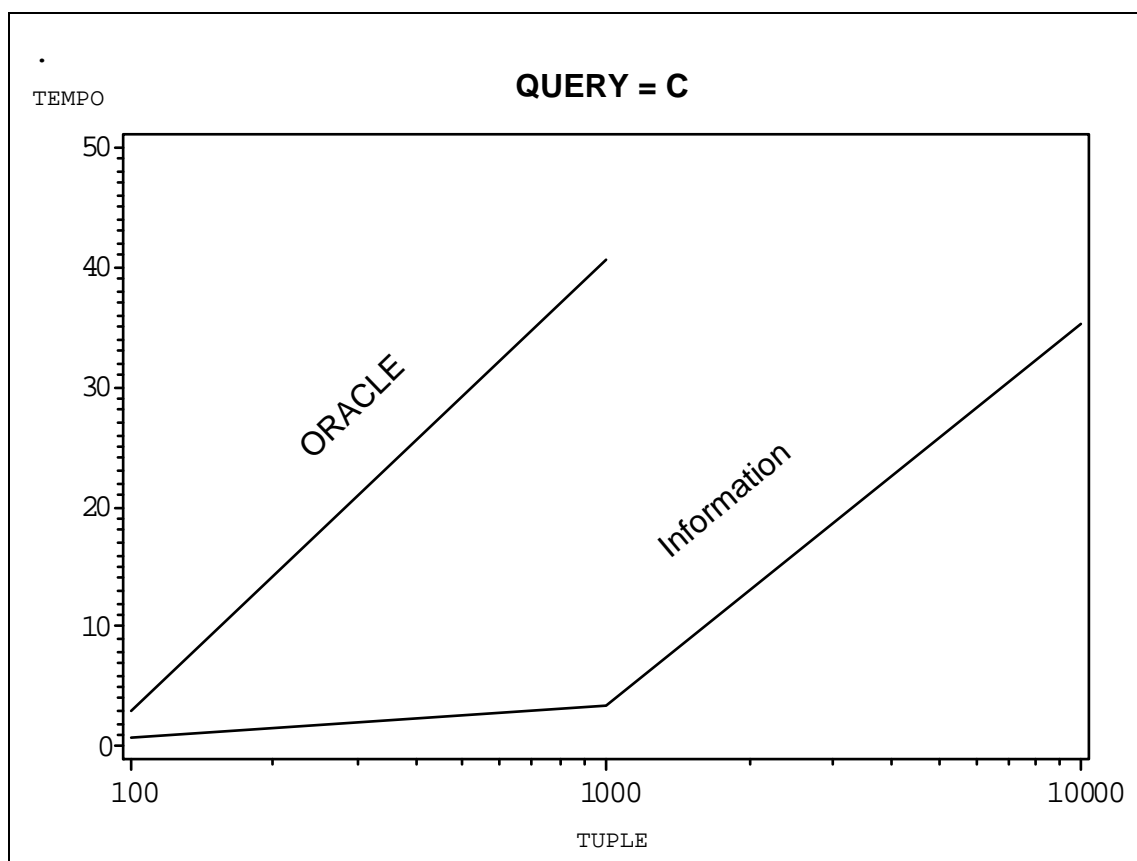


Le query (c)

Generalità dell' attuale proprietario di un' auto

```
SELECT  I_CLIENTI.CF_CLIENTE, NOME, INDIRIZZO, TELEFONO
FROM    I.CLIENTI, CONTRATTI, AUTO
WHERE   TARGA = 'XXX' AND
        AUTO.NUM_TELAIO = CONTRATTI.NUM_TEL_VEND AND
        CONTRATTI.CF_CLIENTE = I_CLIENTI.CF_CLIENTE AND
        DATA_CONTRATTO >= ALL (SELECT DATA_CONTRATTO
                                FROM CONTRATTI
                                WHERE     NUM_TELAIO =
NUM_TEL_VEND)
```

```
SELECT  AUTO WITH TARGA = 'XXX' SAVING NUM_PRATICA
SSELECT CONTRATTI BY.DSND DATA_CONTRATTO
SSELECT CONTRATTI FIRST 1 SAVING CF_CLIENTE
LIST    CLIENTI
```

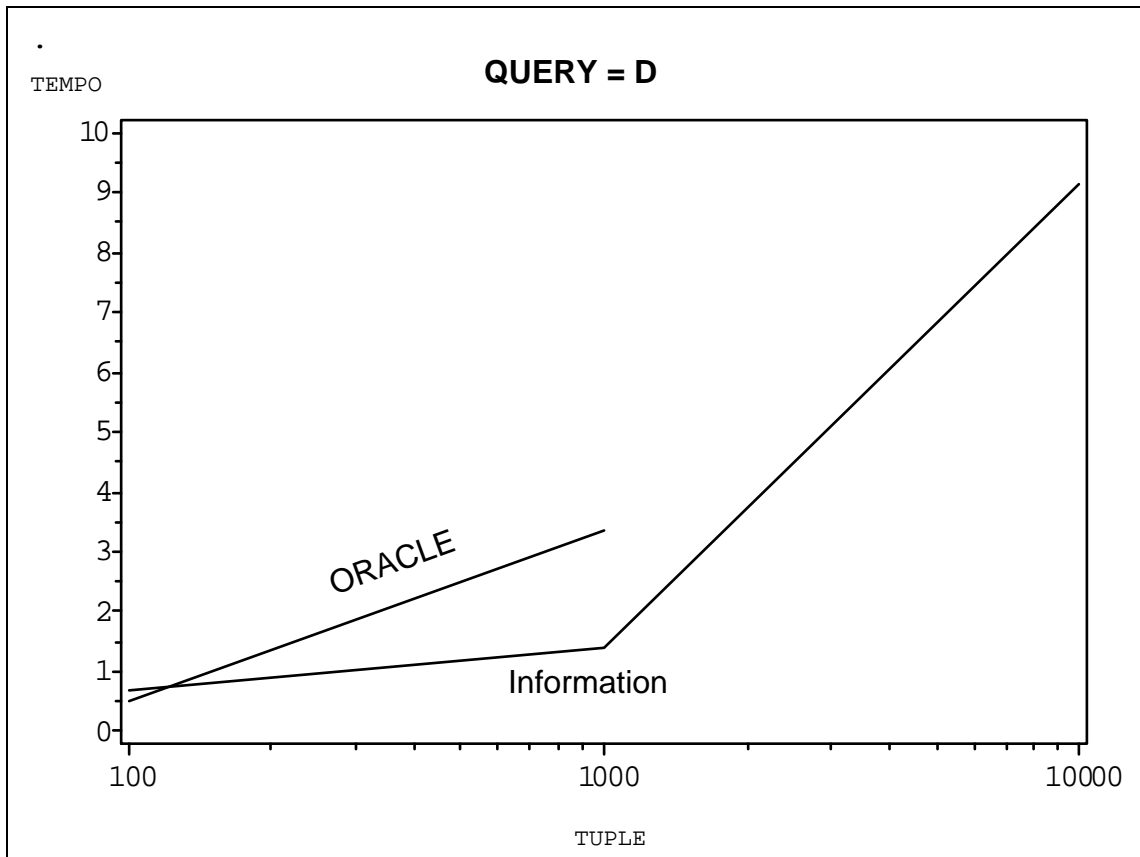


Le query (d)

Generalità del costruttore dell' auto acquistata da un certo cliente

```
SELECT  AUTO.MODELLO, CF_COSTRUTTORE, NOME, INDIRIZZO, TELEFONO
FROM    I.COSTRUTTORI, CONTRATTI, AUTO, MODELLI
WHERE   CF_CLIENTE = 'XXX'  AND
        NUM_TEL_VEND = NUM_TELAIO  AND
        MODELLO = NOME_MODELLO  AND
        MODELLI.CF_COSTRUTTORE   =   I.COSTRUTTORI.CF_COSTRUTTORE
AND
        DATA_CONTRATTO >= ALL (SELECT DATA_CONTRATTO
                                FROM CONTRATTI)
```

```
SELECT  CONTRATTI WITH CF_CLIENTE = 'XXX'
        BY.DSND DATA_CONTRATTO
SELECT  CONTRATTI FIRST 1 SAVING NUM_TEL_VEND
SELECT  AUTO SAVING MODELLO
SELECT  MODELLI SAVING CF_COSTRUTTORE
LIST    COSTRUTTORI
```

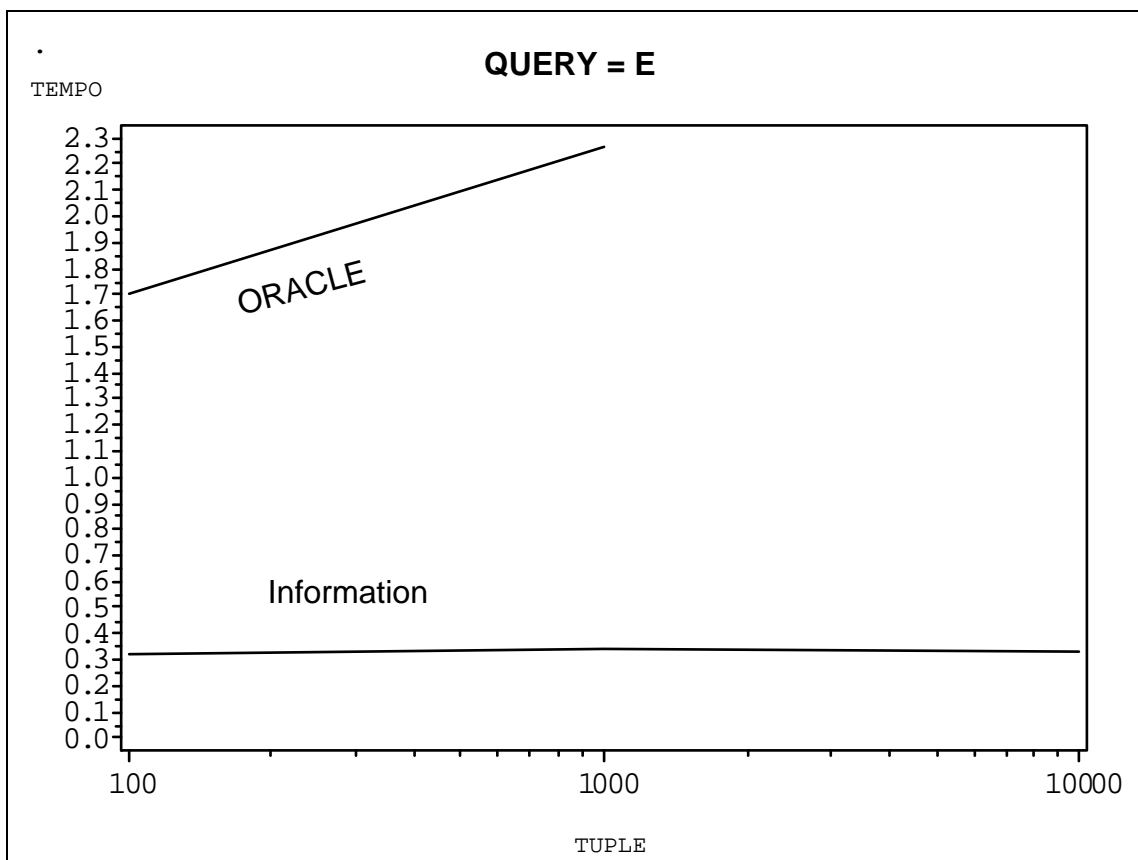


Le query (e)

Prezzo complessivo di un' auto, su cui sono montati un certo numero di optional

```
DELETE FROM DUMMY;
INSERT INTO DUMMY VALUES ('AAA');
INSERT INTO DUMMY VALUES ('BBB');
SELECT OPTIONAL , PREZZO, IMPORTO+PREZZO_BASE, IMPORTO
FROM TOTALE, MODELLI
WHERE NOME_MODELLO = 'XXX' AND
MODELLO = 'XXX';
DELETE FROM DUMMY;
```

```
SELECT DICT COSTO_COMP WITH ID = OPT
MODIFY DICT COSTO_COMP
LIST COSTO_COMP WITH MODELLO = 'XXX'
```

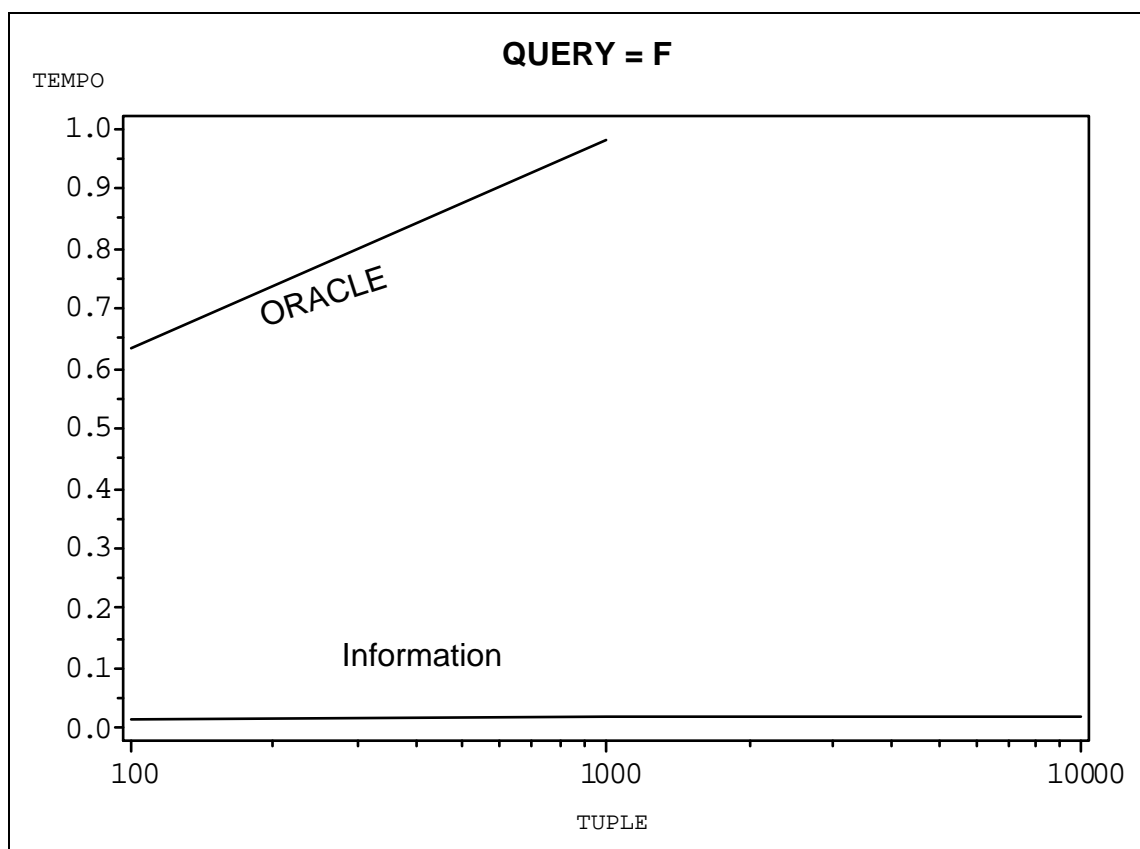


Le query (f)

Inserimento di un nuovo cliente

```
INSERT INTO CLIENTI VALUES  
(RSSFRN46P13L331G','ROSSI FRANCESCO','VIA ROMA N. 13');
```

```
MODIFY CLIENTI
```

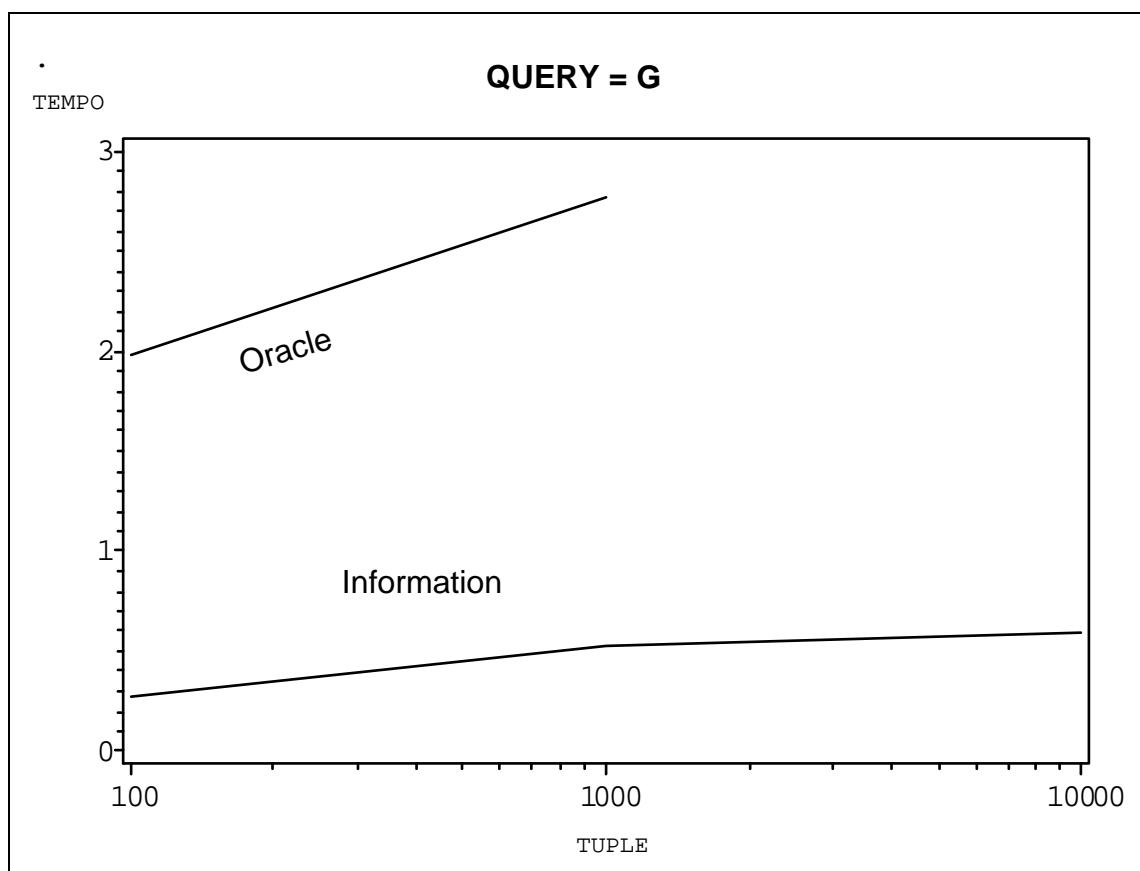


Le query (g)

Creazione di un indice sul campo *Nome* della classe CLIENTI.

```
CREATE INDEX INDICE_CLIENTI  
ON CLIENTI(NOME);
```

```
CREATE INDEX INDICE_CLIENTI CLIENTI NOME
```



Conclusioni

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

Da un punto di vista teorico ...

- **Oracle:**
 - **linguaggio SQL (espressività, standard)**
- **Information:**
 - **mancano gli operatori dell' algebra relazionale**
 - **verifica automatica di un gran numero di vincoli**
 - **gestione dei campi multivalore**
(minor numero di tabelle, bilancia gli effetti della normalizzazione)

- **Oracle:**
 - **rollback e recovery**
- **Information:**
 - **file nativi**
accesso a file esterni, accessibilità da programmi esterni

- **Information**
 - **indici hash sulla chiave primaria**
- **Oracle:**
 - **indici B*-trees**
 - **definizione di join fisici (cluster)**
maggiore efficienza dei join su più tabelle,
mitigazione degli effetti della normalizzazione su attributi multivalore

Conclusioni (cont.)

Da un punto di vista operativo ...

- **SQL*FORMS molto più sofisticato e completo di SIMPLE
ma ...**
macchinoso, richiede molta pratica per un uso efficiente
- **Interventi di manutenzione più frequenti e complessi in Oracle.**
- **Tempi di risposta migliori con Information**
(in alcuni casi differenze notevoli)

Conclusioni (cont.)

In definitiva ...

- **Oracle**
 - sistema molto sofisticato
 - portabilità
 - rispetto del modello relazionale
 - integrità fisica e logica dei dati
 - prestazioni non eccezionali
 - richiede la presenza di un DBA
 - documentazione vasta ma dispersiva.

- **Information:**
 - integrità dei dati non assicurata
 - dipendenza dall' ambiente hardware e software
 - semplice e di buone prestazioni.

Quanto vale il rispetto del modello relazionale?

E' conveniente l' approccio relazionale anche in applicazioni non critiche?

