

Algebra e calcolo relazionale

Ripasso

Tipologie dei linguaggi:

Dichiarativi - specificano le proprietà del risultato ("che cosa")

Procedurale - specificano le modalità di generazione del risultato ("come")

DML: data manipulation language

DDL: data definition language

Operazioni di algebra relazionale

Unione

Intersezione

Differenza

Selezione

Ridenominazione

Join

Naturale

Thera-Join

Esterno

Sinistro

Destro

Completo

Algebra (continua)

Equivalenza di espressioni

Push selections (se A è attributo di R₂)

$$SEL A=10 (R1 JOIN R2) = R1 JOIN SEL A=10 (R2)$$

Selezione con valori nulli

Viste (relazioni derivate)

Conclusione Algebra relazionale

Operatori dell'algebra relazionale

unione, intersezione, differenza

selezione

proiezione

ridenominazione

join (join naturale, prodotto cartesiano, theta-join)

Calcolo relazionale

Il Calcolo relazionale fa parte di una famiglia di linguaggi dichiarativi, basati sul calcolo dei predicati del primo ordine.

Calcolo su Domini

Questo tipo di calcolo è composto da espressioni applicate a domini.

Le espressioni hanno la forma:

$$\{ A1: x1, \dots, Ak: xk \mid f \}$$

dove:

f è una formula (che può utilizzare connettivi booleani e quantificatori)

$A1: x1, \dots, Ak: xk$ è una lista di coppie attributo – variabile, detta “target list”, con $A1, \dots, Ak$ attributi distinti (che possono anche non far parte della base di dati)

$x1, \dots, xk$ variabili tra loro distinte

Il significato o risultato di un'espressione e è una relazione sui domini $A1, \dots, Ak$ che contiene le n -uple di valori per $x1, \dots, xk$ tali per cui la formula f risulta verificata

Nelle formule possono essere utilizzati i qualificatori universali ed esistenziali:

- per ogni \forall
- esiste \exists

Il calcolo sui domini ha il difetto di essere molto lungo (per scrivere una semplice interrogazione ci vogliono predicati molto lunghi, e quindi è di difficile lettura) e soprattutto permette di scrivere espressioni senza senso.

Calcolo su ennuple con dichiarazioni di range

Per superare i limiti del calcolo sui domini viene definito il “calcolo su ennuple con dichiarazioni di range”, dove per semplificare e snellire la scrittura ci si può riferire ad un'intera n -upla utilizzando una singola variabile (invece della lista delle coppie attributo-variabile), ed i valori ed i domini vengono ristretti ai soli provenienti da una definita base di dati.

Le espressioni del “calcolo su ennuple con dichiarazioni di range” hanno la seguente forma:

$$\{ TargetList \mid RangeList \mid Formula \}$$

dove

- ▲ *RangeList* è la lista delle relazioni associate alle relative variabili d'uso
- ▲ *TargetList* è la lista degli attributi che verranno estratti dalla *RangeList*
- ▲ *Formula* è appunto una formula che deve essere soddisfatta dalle n -uple delle relazioni della *RangeList* affinché i loro valori entrino a far parte del risultato nella *TargetList*

Calcolo e algebra

Calcolo e algebra sono "equivalenti"

per ogni espressione del calcolo relazionale che sia indipendente dal dominio esiste un'espressione dell'algebra relazionale equivalente a essa

per ogni espressione dell'algebra relazionale esiste un'espressione del calcolo relazionale equivalente a essa (e di conseguenza indipendente dal dominio)

Limiti dell'algebra e del calcolo relazionale

Chiusura transitiva

Supervisione(Impiegato, Capo)

Per ogni impiegato, trovare tutti i superiori (cioè il capo, il capo del capo, e così via)

Nell'esempio, basterebbe il join della relazione con se stessa, facendo però ad ogni passo un'opportuna ridenominazione, ma servirebbero potenzialmente infinite join e ridenominazioni

Chiusura transitiva, impossibile!

Non esiste in algebra e calcolo relazionale la possibilità di esprimere l'interrogazione che, per ogni relazione binaria, ne calcoli la chiusura transitiva

Mercato dei DBMS

Players:

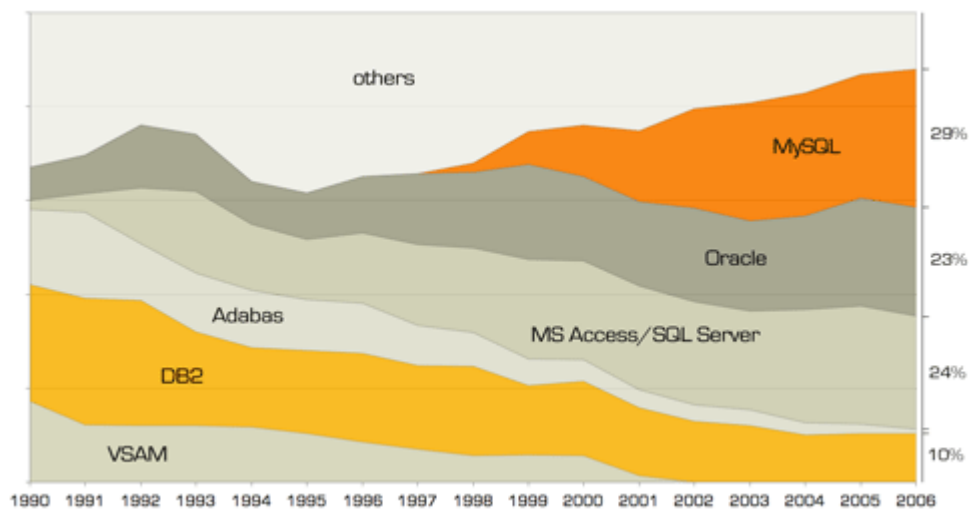
<ul style="list-style-type: none"> ^ 4th Dimension ^ Adabas D ^ Alpha Five ^ Apache Derby ^ Aster Data ^ BlackRay ^ CA-Datacom ^ CSQL ^ CUBRID ^ Daffodil database ^ DataEase ^ Database Management Library ^ Dataphor ^ DB-Fast ^ DB2 ^ Derby aka Java DB ^ ElevateDB ^ Empress Embedded Database ^ EnterpriseDB ^ EffiProz ^ eXtremeDB ^ fastDB ^ FileMaker Pro ^ Firebird ^ FrontBase ^ Gladius DB ^ Greenplum ^ H2 ^ Helix database ^ HSQLDB ^ SQLDB ^ IBM DB2 ^ IBM Lotus Approach ^ WCE SL Plus ^ IBM DB2 Express-C ^ Infobright ^ Informix ^ Ingres ^ InterBase ^ InterSystems Caché ^ Kognitio ^ Linter ^ MaxDB ^ Mckoi SQL Database ^ Microsoft Access 	<ul style="list-style-type: none"> ^ Microsoft Jet Database Engine (part of Microsoft Access) ^ Microsoft SQL Server ^ Microsoft SQL Server Express ^ Microsoft Visual FoxPro ^ Mimer SQL ^ MonetDB ^ mSQL ^ MySQL ^ Netezza ^ NonStop SQL ^ Openbase ^ OpenLink Virtuoso (Open Source Edition) ^ OpenLink Virtuoso Universal Server ^ Oracle ^ Oracle Rdb for OpenVMS ^ Panorama ^ Pervasive ^ PostgreSQL ^ Progress Software ^ RDM Embedded ^ RDM Server ^ The SAS system ^ Sav Zigzag ^ ScimoreDB ^ SmallSQL ^ solidDB ^ SQLBase ^ SQLite ^ Sybase Adaptive Server Enterprise ^ Sybase Adaptive Server IQ ^ Sybase SQL Anywhere ^ Sybase Advantage Database Server ^ tdbengine ^ Teradata ^ TimesTen ^ txtSQL ^ UniData ^ UniVerse ^ Valentina (Database) ^ Vertica ^ VistaDB ^ VMDS ^ XSPRADA
--	--

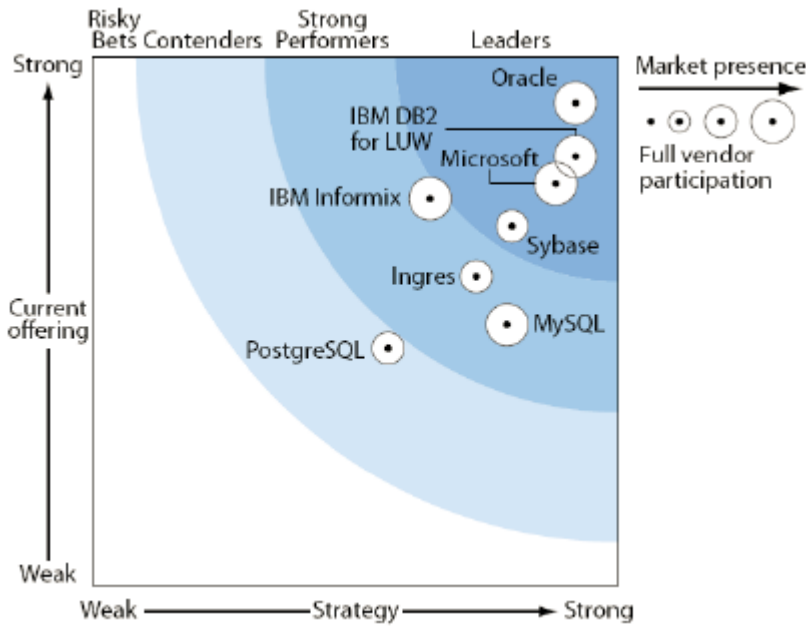
Market Share

Worldwide Embedded DBMS Software Revenue by Vendor, 2007-2009 (\$M)						
	2007	2008	2009	2008 Share (%)	2008-2009 Growth (%)	2009 Share (%)
Oracle	468.4	621.3	643.9	29.6	3.6	30.3
IBM*	238.0	273.0	228.0	13.0	-16.5	10.7
Microsoft	192.3	205.2	215.2	9.8	4.9	10.1
InterSystems Corp.	195.6	212.3	213.1	10.1	0.4	10.0
Progress Software Corp.	203.3	200.3	196.4	9.5	-1.9	9.2
Sybase	165.8	180.8	187.1	8.6	3.4	8.8
Rocket Software*	-	-	50.0	-	NA	2.4
Empress Software	39.1	42.7	39.7	2.0	-7.2	1.9
Pervasive Software Inc.	24.4	25.4	27.9	1.2	9.8	1.3
Sun Microsystems	21.3	24.5	24.6	1.2	0.6	1.2
Apple	12.2	14.6	12.9	0.7	-12.0	0.6
Ingres Corp.	11.2	12.2	11.5	0.6	-5.9	0.5
Versant	11.6	11.9	10.6	0.6	-11.1	0.5
EnterpriseDB	3.7	4.0	3.7	0.2	-5.8	0.2
Objectivity Inc.	3.3	3.3	3.0	0.2	-11.6	0.1

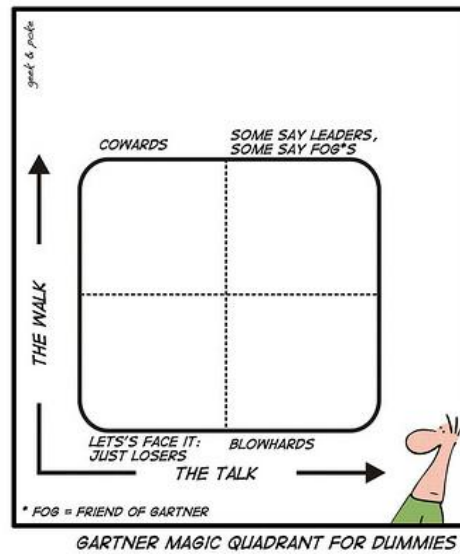
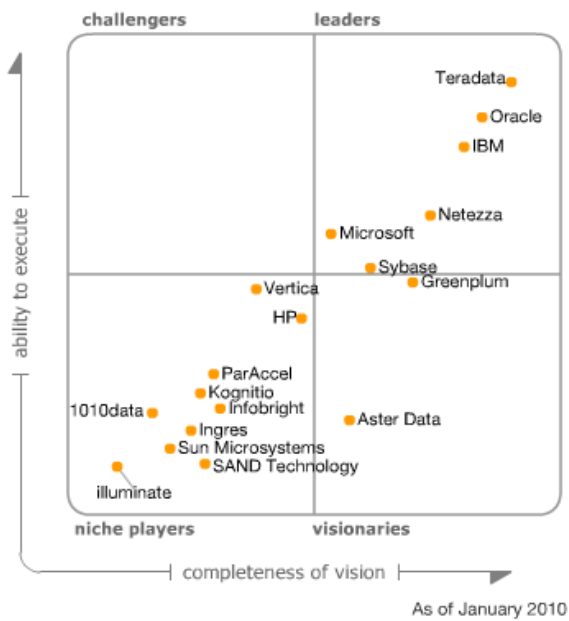
* IBM sold the U2 DBMS technology to Rocket Software in 2009, and so the year's U2 income is attributed to Rocket rather than IBM, causing an artificial dip in IBM revenue and a corresponding rise in Rocket Software revenue.

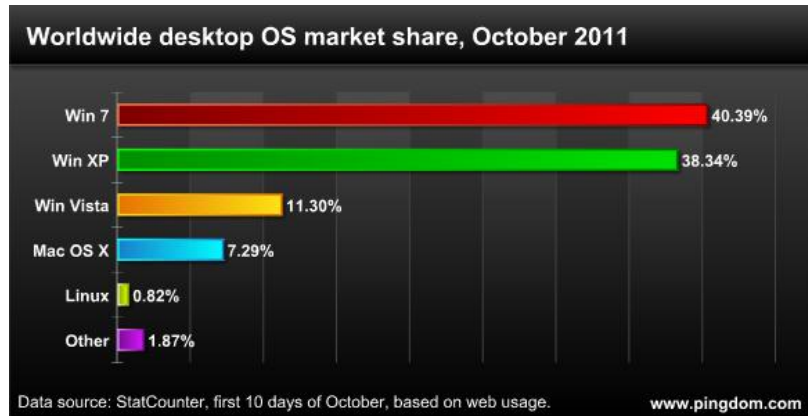
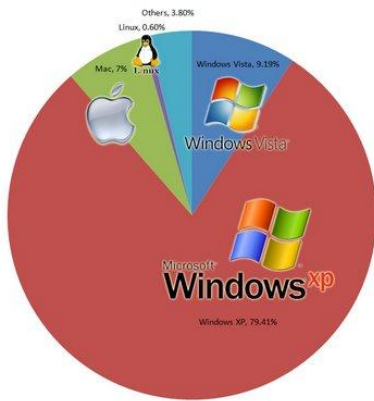
Source: IDC, 2010



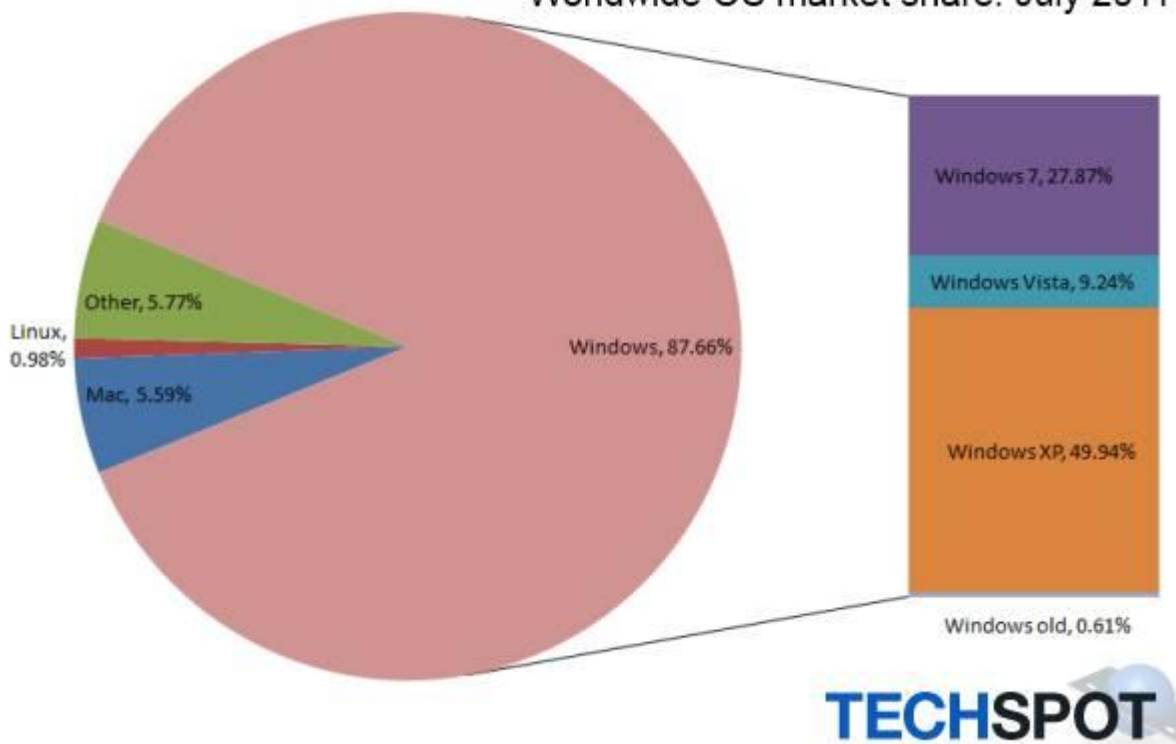


Gartner Magic Quadrant for Data Warehouse Database Management Systems (2010)





Worldwide OS market share: July 2011



Linguaggio e Standard

Structured Query Language

SQL copre sia DDL che DML

Nascita 1970 - Laboratorio IBM – S.Josè – California per DBMS System-R

Prima Standardizzazione 1980 – ANSI – ISO

scarso supporto a definizione schemi ed istanze

successiva introduzione della definizione dei vincoli d'integrità referenziale

1992 – SQL-92 riferito anche come SQL-2

Adozione completa del modello relazionale

definizione di 3 livelli d'implementazione

- entry
- intermediate
- full

1999 – SQL-1999

Modelli ad oggetti

Trigger ed external procedure

2003 – SQL-2003

Eliminazione costrutti obsoleti

Nuove estensioni tra loro indipendenti JRT, XML, etc

Definiti come SQL-3

Nessun sistema commerciale implementa SQL-3, molti arrivano a SQL-2 intermediate

Definizione dei Dati

Domini Elementari

Caratteri

character / char

varchar (character varying)

character set

Tipi numerici esatti

numeric e decimal

precision + scale

integer e smallint

Tipi numerici approssimati

float (precisione)

real

double precision

Date e tempo

date

time e timestamp (precisione) (with time zone)

Intervalli temporali

interval (unita' di tempo di massima) to (Unita' di tempo di dettaglio)

SQL-3

boolean

bigint

BLOB e CLOB

Domini definiti dall'utente

Domini definiti dall'utente

Concetto di dominio definito dall'utente

Vantaggio per la consistenza e la manutenzione dei domini utente

Definizione generale:

```
CREATE DOMAIN nomeDomain as TipoDiDato [valoreDiDefault] [vincolo]
```

PostgreSQL

```
CREATE DOMAIN name [AS] data_type
  [ DEFAULT expression ]
  [ constraint [ ... ] ]
```

where *constraint* is:

```
[ CONSTRAINT constraint_name ]
{ NOT NULL | NULL | CHECK (expression) }
```

Esempio:

```
CREATE DOMAIN us_postal_code AS TEXT
CHECK (
  VALUE ~ '^\\d{5}$'
OR VALUE ~ '^\\d{5}-\\d{4}$'
);
```

Oracle:

```
CREATE TYPE customer_typ_demo AS OBJECT
  ( customer_id      NUMBER(6)
  , cust_first_name  VARCHAR2(20)
  , cust_last_name   VARCHAR2(20)
  , cust_address     CUST_ADDRESS_TYP
  , phone_numbers    PHONE_LIST_TYP
  , cust_email       VARCHAR2(30)
  , cust_orders      ORDER_LIST_TYP
  ) ;
```

Schema - Database

Concetto di schema e di database

Definizione

```
CREATE SCHEMA [nomeSchema]
CREATE DATABASE [nomeDatabase]
```

Table

Creazione di una tabella

```
CREATE TABLE STUDENTI (
    ID int,
    NOME varchar(40),
    MATRICOLA char(10),
    DATA_NASCITA date,
    SESSO char(1) NULL,
    IN_CORSO char(1) DEFAULT 'S',
    PRIMARY KEY (ID) )
```

Vincoli

Vincoli Intrarelazionali

- null e not null

Vincoli Relazionali

- unique
- primary key

Vincoli Interrelazionali

- Foreign key
 - on update
 - on delete
 - cascade
 - set null
 - set default
 - no action

Creazione di una vincoli interrelazionali

```
CREATE TABLE CORSI (  
    ID_CORSO int,  
    CODICE char(5),  
    TITOLO varchar(40),  
    DESCRIZIONE text,  
    DATA_INIZIO date,  
    CREDITI int DEFAULT 2,  
    ANNO int ,  
    REF_PROF int,  
    PRIMARY KEY (ID_CORSO),  
    UNIQUE INDEX IDX_CORSI_CODICE (CODICE)  
);
```

```
CREATE TABLE ESAMI_STUDENTE (  
    ID_ESAME_STUDENTE SERIAL,  
    RIF_STUDENTE int  
        REFERENCES STUDENTI(ID),  
    ID_CORSO int,  
    DATA_ESAME date,  
    VOTO int,  
    PRIMARY KEY (ID_ESAME_STUDENTE),  
    UNIQUE INDEX IDX_ESAMI_STUDENTE  
        (RIF_STUDENTE, ID_CORSO, DATA_ESAME),  
    CONSTRAINT FOREIGN KEY FK_CORSI (ID_CORSO)  
        REFERENCES CORSI(ID_CORSO)  
);
```

Definizione alternativa del vincolo tra ESAMI_STUDENTE e STUDENTI

```
ALTER TABLE ESAMI_STUDENTE ADD CONSTRAINT FOREIGN KEY FK_STUDENTI  
(RIF_STUDENTE) REFERENCES STUDENTI(ID);
```

Modifica degli schemi

Alter di una tabella o di un dominio

```
ALTER TABLE nomeTabella
    ALTER COLUMN nomeAttributo
    ADD COLUMN nomeAttributo
    DROP COLUMN nomeAttributo
    ADD CONSTRAINT nomeConstraint
    DROP CONSTRAINT nomeConstraint
```

Estensioni in SQL-3

Drop di una tabella

```
DROP TABLE nomeTabella (nomeDomain / nomeSchema / ..... )
    RESTRICT
    CASCADE
```

Cataloghi relazionali

Concetto di metadati

Catalogo interno DEFINITION SCHEMA

Catalogo esterno INFORMATION SCHEMA