

Dichiarazione degli schemi in SQL

Storia di SQL

- **Definito nell'ambito del progetto SYSTEM R (IBM S. JOSE) nel 1976**
- **Nome originario: SEQUEL**
- **Adottato progressivamente da tutti i sistemi commerciali**
- **Standardizzato da ANSI e ISO**

Composizione di SQL

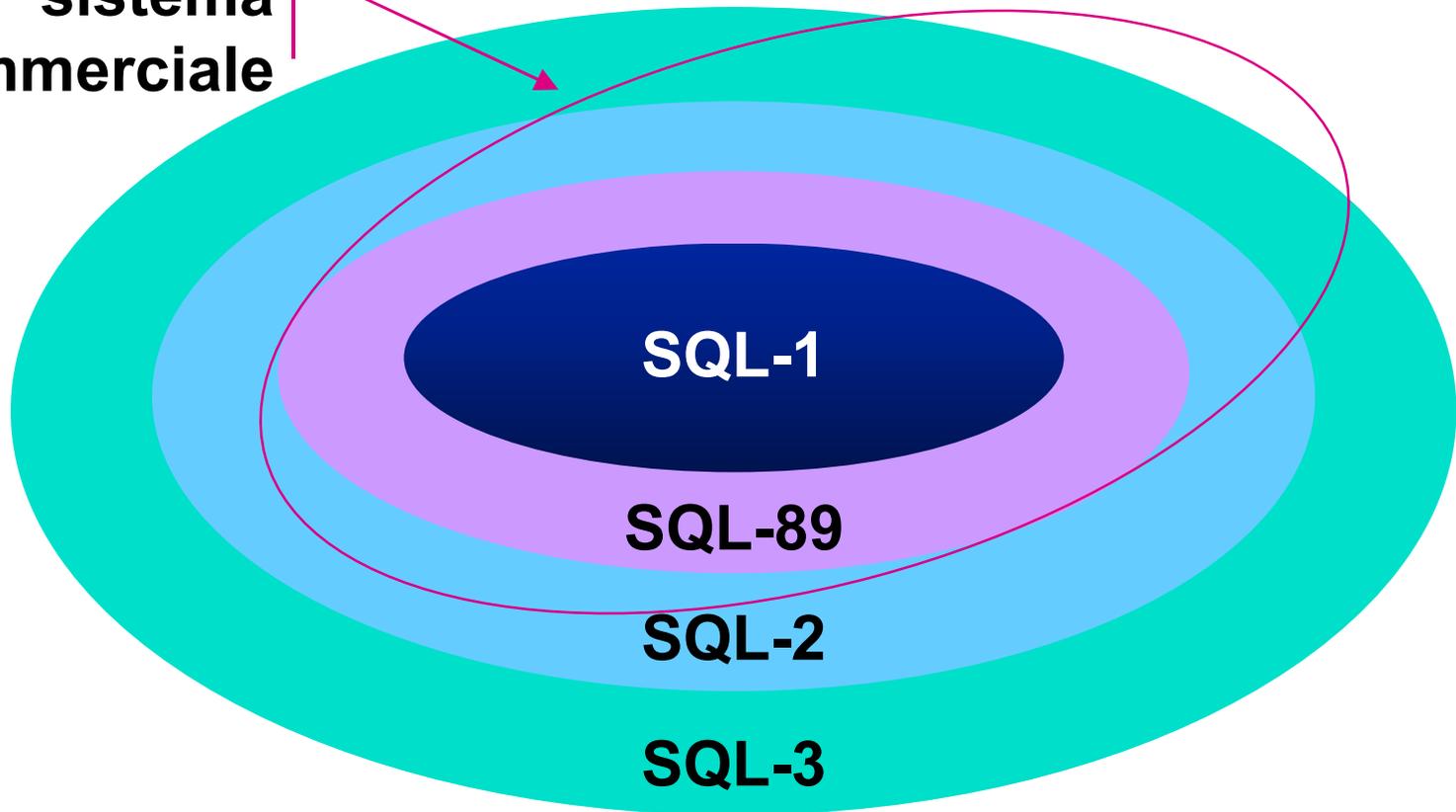
- **Data Definition Language (DDL)**
definizione di: domini, tabelle, indici
autorizzazioni, viste, vincoli, procedure,
trigger
- **Data Manipulation Language (DML)**
linguaggio di query, linguaggio di
modifica, comandi transazionali

Standardizzazione di SQL

- **Progressione dello standard per estensioni quasi-compatibili**
 - prima versione nel 1986 (**SQL-1**)
 - modifica alla prima versione nel 1989 (**SQL-89**)
 - seconda versione nel 1992 (**SQL-2, SQL-92**)
 - terza versione attesa per il 1998 (**SQL-3**)
- **In SQL-2:**
 - **entry SQL** (più o meno equivalente a SQL-89)
 - **intermediate SQL**
 - **full SQL**

Potere espressivo di standard e sistemi commerciali

un tipico
sistema
commerciale



Domini elementari in SQL-2

a **stringhe**

CHAR (N)

VARCHAR (N)

BIT (N)

VARBIT (N)

b **numerici esatti**

NUMERIC (N)

DECIMAL (N)

INTEGER

SMALLINT

Altri domini elementari in SQL-2

c numerici approssimati

FLOAT(N)
N=2, 0.172E04 = 1700

REAL

**DOUBLE
PRECISION**

b domini speciali

DATE
YYYYMMDD
(es. 1998-03-12)
TIME(N)
HHMMSS.NNNN
(es. 14:05:23.123)
TIMESTAMP
(DATE+TIME)
INTERVAL
(pura durata)

Il valore "null"

null è un valore polimorfo
(che appartiene a tutti i domini)
col significato di valore non noto

- il valore esiste in realtà
ma è ignoto al database
(es.: data di nascita)
- il valore è inapplicabile
(es.: numero patente per minorenni)

Vincoli di integrità di dominio

NOT NULL

esclude il valore nullo

PRIMARY KEY

chiave primaria
(implica NOT NULL)

UNIQUE

chiave secondaria
(non implica NOT NULL)

CHECK

predicato che deve essere
soddisfatto

Definizione dei domini applicativi

```
CREATE DOMAIN PrezzoQuotidiani  
AS INTEGER  
    DEFAULT 1500  
    NOT NULL
```

```
CREATE DOMAIN OreLezione  
AS SMALLINT  
    DEFAULT 80
```

Definizione delle tabelle

CREATE TABLE STUDENTE

**(MATR CHAR(6) PRIMARY KEY,
NOME VARCHAR (30) NOT NULL,
NOME VARCHAR (30) NOT NULL,
CITTÀ VARCHAR (20),
C-DIP CHAR (3))**

CREATE TABLE CORSO

**(COD-CORSO CHAR(6) PRIMARY KEY,
TITOLO VARCHAR(30) NOT NULL,
DOCENTE VARCHAR(20))**

Definizione delle tabelle (2)

```
CREATE TABLE ESAME  
( MATR CHAR(6),  
  COD-CORSO CHAR(6),  
  DATA DATE NOT NULL,  
  VOTO SMALLINT NOT NULL,  
  PRIMARY KEY(MATR,COD-CORSO) )
```

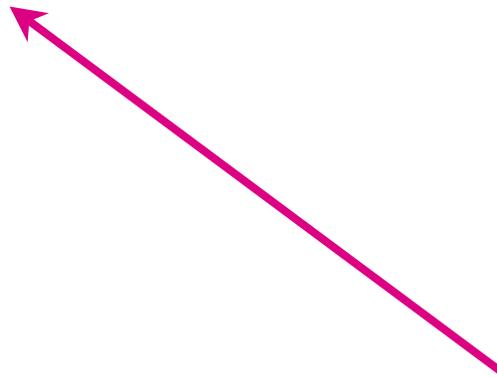
Integrità referenziale

- Esprime un legame gerarchico (padre-figlio) fra tabelle
- Alcuni attributi della tabella figlio sono definiti **FOREIGN KEY**
- I valori contenuti nella **FOREIGN KEY** devono essere sempre presenti nella tabella padre

Esempio: studente - esame

studente

MATR	
123	
415	
702	



esame

MATR	
123	
123	
702	

Il problema degli orfani

studente

MATR	
123	
415	
702	

orfani:

tuple che restano prive di padre a causa di cancellazioni e modifiche della tabella padre

esame

MATR	
123	
123	
702	

Gestione degli orfani

Cosa succede degli esami se si **cancellano** gli studenti?

- a **CASCADE** (si cancellano anche gli esami)
- b **SET NULL**
- c **SET DEFAULT**
- d **NO ACTION** (viene impedita l'operazione)

Cose succede degli esami se si **modifica** la matricola nella tabella **STUDENTE**?

- a **CASCADE** (si modifica la **FOREIGN KEY** anche nei corrispondenti esami)
- b **SET NULL**
- c **SET DEFAULT**
- d **NO ACTION** (viene impedita l'operazione)

Definizione : nella tabella figlio

```
CREATE TABLE ESAME
```

```
( .....  
  FOREIGN KEY MATR REFERENCES STUDENTI ,  
  ON DELETE CASCADE ON UPDATE CASCADE )
```

È lecito essere figli di più padri

```
CREATE TABLE ESAME
```

```
( .... PRIMARY KEY(MATR,COD-CORSO) ,  
  FOREIGN KEY MATR REFERENCES STUDENTI  
  ON DELETE CASCADE ON UPDATE CASCADE ,  
  FOREIGN KEY COD-CORSO REFERENCES CORSO  
  ON DELETE NO ACTION ON UPDATE NO ACTION )
```

MATR	NOME	CITTÀ	C-DIP
123			
415			
702			

Una istanza scorretta

esame

MATR	COD-CORSO	DATA	VOTO
123	1	1997-09-07	30
123	2	1998-01-08	28
123	2	1997-08-01	28
702	2	1997-09-07	20
702	1	NULL	NULL
714	1	1997-09-07	28

viola la chiave

viola il NULL

viola la integrità referenziale

Una istanza corretta

MATR	NOME	CITTÀ	C-DIP
123			
415			
702			

esame

MATR	COD-CORSO	DATA	VOTO
123	1	1997-09-07	30
123	2	1998-01-08	28
702	2	1997-09-07	20

Esempio : gestione ordini

COD-CLI	INDIRIZZO	P-IVA

cliente



COD-ORD	COD-CLI	DATA	IMPORTO

ordine



COD-ORD	COD-PROD	QTA

dettaglio



COD-PROD	NOME	PREZZO

prodotto

DDL

Definizione delle tabelle CLIENTE e ORDINE

```
CREATE TABLE CLIENTE  
( COD-CLI CHAR(6) PRIMARY KEY ,  
  INDIRIZZO CHAR(50) , P-IVA CHAR(12) UNIQUE )
```

```
CREATE TABLE ORDINE  
( COD-ORD CHAR(6) PRIMARY KEY ,  
  COD-CLI CHAR(6) NOT NULL DEFAULT='999999' ,  
  DATA DATE , IMPORTO INTEGER ,  
  FOREIGN KEY COD-CLI REFERENCES ORDINE  
  ON DELETE SET DEFAULT  
  ON UPDATE SET DEFAULT)
```

Definizione delle tabelle DETTAGLIO e PRODOTTO

```
CREATE TABLE DETTAGLIO
```

```
( COD-ORD CHAR(6), COD-CLI CHAR(6) , QTA SMALLINT,  
PRIMARY KEY(COD-ORD,COD-PROD) ,  
FOREIGN KEY COD-ORD REFERENCES ORDINE  
ON DELETE CASCADE ON UPDATE CASCADE ,  
FOREIGN KEY COD-PROD REFERENCES PRODOTTO  
ON DELETE NO ACTION ON UPDATE NO ACTION)
```

```
CREATE TABLE PRODOTTO
```

```
( COD-PROD CHAR(6) PRIMARY KEY,  
NOME CHAR(20) , PREZZO SMALLINT )
```

Esercizio: gestione personale

esprimere in SQL la dichiarazione dello schema

MATR	NOME	DATA-ASS	SALARIO	MATR-MGR
1	Piero	1995-01-01	3 M	2
2	Giorgio	1997-01-01	2,5 M	null
3	Giovanni	1996-07-01	2 M	2

impiegato

assegnamento

MATR	NUM-PROG	PERC
1	3	50
1	4	50
2	3	100
3	4	100

progetto

NUM-PROG	TITOLO	TIPO
3	Idea	Esprit
4	Wide	Esprit