Prof. Ing. Alfredo GARRO

BASI DI DATI RELAZIONALI

La maggior parte dei dati strutturati sono oggi organizzati come basi di dati relazionali basate sul modello LOGICO RELAZIONALE dei dati, introdotto nel 1970, e sul linguaggio di descrizione, di manipolazione ed interrogazione dei dati chiamato SQL.

Uno ***schema di relazione*** è una stringa R (A_1 D_1,..., A_n D_n) dove R è il *nome* della relazione, e per ogni i da 1 a n, A_i è un *attributo* e D_i il suo *dominio*, cioè un insieme di valori che possono essere assunti per quell'attributo.
Tipici domini sono:

- stringhe di caratteri:
  - di lunghezza fissa $k > 0$: **character (*k*)** o semplicemente **char (*k*)**
  - di lunghezza variabile fino a massimo $k > 0$: **varying character (*k*)** o semplicemente **varchar (*k*)**
- tipi numerici esatti
  - decimali con k cifre a piacere: **decimal (*k*)** con 0 cifre decimali o **decimal (*k*,*d*)** con d cifre decimali e k-d intere
  - interi con numero prefissato di cifre: **integer**, tipicamente con valori tra -32.768 e 32.767 (cioè rappresentabili con 16 bit), **longint** tipicamente con valori tra -2147483648 e 2147483647 (cioè rappresentabili con 32 bit**)** o **smallint** tipicamente con valori tra 0 e 255 (cioè rappresentabile con 8 bit)
- data con giorno, mese ed anno: **date**
- di tipo booleano: **bit**, con valori 0 (o FALSO o NO) e 1 (o VERO o SI)

La sintassi SQL per descrivere uno schema di relazione è:
TABLE *Studente* (*Matricola* longint, Cog*nome* varchar(20), *Nome* varchar(20), *DataDiNascita* date);

ESERCIZI: *Scrivere vari schemi di relazione.*

Sia $S = R$ (A_1 D_1,..., A_n D_n) uno schema di relazione.
Una *tupla* su S è una sequenza di valori (v_1,..., v_n) tale che per ogni i da 1 a n, v_i è un valore in D_i.
Una ***relazione*** (o ***istanza di schema di relazione*** o *tabella*) su S è un insieme finito di tuple su S.
Un esempio di relazione sullo schema *Studente* è mostrato di seguito. La relazione è rappresentata come una tabella in cui la riga iniziale riporta gli attributi e le altre righe corrispondono alle tuple. La colonna corrispondente all'attributo *Matricola* è indicata con *Studente.Matricola*; le altre allo stesso modo.

Matricola	*Cognome*	*Nome*	*DataDiNascita*
2020	Rossi	Luca	03/10/1980
11245	Bianchi	Giorgio	08/11/1982

La *cardinalità* (cioè il numero di tuple) della relazione è 2. Si noti che essendo un insieme, le tuple in una relazione non sono ordinate, cioè esse possono essere disposte in qualsiasi ordine.

Un esempio di relazione con cardinalità 4 è il seguente:

Matricola	*Cognome*	*Nome*	*DataDiNascita*
2020	Rossi	Luca	03/10/1980
11245	Bianchi	Giorgio	08/11/1982
11909	Verdi	Federica	07/03/1983
1124	Ferrari	Antonio	05/06/1980

Uno ***schema di base di dati*** è un insieme di schemi di relazione.

Un esempio di schema di base di dati è:

DATABASE *Università*
{
  TABLE *Studente* (*Matricola* longint, Cog*nome* varchar(20), *Nome* varchar(20), *DataDiNascita* date);
  TABLE *Corso* (*Codice* char(4), *Titolo* varchar(50), *Docente* varchar(20));
  TABLE *Esame* (*Studente* longint, *Voto* smallint, *Corso* char(4));
}

Sia $D = \{S_1,...,S_m\}$ uno schema di basi di dati.
Una ***base di dati*** su è un insieme di relazioni $\{r_1,...,r_m\}$, dove per ogni i da 1 a m, r_i è una relazione su S_i.
Un esempio di base di dati è:

Studente

Matricola	*Cognome*	*Nome*	*DataDiNascita*
2020	Rossi	Luca	03/10/1980
11245	Bianchi	Giorgio	08/11/1982
11909	Verdi	Federica	07/03/1983
1124	Ferrari	Antonio	05/06/1980

Corso

Codice	*Titolo*	*Docente*
CS02	Sistemi Informativi	Garro
ME05	Microeconomia	Ponti
MA07	Matematica	De Rossi

Esame

Studente	*Voto*	*Corso*
2020	18	CS02
2020	22	MA07
11909	30	CS02
11909	29	ME05
1124	30	ME05

VINCOLI SUI DATI

VALORI NULLI

E' possibile non assegnare alcun valore ad un attributo di una tupla - si dice allora che la tupla ha valore nullo per quell'attributo. La presenza di un valore nullo può significare varie cose: (1) che la tupla non può avere un valore (ad esempio per una nubile la data del matrimonio), (2) che non si conosce un valore ma potrebbe averlo (ad esempio non si conosce il nome del docente di un corso), (3) che non ha un valore (ad esempio il corso non è stato ancora assegnato ad alcun docente).
Se si vuole impedire di avere valori nulli per un attributo bisognerà aggiungere nella definizione *not null*.
Ad esempio con la seguente definizione
  TABLE *Corso* (*Codice* char(4), *Titolo* varchar(50), *Docente* varchar(20) **not null**);
si impone che il docente di un corso non possa avere valore nullo.

VALORI DI DEFAULT

E' possibile preassegnare valori per alcuni attributi allorquando questi ricorrono spesso. Ad esempio, se introduciamo l'attributo Lode nella Tabella Esame:
  TABLE *Esame* (*Studente* longint, *Corso* char(4), *Voto* smallint, *Lode* bit)

per il campo *Lode* conviene definire come valore di default 0 considerato che la maggior parte degli esami non ha la lode:

TABLE *Esame* (*Studente* longint, *Corso* char(4), *Voto* smallint, *Lode* bit **default 0**);

CHIAVE PRIMARIA E UNICITA'

La **chiave primaria** di una relazione è costituita da uno o più attributi che identificano univocamente le tuple della relazione. Una relazione ha al più una chiave primaria. Conviene sempre individuare una chiave primaria. Le chiave primarie non ammettono valori nulli per cui è inutile specificare not null.
Segue lo Schema della Base di Dati *Università* con indicazione delle chiavi primarie:

DATABASE *Università*
{
TABLE *Studente* (*Matricola* longint, Cog*nome* varchar(20), *Nome* varchar(20), *DataDiNascita* date,
primary key(*Matricola*));
TABLE *Corso* (*Codice* char(4), *Titolo* varchar(50), *Docente* varchar(20) not null, **primary key(Codice)**);
TABLE *Esame* (*Studente* longint, *Corso* char(4), *Voto* smallint not null, *Lode* bit default 0,
primary key(*Studente, Corso*));
}

Si noti che le tuple della relazione *Esame* possono avere lo stesso valore per l'attributo *Studente* o lo stesso valore per l'attributo *Corso* ma non tutti e due uguali. In pratica non posso registrare più di una volta ad un dato studente l'esame relativo ad un dato corso.

E' possibile imporre che le tuple di una relazione abbiano valori unici per ulteriori attributi oltre la chiave primaria. Tali attributi sono chiamati *unici* e non sono necessariamente non nulli. Ad esempio se nella relazione *Studente* ci fosse anche l'attributo *CodiceFiscale* questo sarebbe ovviamente unico e potrebbe anche essere non noto per qualche studente. Avremmo allora la seguente definizione:

TABLE *Studente* (*Matricola* longint, Cog*nome* varchar(20), *Nome* varchar(20), *DataDiNascita* date,
CodiceFiscale char(16), primary key(*Matricola*), **unique(*CodiceFiscale*)**);

Nel caso volessimo **anche** imporre un valore non nullo per il codice fiscale dovremmo scrivere nello schema precedente *CodiceFiscale* char(16) not null

INTEGRITA' REFERENZIALE

I vincoli di integrità referenziale servono a stabilire dei legami fra gli attributi delle diverse relazioni che costituiscono la base di dati.
Se, ad esempio, si vuole legare l'attributo *Studente* della relazione *Esame* con l'attributo *Matricola* della relazione *Studente* si scriverà:

TABLE *Esame* (*Studente* longint, *Corso* char(4), *Voto* smallint not null, *Lode* bit default 0,
primary key(*Studente, Corso*),
foreign key *Studente* references Studente(Matricola));

Tale vincolo mi IMPEDISCE di attribuire ad una tupla di *Esame* un valore per l'attributo *Studente* che non è già presente nella Tabella *Studente* come valore dell'attributo *Matricola*. In pratica, ciò equivale ad impedire di registrare un esame ad uno studente inesistente, ossia ad uno studente la cui matricola non compare nella tabella *Studente*.

Allo stesso modo per impedire di registrare un *Esame* relativo ad un *Corso* inesistente, ossia il cui *Codice* non è presente nella Tabella *Corso* scriverò:

TABLE *Esame* (*Studente* longint, *Corso* char(4), *Voto* smallint not null, *Lode* bit default 0,
primary key(*Studente, Corso*),
foreign key *Studente* references Studente(Matricola)
foreign key *Corso* references Corso(Codice));

È opportune notare che affinché si possano definire tali vincoli non ha importanza il nome degli attributi (avrei potuto chiamare l'attributo *Studente* della tabella *Esame* ad esempio *Stud* oppure *CodiceStudente*) importante è invece che tali attributi siano definiti sullo stesso dominio; quindi, è del tutto legittimo il seguente schema di relazione:

TABLE *Esame* (***CodiceStudente*** longint, ***CodiceCorso*** char(4), *Voto* smallint not null, *Lode* bit default 0,
primary key(*Studente, Corso*),
foreign key *CodiceStudente* references Studente(Matricola)
foreign key *CodiceCorso* references Corso(Codice));

mentre è completamente scorretto il seguente schema di relazione:

TABLE *Esame* (*CodiceStudente* **char(10)**, *CodiceCorso* char(4), *Voto* smallint not null, *Lode* bit default 0,
primary key(*Studente, Corso*),
foreign key *CodiceStudente* references Studente(Matricola)
foreign key *CodiceCorso* references Corso(Codice));

Infatti non è possibile legare l'attributo *CodiceStudente* definito sul dominio char(10) con l'attributo *Matricola* di *Studente* che è definito sul dominio longint.

DEFINIZIONE DI UNO SCHEMA LOGICO RELAZIONALE

Ricordiamo che uno *schema di una base di dati* relazionale è un insieme di schemi di relazione o di tabelle $\{S_1, S_2, ..., S_n\}$.
Per ogni schema di relazione (o tabella) S_i:

- va specificato il nome della tabella
- vanno individuati i suoi attributi
- per ogni attributo
  - va specificato il nome dell'attributo, ad esempio *A*
  - va indicato il suo dominio, ad esempio *char*(*k*) o *varchar*(*k*) o *int* o *smallint* o *longint* o *decimal*(*k,d*) o *date* o *bit*
- va specificata la chiave primaria della relazione
- successivamente per ogni attributo non presente nella chiave primaria
  - va specificato se l'attributo identifica univocamente una tupla pur non essendone la chiave primaria: *unique*(*NomeAttributo*)
  - va specificato se l'attributo non può assumere valori nulli (esempio *A longint not null*)
  - vanno specificati eventuali valori di default cioè valori che vengono inseriti automaticamente per semplificare il caricamento di dati, ad esempio *A longint default* 0
- quindi, per ogni attributo
  - vanno inseriti eventuali vincoli sui valori, ad esempio check (A = 'M' OR A='F') per imporre il fatto che A debba assumere solo i valori 'M' o 'F'.
- vanno specificati i **vincoli di integrità referenziale**, ad esempio foreign key(*NomeAttributo*) references NomeAltraTabella(*NomeAltroAttributo*).

SEGUE UN ESEMPIO DI SCHEMA DI BASE DI DATI DEFINITO IN ACCORDO ALLE INDICAZIONI SOPRA RIPORTATE:

DATABASE *Università*
{
TABLE **Studente (***Matricola* longint, Cog*nome* varchar(20) not null, *Nome* varchar(20),
DataDiNascita date not null, *CF* char(16) not null, primary key(*Matricola*),
unique(CF)**);**

TABLE **Corso (***Codice* char(4), *Titolo* varchar(50) not null, *Docente* varchar(20) not null, primary key(Codice)**);**

TABLE **Esame (***Studente* longint, *Corso* char(4), *Voto* smallint not null, *Lode* bit default 0,
primary key(*Studente, Corso*),
check (Voto>=18 AND Voto<=30),
foreign key *Studente* references Studente(Matricola),
foreign key *Corso* references Corso(Codice)**);**
}