

Basi di dati - A.A 2017/2018

Programma accademico “Analisi Matematica 2”

Bregolato Simone, Dalla Via Mattia, Michelotto
Federico

INDICE

1	Descrizione del progetto	2
1.1	Requisiti strutturati	2
1.2	Operazioni sulla base di dati	3
2	Progettazione Concettuale	4
2.1	Modello Concettuale: Entità-Associazione (E-R)	4
2.2	Schema Concettuale, Regole di vincolo	4
3	Progettazione Logica	5
3.1	Ristrutturazione schema E.R.	5
3.2	Schema Concettuale ristrutturato, Regole di vincolo	5
3.3	Modello Logico: Relazionale	6
3.4	Schema Logico, Regole di vincolo	6
4	Codice SQL	7
4.1	Struttura	7
4.2	Query	9

1 Descrizione del progetto

Si vuole realizzare una base di dati per la gestione del programma accademico del corso universitario di Analisi Matematica 2 A.A 2017/2018 dell'Università di Padova, con lo scopo di riunire, per ogni argomento, tutte le definizioni e i teoremi affrontati nel suddetto insegnamento.

1.1 Requisiti strutturati

Definizione:

Una definizione possiede un Nome, un Testo, una Descrizione ed un Codice attraverso il quale viene identificata. Una definizione può basarsi su altre definizioni. Una definizione appartiene ad un unico argomento e può essere usata come ipotesi di più teoremi.

Teorema:

Un teorema è composto da una Ipotesi, una Tesi, un Nome, una Descrizione, un attributo Tipologia che specifica se è propriamente un teorema oppure una proposizione o un corollario. Le proposizioni e i corollari sono anch'essi teoremi dal punto di vista della formulazione. La proposizione è un teorema solitamente di importanza non primaria. Il corollario è invece una proposizione che può essere dedotta facilmente da un teorema enunciato e dimostrato. Nel caso si ritenga opportuno è previsto un Disegno. In mancanza di un buon identificatore naturale introduciamo un Codice univoco come identificatore.

Un teorema può avere in alcuni casi una dimostrazione e degli esempi, inoltre è associato ad un unico argomento.

Sezione:

Una sezione può avere uno o più argomenti ad essa associati, ogni sezione possiede un Nome ed una Descrizione, inoltre è identificata da un Codice numerico.

Argomento:

Un argomento ha come proprietà: un Nome, una Descrizione ed un Numero. L'attributo Numero permette di identificare un argomento all'interno della sezione cui appartiene. Ad un argomento possono essere associate più definizioni e teoremi.

Dimostrazione:

Ogni dimostrazione ha un Testo, inoltre attraverso un attributo di nome Idea si può descrivere l'intuizione che c'è alla base della dimostrazione, in modo da poter facilitare la

comprensione.

All'interno del testo di dimostrazione, ci possono essere più richiami a diversi teoremi.

Esempio:

Un esempio deve avere un Testo ed un attributo che esprime la Difficoltà da 1 a 5, inoltre può avere un Disegno.

Un esempio può essere associato ad un solo teorema, e l'attributo Numero permette di identificare un esempio relativo ad un teorema.

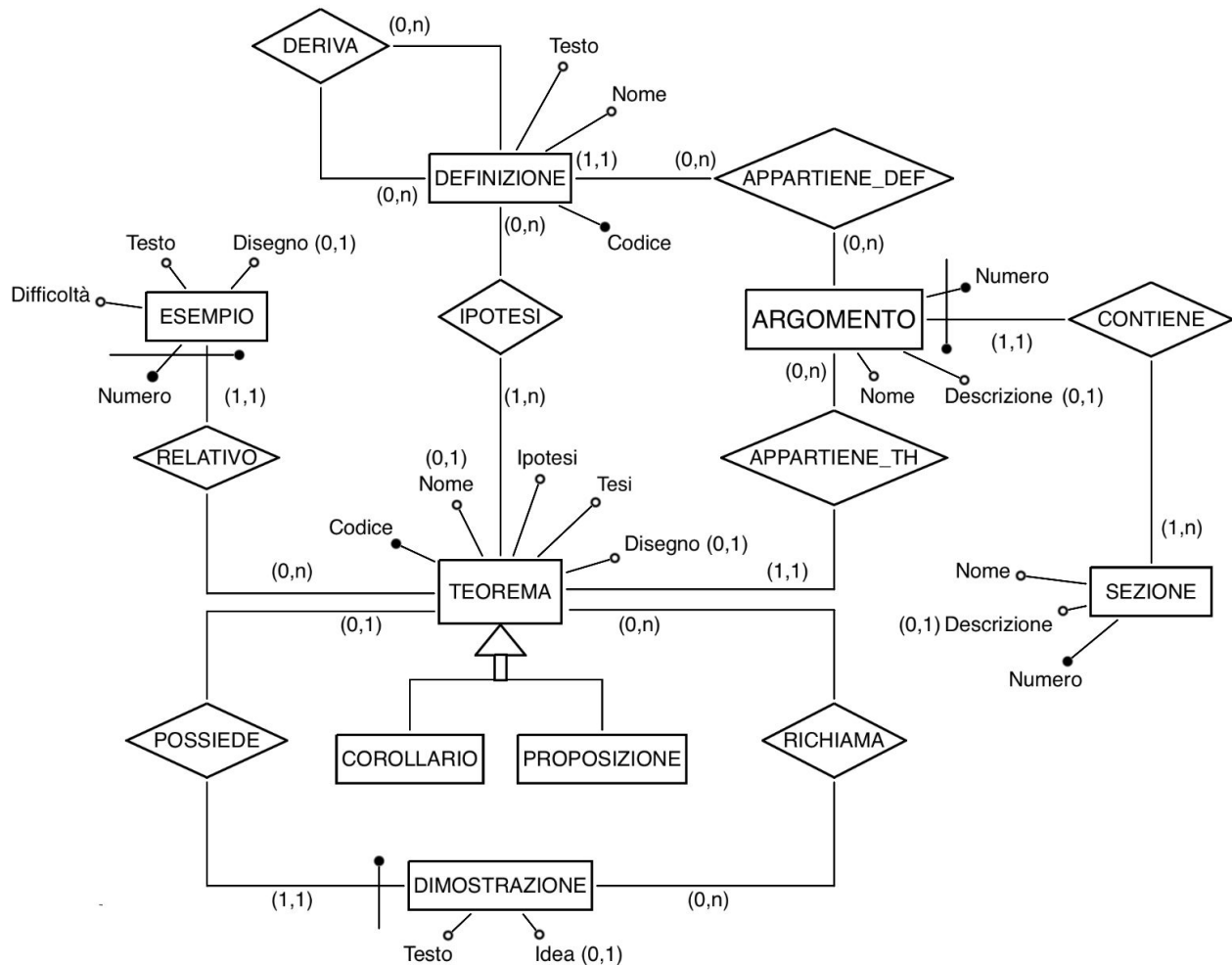
1.2 Operazioni sulla base di dati

Operazione	Tipo	Frequenza
Lista di 5 definizioni selezionate casualmente	Interrogazione	1000/Giorno
Lista di 4 teoremi selezionati casualmente	Interrogazione	1000/Giorno
Lista delle definizioni ordinata secondo sezione e argomento	Interrogazione	500/Giorno
Lista dei teoremi ordinati secondo sezione e argomento	Interrogazione	500/Giorno
Lista delle definizioni appartenenti ad un certo argomento	Interrogazione	100/Giorno
Lista delle definizioni appartenenti ad una certa sezione	Interrogazione	100/Giorno
Lista dei teoremi appartenenti ad un certo argomento	Interrogazione	100/Giorno
Lista dei teoremi appartenenti ad una certa sezione	Interrogazione	100/Giorno
Ricerca delle definizioni contenenti una certa stringa nel Nome o nel Testo	Interrogazione	200/Giorno
Ricerca dei teoremi contenenti una certa stringa nel Nome, Ipotesi e Tesi	Interrogazione	200/Giorno

Le operazioni di inserimento vengono effettuate in modalità batch nel momento della creazione della base di dati. Le modifiche, seppur molto rare, potranno essere effettuate mediante l'utilizzo diretto del dbms mediante un client sql.

2 Progettazione Concettuale

2.1 Modello Concettuale: Entità-Associazione (E-R)



2.2 Schema Concettuale, Regole di vincolo

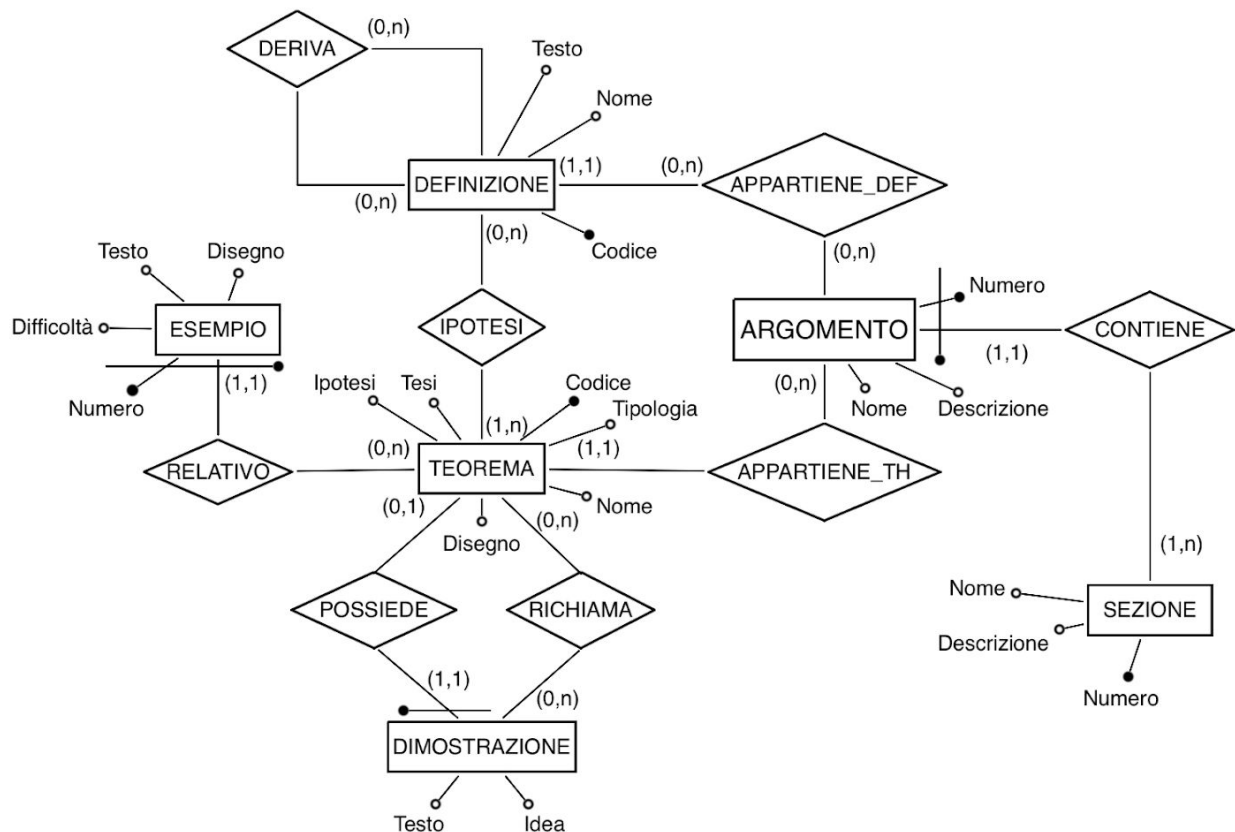
Per poter rappresentare fedelmente la realtà d'interesse descritta nella sezione 1.1, è necessario definire delle regole per esprimere quei vincoli non esprimibili utilizzando il modello E.R.

- R1: La difficoltà di ESEMPIO dev'essere compresa tra 1 e 5.

3 Progettazione Logica

3.1 Ristrutturazione schema E.R.

La generalizzazione di TEOREMA presente nello schema E.R. è stata risolta accorpendo le entità figlie nell'entità genitore come conseguenza del fatto che l'entità COROLLARIO e l'entità PROPOSIZIONE non hanno differenze strutturali. È stato aggiunto l'attributo Tipo per specificare se si tratta di un teorema, corollario o proposizione.

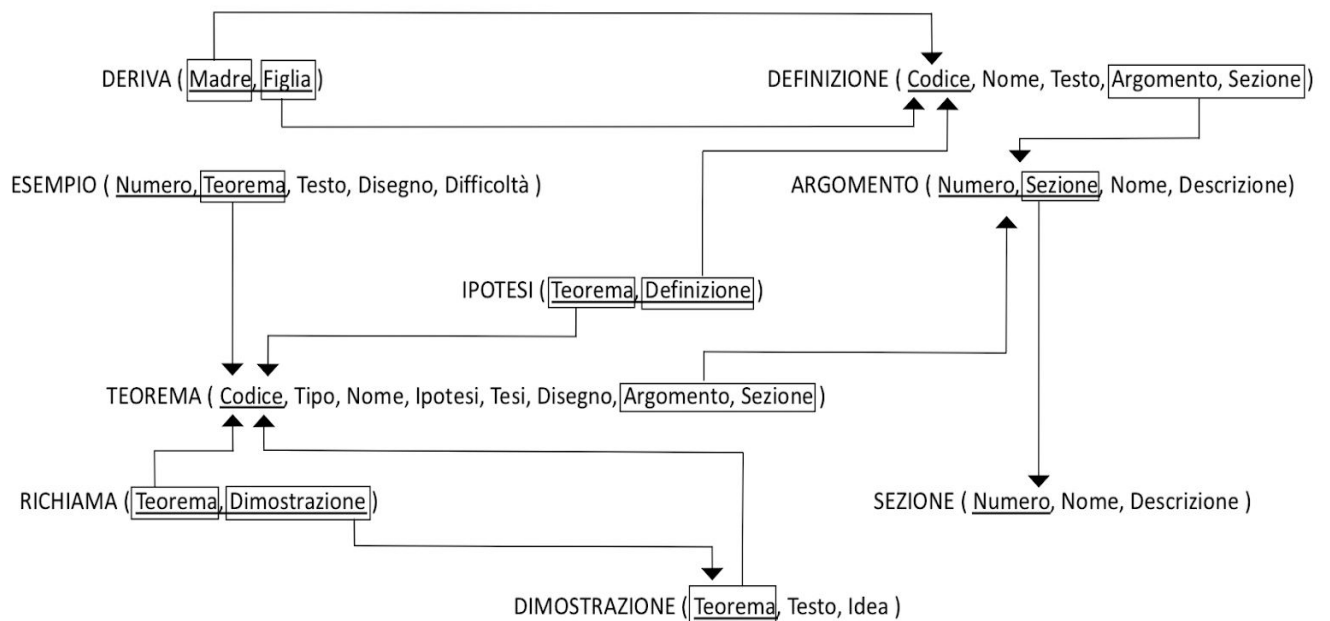


3.2 Schema Concettuale ristrutturato, Regole di vincolo

Oltre alle regole già specificate nello schema non ristrutturato

- R2: L'attributo Tipo può assumere solo i valori "teorema", "corollario" o "proposizione".

3.3 Modello Logico: Relazionale



3.4 Schema Logico, Regole di vincolo

- R3: Gli attributi Argomento, Sezione e Testo di DEFINIZIONE non possono assumere valore nullo.
- R4: Gli attributi Argomento, Sezione, Ipotesi e Tesi di TEOREMA non possono assumere valore nullo.
- R5: L'attributo Nome di SEZIONE non può assumere valore nullo.
- R6: L'attributo Nome di ARGOMENTO non può assumere valore nullo.
- R7: L'attributo Testo di DIMOSTRAZIONE non può assumere valore nullo.
- R8: L'attributo Testo di ESEMPIO non può assumere valore nullo.

4 Codice SQL

4.1 Struttura

```
CREATE TABLE SEZIONE (  
    Numero SERIAL,  
    Nome VARCHAR(50) UNIQUE NOT NULL,  
    Descrizione TEXT,  
  
    PRIMARY KEY (Numero)  
);  
  
CREATE TABLE ARGOMENTO (  
    Numero SERIAL,  
    Sezione SERIAL,  
    Nome VARCHAR(60) UNIQUE NOT NULL,  
    Descrizione TEXT,  
  
    PRIMARY KEY (Numero, Sezione),  
    FOREIGN KEY (Sezione) REFERENCES SEZIONE(Numero)  
        ON DELETE NO ACTION  
        ON UPDATE CASCADE  
);  
  
CREATE TABLE DEFINIZIONE (  
    Codice SERIAL,  
    Nome VARCHAR(50) UNIQUE NOT NULL,  
    Testo TEXT NOT NULL,  
    Argomento SERIAL NOT NULL,  
    Sezione SERIAL NOT NULL,  
  
    PRIMARY KEY (Codice),  
    FOREIGN KEY (Argomento, Sezione) REFERENCES ARGOMENTO(Numero, Sezione)  
        ON DELETE NO ACTION  
        ON UPDATE CASCADE  
);  
  
CREATE TABLE DERIVA (  
    Madre SERIAL,  
    Figlia SERIAL,  
  
    PRIMARY KEY (Madre, Figlia),  
    FOREIGN KEY (Madre) REFERENCES DEFINIZIONE(Codice)  
        ON DELETE NO ACTION  
        ON UPDATE CASCADE,  
    FOREIGN KEY (Figlia) REFERENCES DEFINIZIONE(Codice)  
        ON DELETE NO ACTION  
        ON UPDATE CASCADE  
);  
  
CREATE TYPE TipoTeorema AS ENUM('teorema', 'corollario', 'proposizione');
```



```

CREATE TABLE TEOREMA (
    Codice SERIAL,
    Tipo TipoTeorema DEFAULT 'teorema',
    Nome VARCHAR(100) UNIQUE,
    Ipotesi TEXT NOT NULL,
    Tesi TEXT NOT NULL,
    Disegno BYTEA,
    Argomento SERIAL NOT NULL,
    Sezione SERIAL NOT NULL,

    PRIMARY KEY (Codice),
    FOREIGN KEY (Argomento, Sezione) REFERENCES ARGOMENTO(Numero, Sezione)
        ON DELETE NO ACTION
        ON UPDATE CASCADE
);

```

```

CREATE TABLE IPOTESI (
    Teorema SERIAL,
    Definizione SERIAL,

    PRIMARY KEY (Teorema, Definizione),
    FOREIGN KEY (Teorema) REFERENCES TEOREMA(Codice)
        ON DELETE NO ACTION
        ON UPDATE CASCADE,
    FOREIGN KEY (Definizione) REFERENCES DEFINIZIONE(Codice)
        ON DELETE NO ACTION
        ON UPDATE CASCADE
);

```

```

CREATE TABLE DIMOSTRAZIONE (
    Teorema SERIAL,
    Testo TEXT NOT NULL,
    Idea TEXT,

    PRIMARY KEY (Teorema),
    FOREIGN KEY (Teorema) REFERENCES TEOREMA(Codice)
        ON DELETE NO ACTION
        ON UPDATE CASCADE
);

```

```

CREATE TABLE RICHIAMA (
    Teorema SERIAL,
    Dimostrazione SERIAL,

    PRIMARY KEY (Teorema, Dimostrazione),
    FOREIGN KEY (Teorema) REFERENCES TEOREMA(Codice)
        ON DELETE NO ACTION
        ON UPDATE CASCADE,
    FOREIGN KEY (Dimostrazione) REFERENCES DIMOSTRAZIONE(Teorema)
        ON DELETE NO ACTION
        ON UPDATE CASCADE
);

```

```

CREATE TABLE ESEMPIO (
    Numero SERIAL,

```

```

Teorema SERIAL,
Testo TEXT NOT NULL,
Disegno BYTEA,
Difficolta INT,

PRIMARY KEY (Numero, Teorema),
FOREIGN KEY (Teorema) REFERENCES TEOREMA(Codice)
    ON DELETE NO ACTION
    ON UPDATE CASCADE,
CHECK (Difficolta >= 1 AND Difficolta <= 5)
);

```

4.2 Query

- ❖ Trovare la definizione che è ipotesi del maggior numero di teoremi

```

SELECT Codice, Nome, COUNT(*) AS NumeroTeoremi
FROM DEFINIZIONE D
JOIN IPOTESI I ON I.Definizione = D.Codice
GROUP BY Codice, Nome
HAVING COUNT(*) = (SELECT MAX(N)
                    FROM (SELECT COUNT(*) AS N
                          FROM IPOTESI
                          GROUP BY Definizione) AS M);

```

- ❖ Per ogni argomento, trovare il numero medio di ipotesi appartenenti a ciascun teorema

```

SELECT A.Sezione, A.Numero, A.Nome, AVG(NumeroIpotesi) AS MediaIpotesi
FROM ARGOMENTO A
JOIN TEOREMA T ON T.Argomento = A.Numero AND T.Sezione = A.Sezione
JOIN (SELECT Teorema, COUNT(*) AS NumeroIpotesi
      FROM IPOTESI
      GROUP BY Teorema) I ON I.Teorema = T.Codice
GROUP BY A.Sezione, A.Numero, A.Nome;

```

- ❖ Lista delle definizioni che non sono ipotesi di alcun teorema

```

SELECT Codice, Nome
FROM DEFINIZIONE
WHERE Codice <> all (SELECT Definizione FROM IPOTESI);

```

- ❖ Per la dimostrazione 1 mostrare quali teoremi richiama

```
SELECT Codice, Tipo, Nome
FROM TEOREMA T
JOIN RICHIAMA R ON R.Teorema = T.Codice
WHERE Dimostrazione = 1;
```

- ❖ Data la definizione di Insieme aperto, mostrare quali teoremi ce l'hanno come ipotesi

```
SELECT T.Codice, T.Tipo, T.Nome
FROM TEOREMA T
JOIN IPOTESI I ON I.Teorema = T.Codice
JOIN DEFINIZIONE D ON D.Codice = I.Definizione
WHERE D.Nome = 'Insieme aperto';
```

- ❖ Mostrare i teoremi che hanno un solo esempio, assieme al relativo esempio

```
SELECT T.Codice, T.Tipo, T.Nome, E.Testo AS Esempio, E.Difficolta
FROM TEOREMA T
JOIN ESEMPIO E ON E.Teorema = T.Codice
WHERE E.Teorema = any (SELECT Teorema
                        FROM ESEMPIO
                        GROUP BY Teorema
                        HAVING COUNT(*) = 1);
```